CONTRIBUIÇÕES À CODIFICAÇÃO EFICIENTE DE IMAGEM E VÍDEO UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALA

Nelson Carreira Francisco

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
Nuno Miguel Morais Rodrigues

Rio de Janeiro
Novembro de 2012

CONTRIBUIÇÕES À CODIFICAÇÃO EFICIENTE DE IMAGEM E VÍDEO UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALA

Nelson Carreira Francisco

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. Nuno Miguel Morais Rodrigues, Ph.D

Prof. Sérgio Lima Netto, Ph.D.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. Ricardo Lopes de Queiroz, Ph.D

Prof. Carla Liberal Pagliari, Ph.D

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2012

Francisco, Nelson Carreira

Contribuições à Codificação Eficiente de Imagem e Vídeo Utilizando Recorrência de Padrões Multiescala/Nelson Carreira Francisco. – Rio de Janeiro: UFRJ/COPPE, 2012.

XXII, 270 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva

Nuno Miguel Morais Rodrigues

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2012.

Referências Bibliográficas: p. 257 – 270.

1. Casamento de Padrões Multiescalas. 2. Compressão de imagens estáticas. 3. Compressão de Documentos Compostos. 4. Compressão de Vídeo. 5. Filtragem de Redução de Efeito de Bloco. I. da Silva, Eduardo Antônio Barros *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Aos meus pais,
Lúcia e Isidro.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores: Prof. Eduardo Silva, Prof. Nuno Rodrigues e Prof. Sérgio Faria, pela sua enorme capacidade de orientação, que ajudou a enriquecer a qualidade do trabalho apresentado nesta tese. Foram os grande responsáveis pelo rumo tomado pelo meu percurso acadêmico, e trabalhar com alguém tão motivador foi sem dúvida um enorme privilégio. Também pela amizade que demonstraram ao longo dos últimos anos, transcendendo em muito o conceito de orientadores.

Seguidamente gostaria de expressar os meus agradecimentos à Fundação para a Ciência e a Tecnologia, pelo suporte financeiro prestado. Também ao Instituto de Telecomunicações, à Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e ao Laboratório de Processamento de Sinais da Universidade Federal do Rio de Janeiro, pelas excelentes condições físicas e materiais proporcionadas, que permitiram levar a cabo este trabalho.

Uma agradecimento especial aos meus colegas do IT: Danillo, Sylvain, Sandro e Lucas, pela amizade e pelas trocas de ideias construtivas que me proporcionaram ao longo destes anos. Também pelo privilégio de termos trabalhado em conjunto nalgumas ocasiões. Do mesmo modo, gostaria de agradecer ao colegas do LPS, pela forma calorosa como me receberam e pela amizade que demonstraram durante os períodos que passei no Rio. Um agradecimento especial ao José Antonio, pela amizade e apoio num dos momentos mais difíceis desta longa jornada.

À minha namorada Auridélia, pelo carinho, compreensão e apoio, porque as conquistas têm outro sabor quando temos com quem compartilhá-las. "*Cause it's always better together...*"

Aos meus pais, Isidro e Lúcia, meu muito obrigado pelo seu apoio e amor incondicional. Também pelos valores que me transmitiram, fazendo-me acreditar que grandes feitos podem ser atingidos através de trabalho árduo e muita dedicação.

Um agradecimento especial a todos os meus amigos, que pelos momentos de descontração e lazer me ajudaram a manter o equilíbrio entre o trabalho e uma vida social saudável. Desculpem não mencionar todos, mas foram com certeza lembrados.

Por último, gostaria de agradecer aos revisores anônimos e a todos aqueles que, com críticas e sugestões, contribuíram para aumentar a qualidade do trabalho apresentado nesta tese.

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 CODIFICAÇÃO EFICIENTE DE IMAGEM E VÍDEO UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALA

Nelson Carreira Francisco

Novembro/2012

Orientadores: Eduardo Antônio Barros da Silva
Nuno Miguel Morais Rodrigues

Programa: Engenharia Elétrica

A crescente utilização do suporte digital como meio privilegiado de partilha e armazenamento motivou a pesquisa apresentada nesta tese, por codificadores eficientes de imagens e vídeo baseados no MMP (do original *Multidimentional Multiscale Parser*).

São propostas várias técnicas inovadoras, que incluem um filtro redutor de efeito de bloco, e técnicas de redução da complexidade computacional, que reduziram a um décimo o tempo de codificação sem perdas significativas de desempenho de compressão. Essas melhorias foram combinadas em novos algoritmos, vocacionados para a compressão de documentos compostos digitalizados e sinais de vídeo.

Os resultados do codificador de documentos compostos digitalizados proposto foram comparados com os de alguns dos melhores algoritmos existentes, como os baseados no modelo MRC (do inglês *Mixed Raster Content*), superando o desempenho destes.

Para codificação de vídeo, foram desenvolvidos dois novos algoritmos. O primeiro, denominado MMP-video, é baseado na norma H.264/AVC, com as transformadas substituídas pelo MMP. São usadas algumas das técnicas desenvolvidas para codificação de imagens, conjuntamente com novos métodos otimizados em função das características dos sinais de vídeo. O resultado é um codificador de vídeo totalmente baseado em casamento de padrões, que supera o desenpenho taxa-distorção do H.264/AVC. O segundo, denominado 3D-MMP, combina uma predição hierárquica com uma extensão 3D do MMP, usada na codificação do resíduo. Este algoritmo abriu novas linhas de pesquisa, que incluem a compressão de sinais provenientes de radares meteorológicos ou imagiologia médica.

Os resultados obtidos validaram os métodos propostos e demonstraram que apesar da sua ainda elevada complexidade computacional, o MMP pode ser visto como uma alternativa ao tradicional paradigma das transformadas.

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 EFFICIENT IMAGE AND VIDEO COMPRESSION USING MULTISCALE RECURRENT PATTERNS

Nelson Carreira Francisco

November/2012

Advisors: Eduardo Antônio Barros da Silva
Nuno Miguel Morais Rodrigues

Department: Electrical Engineering

Given the increasing popularity of the digital media support as a privileged way for sharing and storing visual information, the present thesis investigates efficient image and video compression frameworks based on MMP (Multidimensional Multiscale Parser).

Several new techniques are proposed, including a new post-processing deblocking filter, which increased both the objective and perceptual quality of the reconstructed images and video sequences, as well as complexity reduction techniques, which allowed to reduce to one tenth the time needed to encode an image, without significant performance losses. These improvements were combined in several compression frameworks for scanned compound documents and video signals.

The scanned compound document encoder was evaluated against several state-of-the-art codecs, including some based on the successful MRC (Mixed Raster Content) model, being able to outperform both objectively and perceptually all of them.

For the case of video coding, two new encoders were developed. The first, referred to as MMP-video, is based on the H.264/AVC's reference software, with the original transform being replaced by MMP. This codec uses some of the techniques developed for image compression, together with new procedures optimized to exploit particular features from video signals. The result is a fully pattern-matching based video codec, able to consistently outperform the H.264/AVC. The second codec, named 3D-MMP, combines a hierarchical volumetric prediction with a 3D extension of MMP for residue coding. This compression framework opened several new research lines, including the compression of signals from other sources, such as meteorological radar signals or tomographic scans.

The results obtained in this thesis validate the proposed techniques and show that, in spite of its higher computational complexity, MMP can be regarded as an alternative to the traditional transform-based methods.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ABS	Adaptive Block Size, p. 24
AVC	Advanced Video Coding, p. 4
BD	Bjøntegaard delta, p. 27
CABAC	Context-Adaptive Binary Arithmetic Coding, p. 25
CAVLC	Context-Adaptive Variable Length Code, p. 25
CBP	Coded Block Pattern, p. 26
CIF	Common Intermediate Format, p. 27
DCT	Discrete Cosine Transform, p. 1
DC	Direct Current, p. 10
DV	Directional Vector, p. 219
DWT	Discrete Wavelet Transform, p. 1
EPZS	Enhanced Predictive Zonal Search, p. 44
FIR	Finite Impulse Response, p. 38
GOP	Group Of Pictures, p. 27, 52
GPU	Graphic Processing Unit, p. 36
HEVC	High Efficiency Video Coding, p. 4
JBIG	Joint Bi-level Image Experts group, p. 13
JPEG	Joint Photographic Experts Group, p. 11
LSP	Least Squares Prediction, p. 10
LZ	Lempel-Ziv, p. 7

MB	MacroBlocks, p. 24
MDC	Maximum Dictionary Capacity, p. 32
ME	Motion Estimation, p. 22
MFV	Most Frequent Value, p. 10
MMP	Multidimensional Multiscale Parser, p. 2
MRC	Mixed Raster Content, p. 92
MSE	Mean Square Error, p. 84
MV	Motion Vector, p. 22
OCR	Optical Character Recognition, p. 108
PSNR	Peak Signal-to-Noise Ratio, p. 84
QP	Quantization Parameter, p. 24
RD	Rate-Distortion, p. 24
SAD	Sum of Absolute Differences, p. 24
SATD	Sum of Absolute Transformed Differences, p. 24
SPITH	Set Partitioning in Hierarchical Trees, p. 13
SSE	Sum of Square Errors, p. 30
VBR	Variable Bit-Rate, p. 27
VQ	Vector Quantization, p. 7
dB	Decibel, p. 84

Capítulo 1

Introdução

1.1 Motivações

Nos decorrer dos últimos anos, os conteúdos multimídia digitais têm sido alvo de uma crescente popularidade, que se deveu principalmente aos avanços verificados no campo da eletrônica de consumo, cada vez mais acessível e com maiores potencialidades. Como consequência, a quantidade de informação que necessita de ser manipulada e armazenada é cada vez maior.

O vídeo digital encontra-se atualmente em toda a parte: a tradicional televisão analógica deu lugar a novos serviços de televisão digital, e temos assistido ao aparecimento de inúmeras aplicações e provedores de vídeo, tais como o *Youtube*, onde os usuários podem assistir e compartilhar vídeos com utilizadores do mundo inteiro. Vídeos e imagens tornaram-se habituais em sítios de internet, e a maioria de nós recorre usualmente a computadores ou dispositivos móveis para consultar as últimas notícias.

Paralelamente, as bibliotecas de documentos digitais também têm vindo a tornar-se cada vez mais comuns. Muitos jornais internacionais passaram a disponibilizar versões eletrônicas, e um crescente número de bibliotecas têm vindo a criar cópias digitais das suas coleções, como forma de disponibilizar documentos históricos sensíveis a um maior número de utilizadores, sem os problemas relacionados com a sua preservação.

A enorme quantidade de informação que necessita ser armazenada e transmitida impõe a necessidade de desenvolvimento de algoritmos de compressão eficientes para imagens e vídeo, visto que o crescimento da capacidade dos dispositivos de armazenamento e da largura de banda dos sistemas de comunicações não é por si só suficiente para suprir esta demanda.

Os algoritmos baseados no paradigma da transformada e quantização têm dominado esta área de aplicação no decorrer das últimas décadas, quer usando as tradicionais transformadas discreta do cosseno (DCT) e de *wavelet* discreta (DWT), ou as novas transformadas inteiras, adotadas recentemente por algumas normas de codificação. No entanto,

apesar de se revelarem particularmente eficientes para imagens suaves, os algoritmos baseados neste paradigma tendem a apresentar um fraco desempenho quando usados para comprimir outros tipos de imagens que apresentam conteúdos de alta frequência, como imagens de texto, imagens sintéticas, documentos compostos e texturas, entre outros.

A eficiência destes métodos baseia-se na compactação de energia proporcionada pelas transformadas, quando a imagem a codificar apresenta um elevado grau de correlação espacial. Nesses casos, os coeficientes da transformada correspondentes às frequências mais elevadas tendem a ser pouco relevantes, ou mesmo negligenciáveis, e podem por isso ser sujeitos a uma quantização agressiva ou simplesmente descartados. Tal fato permite atingir elevadas taxas de compressão sem com isso comprometer a qualidade visual das imagens reconstruídas. Em alguns casos, a eficiência de codificação pode ainda ser melhorada com recursos a técnicas preditivas, que permitem uma melhor exploração da correlação espacial e temporal dos sinais de entrada. Num estágio final, é usado um codificador entrópico [1] para reduzir a correlação estatística remanescente.

No entanto, quando o sinal de entrada não apresenta uma natureza passa-baixas, como é o caso dos documentos compostos, a aplicação de um passo de quantização elevado aos coeficientes de alta-frequência resulta na introdução de alguns artefatos que comprometem a qualidade perceptual da imagem reconstruída. Por outro lado, se esses coeficientes não forem sujeitos a esses passos de quantização elevados, torna-se impossível atingir elevadas taxas de compressão.

Como tentativa de solucionar este problema, foram apresentados alguns algoritmos híbridos vocacionados para a compressão de documentos compostos. A estratégia por eles adoptada passa por segmentar a imagem de entrada numa componente passa-altas (texto) a passa-baixas (zonas de imagem suave), aplicando depois a cada componente um algoritmo especificamente otimizado em função das suas características. No entanto, o sucesso destes métodos depende significativamente do desempenho da segmentação, que não se revela capaz de gerar resultados satisfatórios sob todas as condições.

Estas limitações motivaram a procura por paradigmas de compressão alternativos para imagens e vídeo, mas a busca por um método universal provou ser um desafio difícil de superar.

A investigação descrita nesta tese baseia-se num algoritmo bastante promissor, que já provou no passado a sua versatilidade para um leque variado de sinais de entrada. O casamento recorrente de padrões multiescalas (MMP, do inglês *Multidimensional Multiscale Parser*) [2, 3] foi originalmente proposto como um algoritmo de compressão com perdas genérico. Foi desde então aplicado com sucesso para a compressão com e sem perdas de vários tipos de sinais de entrada, com resultados que competem com o estado da arte para diversas aplicações. A compressão de imagens com perdas [4, 5] e sem perdas [6], a compressão de sinais de vídeo [7], imagens estereoscópicas [8], impressões digitais multi-vistas [9] ou electrocardiogramas [10–12] são alguns exemplos dessas aplicações.

1.2 Objetivos

As limitações dos algoritmos de compressão existentes motivaram o estudo de paradigmas de codificação alternativos, e o desenvolvimento de métodos de compressão versáteis. Entre o vasto leque de propostas, o algoritmo MMP assume uma posição privilegiada, dado já ter dado provas da sua versatilidade e excelente desempenho em várias aplicações de compressão.

O trabalho descrito nesta tese investiga esquemas de compressão eficientes baseados no MMP, de modo a explorar o potencial deste paradigma de codificação, para compressão de informação visual. Os objetivos a atingir prendem-se com a otimização do desempenho global do algoritmo para imagens estáticas e com o desenvolvimento de arquitecturas de compressão eficientes para documentos compostos digitalizados e sinais de vídeo. Pretende-se ainda estudar a viabilidade de uma variante volumétrica do algoritmo MMP num esquema de compressão para sinais tridimensionais que explore simultaneamente a correlação espaciotemporal dos sinais de entrada, com base num esquema preditivo hierárquico.

Deste modo, os tópicos de trabalho principais abordados nesta tese podem ser sumarizados nos seguintes objetivos:

- **Otimizar o desempenho do MMP para a compressão de imagens.**

  O foco desta investigação estará na otimização do algoritmo para a codificação quer de imagens naturais, quer de imagens de texto, de modo a desenvolver um método de compressão de documentos compostos digitalizados, competitivo com o atual estado da arte. A elevada heterogeneidade verificada neste tipo de imagens constitui um importante obstáculo ao desenvolvimento de esquemas eficientes de compressão.

  As otimizações visam não só o acréscimo da qualidade objetiva como também perceptual das imagens reconstruídas, de modo a afirmar o MMP como uma alternativa viável a outros codificadores que compõem o estado da arte nesta área de aplicação. Os resultados dos esquemas de compressão desenvolvidos serão não só comparados com o desempenho de algoritmos que compõem o estado da arte, como também com os das versões anteriores do MMP.

- **Investigar a eficiência do paradigma do MMP para aplicações de compressão de vídeo.**

  Um objetivo desta tese passa pelo desenvolvimento de um codificador de vídeo totalmente baseado no paradigma do casamento de padrões.

  Testes preliminares onde o MMP foi usado para comprimir o resíduo resultante da estimação de movimento num codificador híbrido forneceram resultados promissores [7, 13, 14]. No entanto, estas investigações anteriores eram suportadas

por uma versão mais rudimentar do MMP [15], tendo as transformadas sido ainda usadas na codificação dos quadros de referência.

Arquiteturas de codificação de vídeo otimizadas deverão ser estudadas, de modo a permitir a total substituição das transformadas no novo codificador proposto.

Os resultados do codificador desenvolvido serão avaliados por comparação aos da norma vigente para compressão de sinais de vídeo: o codificador H.264/AVC, no seu perfil *high*. A mais recente proposta de norma, HEVC [16], não foi utilizada para efeitos de comparação de resultados, visto que no decorrer do tempo de desenvolvimento do trabalho apresentado nesta tese, a mesma ainda não se encontrava completamente implementada.

- **Abordar os problemas relativos à complexidade computacional**

  O algoritmo MMP já demonstrou o seu elevado desempenho taxa-distorção e a sua versatilidade em investigações anteriores, mas ainda apresenta um entrave significativo ao seu uso prático para um elevado número de aplicações: a sua complexidade computacional.

  A redução da complexidade computacional do MMP poderá constituir um passo decisivo na afirmação do MMP como uma alternativa prática viável ao paradigma da transformada e quantização.

  Os resultados atingidos com os métodos desenvolvidos serão avaliados por comparação com versões de referência do MMP e outros trabalhos anteriores nesta área.

- **Desenvolver uma arquitetura de codificação volumétrica baseada no MMP.**

  A investigação de um esquema de codificação preditivo tridimensional também constitui um tópico de pesquisa para esta tese. Combinando uma extensão volumétrica do MMP com um esquema de predição hierárquico tridimensional, será possível desenvolver um algoritmo de compressão de sinais volumétricos, aplicável a uma vasta gama de sinais de entrada, tais como os sinais provenientes de radares meteorológicos ou mesmo sinais de vídeo.

  Este tópico de pesquisa inclui o desenvolvimento de modos de predição tridimensionais a arquiteturas de codificação otimizadas, de modo a explorar de forma eficiente a redundância espaciotemporal.

  Os resultados experimentais serão comparados com os de outros algoritmos anteriormente desenvolvidos para as aplicações em questão, bem como de codificadores que compõem o estado da arte para essas aplicações.

1.3 Organização da tese

A presente tese encontra-se organizada da seguinte forma. Os capítulos 1 a 8 encontram-se escritos em português e têm por objetivo fornecer uma visão geral do trabalho realizado no âmbito desta tese. Estes capítulos são complementados por uma descrição mais exaustiva, apresentada nos apêndices A a H, estando estes escritos em inglês.

O presente capítulo apresenta uma introdução relativa aos tópicos de pesquisa abordados nesta tese. A motivação que levou ao desenvolvimento deste trabalho é enquadrada, sendo ainda discutidos os principais objetivos e metas a atingir.

O Capítulo 2 apresenta uma breve revisão sobre os principais aspectos do codificador MMP. Alguns resultados experimentais obtidos para a codificação de diversos tipos de imagens são igualmente apresentados neste capítulo, sendo comparados com os de algoritmos baseados em transformadas, que constituem o estado da arte nesta área de aplicação. Maiores detalhes sobre os algoritmos baseados no MMP e seus resultados se encontram no Apêndice B.

No Capítulo 3, é descrito um novo esquema de compressão direcionado à codificação de documentos compostos digitalizados. São apresentadas algumas modificações operadas no MMP, com o intuito de otimizar o seu desempenho especificamente para a compressão das componentes correspondentes a imagens naturais e regiões de texto. Os resultados deste novo esquema de codificação são avaliados por comparação com os dos algoritmos que constituem o estado da arte na codificação de documentos compostos. Mais detalhes relativos ao método proposto, bem como uma análise mais abrangente dos resultados são apresentados no Apêndice C.

No Capítulo 4, é descrita a utilização do MMP num algoritmo de compressão de vídeo totalmente baseado no paradigma do casamento de padrões. No seguimento do desempenho competitivo atingido pelo MMP na codificação de imagens estáticas e do resíduo resultante da estimação de movimento, foi desenvolvido um novo codificador de vídeo no qual todas as transformadas utilizadas pela norma H.264/AVC foram substituídas pelo MMP. Os resultados deste novo codificador são avaliados por comparação com a norma H.264/AVC. Mais resultados e detalhes de implementação do algoritmo proposto são apresentados no Apêndice D.

No Capítulo 5, são propostas duas novas técnicas de redução de complexidade computacional. Uma destas técnicas é especificamente orientada para o algoritmo MMP, enquanto que a segunda pode ser facilmente adaptada a outros algoritmos baseados em casamento de padrões. Ambos os métodos apresentam ganhos significativos de tempo de computação, tanto do lado do codificador como do decodificador. A redução da complexidade computacional é avaliada por comparação com versões de referência do algoritmo MMP. O Apêndice E fornece uma descrição mais detalhada destas técnicas.

Um novo método de pós-processamento para redução do efeito de bloco nas imagens

reconstruídas é apresentado no Capítulo 6. Este método foi originalmente desenvolvido com o intuito de aumentar a qualidade perceptual das imagens codificadas com o MMP. O esquema de filtragem desenvolvido ultrapassou algumas limitações dos métodos anteriormente propostos para este efeito. A natureza deste método, combinada com uma otimização cuidada, permitiu que este fosse aplicado com sucesso não só a imagens como vídeos, codificados utilizando vários algoritmos. Deste modo, os resultados do método proposto são avaliados não só para imagens codificadas com o MMP, como também com alguns codificadores baseados em transformadas, tais como o JPEG, o H.264/AVC, ou até a mais recente proposta de norma HEVC. O Apêndice F complementa com mais detalhe a descrição do método de filtragem proposto.

No Capítulo 7, é proposta uma nova arquitetura de codificação baseada na exploração conjunta da redundância espaciotemporal. O esquema de compressão apresentado baseia-se na utilização de um esquema preditivo hierárquico tridimensional, sendo o resíduo resultante codificado com recurso a uma extensão volumétrica do algoritmo MMP. São propostos vários modos de predição adaptados ao esquema de codificação proposto. O desempenho do algoritmo desenvolvido é avaliado para a compressão de sinais de vídeo, sendo este potencialmente aplicável a outros tipos de sinais de entrada, como sinais provenientes de radares meteorológicos, ressonâncias magnéticas ou imagens multiespectrais/multivistas. Este capítulo é complementado com mais detalhes apresentados no Apêndice G.

Por fim, o Capítulo 8 irá concluir esta tese e apresentar alguns tópicos para continuação deste trabalho de pesquisa. Este capítulo discute ainda as contribuições dadas em cada um dos tópicos de pesquisa.

Os Apêndice I e J complementam a tese com a apresentação das imagens e de alguns quadros dos vídeos usados nos testes apresentados, e com a lista de publicações resultante deste trabalho.

Capítulo 2

Casamento aproximado de padrões multiescalas: o algoritmo MMP

2.1 Introdução

Os algoritmos de compressão de imagem baseados em casamento de padrões têm vindo a ser alvo de diversas investigações no decorrer das últimas décadas. A sua estratégia passa por dividir o sinal de entrada em segmentos, que são depois aproximados recorrendo a vetores presentes num dicionário. Essa aproximação poderá ser feita seguindo um critério com ou sem perdas. Entre os algoritmos mais conhecidos e usados que se baseiam em casamento de padrões, podemos destacar os algoritmos Lempel-Ziv (LZ) [17–27], e os algoritmos de quantização vetorial (VQ) [28].

Apesar do sucesso que alguns algoritmos baseados em casamento de padrões atingiram para aplicações como compressão sem perdas [29] ou codificação de imagens binárias [30, 31], este paradigma não conseguiu produzir resultados competitivos na compressão com perdas de imagens [32–35] ou vídeo [36–39].

Uma exceção pode no entanto ser encontrada no algoritmo *Multidimensional Multiscale Parser* [2, 3] (MMP). O MMP pode ser visto como uma combinação entre os métodos LZ e a quantização vetorial. O sinal de entrada é decomposto em segmentos, que são aproximados usando elementos de um dicionário, tal como na quantização vetorial, mas esse dicionário é atualizado com segmentos anteriores do sinal de entrada, efetuando-se o casamento para segmentos de dimensão variável, tal como nos métodos LZ.

Adicionalmente, o MMP apresenta uma característica que o distingue dos demais algoritmos baseados em casamento de padrões, e que constitui a base da sua alta capacidade de adaptação às características do sinal de entrada: permite um casamento multiescalas. Em vez de restringir o casamento entre blocos com as mesmas dimensões, o MMP utiliza transformações de escala para permitir o casamento entre blocos com dimensões distintas. Esta ferramenta permite explorar o conceito de auto-similaridade presente nas imagens

naturais, que constitui a base de outros algoritmos de compressão, como os fractais [40].

A adaptatividade do MMP permite-lhe superar o desempenho de algoritmos que compõem o estado da arte para uma vasta gama de aplicações, desde a compressão de imagens naturais com [6] e sem perdas [5], documentos compostos, imagens estereoscópicas [8, 41], sinais de audio [42, 43] ou mesmo eletrocardiogramas [10, 11].

Neste capítulo, serão descritas as características mais importantes do MMP, do ponto de vista da compressão de imagens. No entando, uma descrição mais detalhada do algoritmo poderá ser encontrada no Apêndice B.

2.2 O algoritmo MMP

Tratando-se de um algoritmo que processa o sinal de entrada bloco a bloco, o MMP começa por dividir o sinal de entrada em blocos não sobrepostos, que são processados sequencialmente. Cada um destes blocos é otimizado individualmente, originando uma árvore de segmentação que é posteriormente transformada na sequência de símbolos enviada para o decodificador.

Para cada bloco inicial X^l, pertencente à escala l de dimensões $M \times N$ pixels, o MMP começa por selecionar o elemento S_i^l de um dicionário $\mathcal{D}^l$, que melhor representa o bloco de entrada, segundo um critério de custo definido por:

$$J = D(X^l, S_i^l) + \lambda R(S_i^l), \tag{2.1}$$

onde λ é o multiplicador *lagrangeano* [44] que define o peso da taxa R necessária para representar S_i^l relativamente à distorção D resultante dessa representação.

Depois de identificar o elemento que melhor representa o bloco de entrada na escala l, o algoritmo segmenta esse bloco em duas metades. Aos dois sub-blocos resultantes, X_1^{l-1} e X_2^{l-1}, correspondentes a uma escala inferior, com metade dos pixels do bloco original, é aplicado recursivamente o mesmo procedimento, até se atingirem os blocos elementares de 1×1 pixels (escala 0).

O custo de representação de cada bloco é comparado com a soma dos custos relativos à codificação das duas metades a que dá origem, de modo a decidir se a sua segmentação deverá ou não ser considerada para a sua representação.

Originalmente [3], a segmentação era realizada segundo uma direção pre-estabelecida para cada escala, alternando respectivamente as direções vertical e horizontal. Em [5], foi proposto um novo esquema de segmentação flexível, onde sempre que possível, ambas as direções são testadas, escolhendo-se aquela que resulta no menor custo de representação. A Figura 2.1 representa o número de dimensões distintas para os blocos usando a segmentação flexível, quando comparado com o esquema de segmentação original (a negrito).

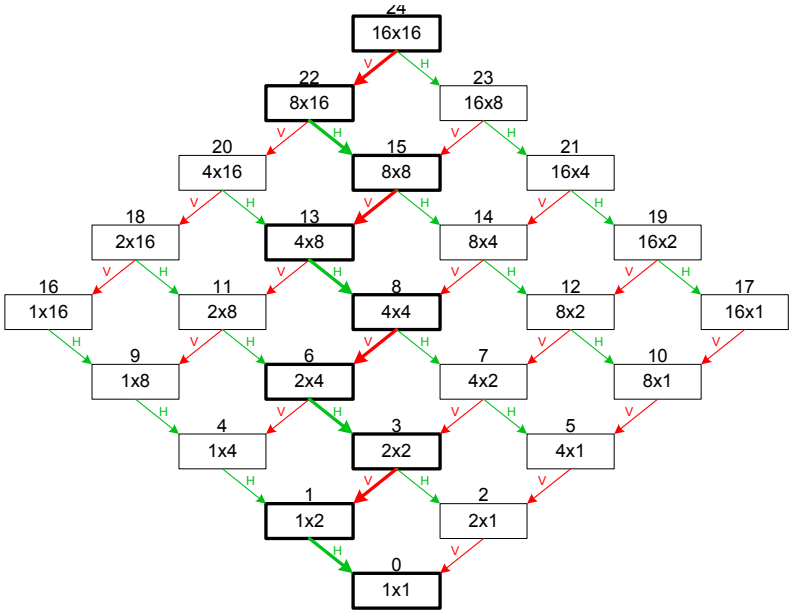


Figura 2.1: Diagrama de escalas para segmentação flexível e para a segmentação original (a negrito).

Este acréscimo no número total de escalas permitiu ao MMP uma melhor adaptação à estrutura da imagem, com ganhos significativos no seu desempenho taxa-distorção.

O padrão ótimo de segmentações para cada bloco é representado por meio de uma árvore de segmentação binária. Cada folha da árvore corresponde a um bloco não segmentado, que será representado por um elemento do dicionário S_i^l, identificado pelo seu índice i. Cada nó n_i^l corresponde a uma segmentação, que corresponde a um bloco representado pela concatenação de 2 sub-blocos. Cada nível da árvore de segmentação tem uma correspondência direta com a escala do bloco à qual diz respeito. No esquema de segmentação flexível, cada nó pode corresponder respectivamente a uma segmentação vertical ou horizontal, se ambas forem definidas para a escala l.

A possibilidade de efetuar casamentos para blocos com dimensões diferentes é uma característica importante do MMP. Através do uso de uma transformação de escalas 2D separável, T_k^l, é possível aproximar um bloco X^l da escala l utilizando um elemento S_i^k da escala k, de dimensões diferentes. Deste modo, um bloco de uma dada escala do dicionário pode ser usado para aproximar blocos de qualquer dimensão.

Em [15], foi proposto combinar o algoritmo MMP com um esquema de predição hierárquico intra-frame. Tal proposta permitiu um acréscimo significativo do desempenho taxa-distorção do MMP na codificação de imagens suaves. O uso da predição tem a particularidade de gerar blocos de resíduo com uma distribuição estatística mais facilmente modelável do que a do sinal original, favorecendo assim a adaptação dos codificadores entrópicos [4]. Com base na vizinhança causal do bloco, é gerado um bloco de predição P_M^l que é depois subtraído ao bloco original, resultando num bloco de resíduo $R_{P_M}^l$, que é codificado usando o MMP, em vez do bloco original.

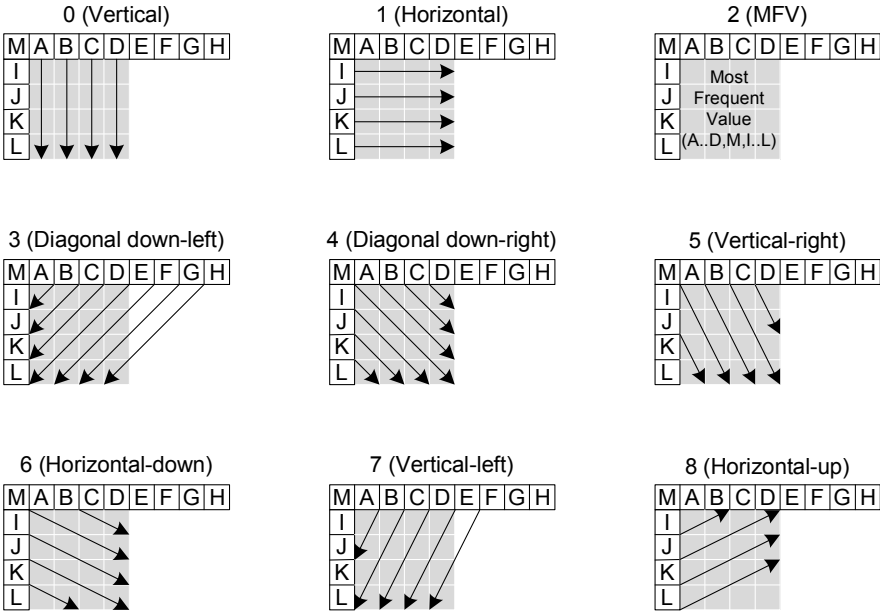


Figura 2.2: Modos de predição utilizados no MMP.

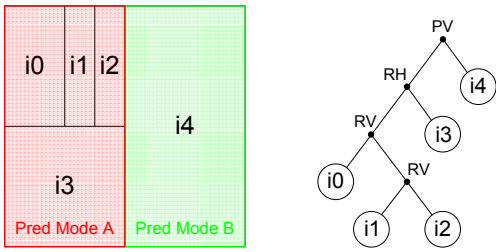
Os modos de predição adotados pelo MMP são semelhantes aos usados pela norma H.264/AVC [45], com apenas algumas exceções. O modo DC foi substituído pelo modo MFV (*Most Frequent Value*), onde o bloco de predição toma o valor que ocorre mais vezes entre os pixels que compõem a vizinhança do bloco, em vez da média do valor desses pixels [15]. A Figura 2.2 representa os modos de predição adotados em [15].

Em [46, 47], foi proposto um modo de predição adicional (LSP), baseado no critério dos mínimos quadrados. Neste modo, a predição para cada pixel é calculada através de uma média ponderada dos pixels vizinhos. Os fatores de ponderação são estimados com base na vizinhança causal do pixel, assumindo que a propriedade de Markov se verifica nessa vizinhança. Mais detalhes relativos a este modo de predição podem ser encontrados em [46].

O uso do esquema de predição hierárquico resulta em dois tipos diferentes de nós na árvore de segmentação, correspondendo respectivamente à segmentação do bloco de predição e do bloco de resíduo, sendo que se considera que sempre que ocorre segmentação da predição, o resíduo é também segmentado.

A Figura 2.3 representa a segmentação de um dado bloco da imagem e a árvore de segmentação correspondente, $\mathcal{T}$. A predição é segmentada em duas metades, sendo o bloco de resíduo da metade da esquerda posteriormente segmentado.

A árvore de segmentação é então convertida num conjunto de símbolos a enviar para o decodificador. A árvore é percorrida de cima para baixo, sendo usadas *flags* para indicar a ocorrência de nós folhas ou segmentações. Neste caso, é necessário discriminar não só a direção da segmentação, bem como se se trata de uma segmentação da predição e resíduo, ou apenas do resíduo. Nos nós terminais do resíduo e predição, são transmitidos respectivamente os índices de dicionário usados para codificar o resíduo, e os modos de



(a) (b)

Figura 2.3: Segmentação de um bloco da imagem (a) e respectiva árvore de segmentação (b).

predição selecionados. Todos os símbolos gerados são codificados com recurso a um codificador aritmético adaptativo [48], com histogramas dependentes da escala à qual dizem respeito.

Uma característica importante do MMP é o fato de utilizar um dicionário adaptativo, que vai sendo atualizado à medida que a codificação prossegue, com recurso à concatenação dos padrões usados para representar os vários nós da árvore. De cada vez que ocorre uma segmentação, os blocos do dicionário usados para representar as metades são concatenados, originando um novo padrão, que sendo inserido no dicionário, passa a estar disponível para a representação de blocos futuros da imagem. Tal procedimento permite ao MMP adaptar-se às características do sinal de entrada. Em [49], são apresentadas algumas técnicas que visaram o incremento da eficiência de aproximação do dicionário usado pelo MMP.

2.3 Resultados experimentais

As Figuras 2.4 e 2.5 apresentam os resultados experimentais do MMP, correspondendo ao PSNR em função da taxa de compressão, quando comparado com o JPEG2000 [50] e o H.264/AVC no seu perfil *high* [45, 51].

As figuras demonstram que o desempenho taxa-distorção do MMP supera o do JPEG2000 e do H.264/AVC tanto para a imagem natural, como para a imagem de texto. Para a imagem Lena, a vantagem do MMP chega aos 1.2 dB. Para o caso da imagem de texto, a vantagem do MMP aumenta consideravelmente, chegando aos 7 dB e 5 dB, respectivamente em relação ao JPEG2000 e ao H.264/AVC. Tal vantagem deve-se ao fato dos codificadores baseados em transformadas assumirem uma compactação da energia do sinal nos coeficientes de baixa-frequência, o que não acontece nestas imagens dadas as transições abruptas nos bordos dos caracteres, que resultam no espalhamento da energia da imagem ao longo de todo o espectro.

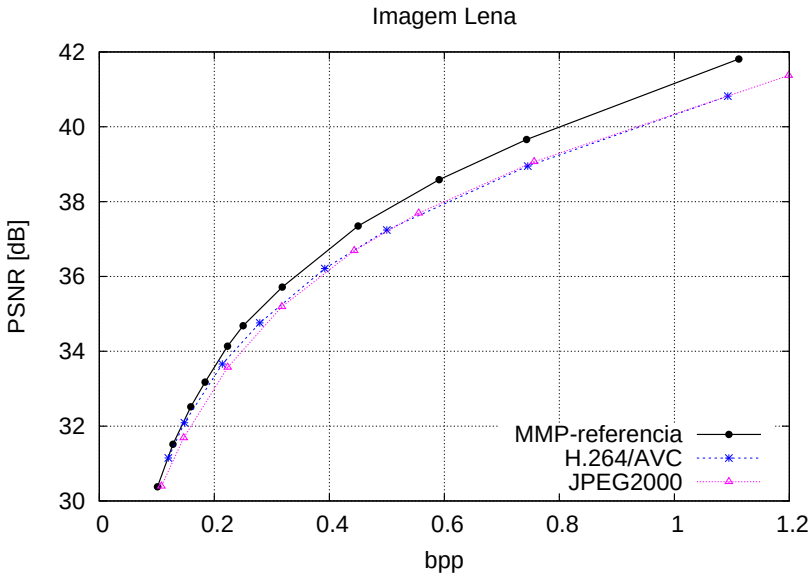


Figura 2.4: Resultados experimentais para imagem natural Lena (512×512).

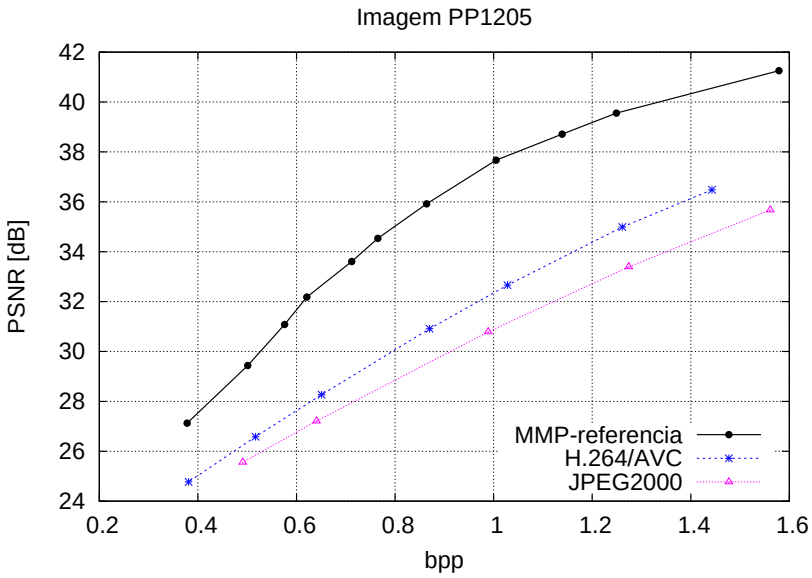


Figura 2.5: Resultados experimentais para imagem de texto PP1205 (512×512).

2.4 Conclusões

Neste capítulo, foi descrito sucintamente o algoritmo MMP, que constitui a base dos esquemas de codificação propostos nesta tese.

A grande adaptabilidade do MMP torna-o adequado para a compressão de um leque alargado de sinais de entrada, incluíndo diversos tipos de imagens, como imagens naturais, imagens de texto ou mesmo imagens sintéticas. O desempenho de compressão do MMP supera o dos codificadores que constituem o estado da arte para a respectiva aplicação.

Mais detalhes e resultados relativos a este algoritmo são apresentados no Apêndice B.

Capítulo 3

Codificação de documentos compostos usando o MMP

3.1 Introdução

A crescente utilização dos suportes multimídia para a transmissão e armazenamento de documentos impõe a necessidade de desenvolvimento de algoritmos de compressão eficientes para este tipo de conteúdos. Os arquivos digitais vão progressivamente substituindo o tradicional suporte de papel, com claras vantagens do ponto de vista do armazenamento e preservação dos documentos, tornando-os acessíveis para um maior número de utilizadores.

A solução mais simples para comprimir este tipo de informação passa pela utilização dos algoritmos de compressão de imagens tradicionais, como o SPIHT [52], o JPEG [53], o JPEG2000 [50] ou o H.264/AVC Intra [45, 54]. No entanto, apesar da sua grande eficiência de compressão para imagens suaves, estes algoritmos não conseguem atingir resultados satisfatórios quando usados para comprimir imagens que apresentam transições de alta frequência, tais como as correspondentes a texto e gráficos, muito comuns nos documentos compostos.

Nas imagens naturais, a maioria dos coeficientes da transformada associados às altas frequências são praticamente negligenciáveis. Tal propriedade permite aplicar a estes coeficientes um passo de quantização elevado ou mesmo descartá-los, sem com isso afetar consideravelmente a qualidade das imagens reconstruídas, o que permite atingir elevadas taxas de compressão. No entanto, quando a imagem apresenta regiões com transições abruptas, os coeficientes de alta frequência deixam de poder ser descartados sem que se introduza um elevado grau de distorção na reconstrução, o que limita a eficiência de compressão.

Uma alternativa a estes métodos poderia passar pela utilização de algoritmos especificamente desenvolvidos para codificar imagens de texto, como o JBIG [55]. No entanto,

estes apresentam sérias limitações quando usados para codificar as regiões suaves presentes nos documentos. Tal desempenho deve-se ao fato das imagens de texto requererem normalmente uma elevada resolução espacial para representar corretamente os caracteres, mas não requererem uma elevada resolução de cor, dado que os caracteres assumem normalmente um número muito limitado de cores. Esta situação é exatamente oposta ao que acontece para as imagens naturais, onde a alta correlação espacial faz com que a imagens não necessitem de uma resolução espacial muito elevada para manter uma qualidade aceitável, mas precisem de uma profundidade de cor elevada para serem satisfatoriamente representadas.

Vários algoritmos como o Digipaper [56], DjVu [57, 58] ou JPEG2000/Part6 [59], entre outros [60, 61], propuseram a adoção do modelo MRC [62] (*Mixed Raster Content*) para decompor a imagem em várias componentes. Uma camada de *background* representa normalmente a componente suave do documento, incluindo as regiões de imagem natural e a textura do papel, enquanto que uma camada de *foreground* contém toda a informação relativa ao formato dos componentes de texto e outros gráficos de alta frequência. A informação presente em ambas as camadas é então combinada com recurso a uma ou várias máscaras de segmentação binárias, podendo todas estas camadas ser normalmente comprimidas de um modo mais eficiente que o documento composto original por si só.

Apesar da grande popularidade destes métodos, o seu desempenho depende da capacidade do algoritmo de segmentação de proceder à correta separação das diversas componentes, o que nem sempre acontece. Em documentos sintéticos, onde as bordas dos caracteres se apresentam bem definidas, a segmentação revela-se relativamente precisa, mas à medida que a complexidade do documento aumenta, aumentam também os erros de segmentação, acabando por comprometer o desempenho geral destes algoritmos.

Neste capítulo, é apresentado um codificador eficiente de documentos compostos digitalizados baseado no MMP. A grande adaptabilidade deste algoritmo permite-lhe ultrapassar algumas limitações apresentadas por outros métodos, o que resulta num esquema de compressão com resultados que constituem o estado da arte para esta aplicação.

3.2 O MMP para codificação de imagens compostas

O esquema de codificação de documentos compostos digitalizados proposto, apelidado de MMP-*compound*, baseia-se na decomposição bloco a bloco do documento nas suas componentes de texto e suave, a serem comprimidas separadamente utilizando variações do algoritmo MMP especificamente otimizadas em função das suas características.

Os métodos de segmentação bloco a bloco [63–68] têm vindo a ser propostos na literatura como uma forma de contornar algumas limitações dos métodos baseados no modelo MRC. Por exemplo, num cenário ideal, as regiões correspondentes à zona mascarada de cada componente não deveriam gerar qualquer tipo de informação adicional, mas na

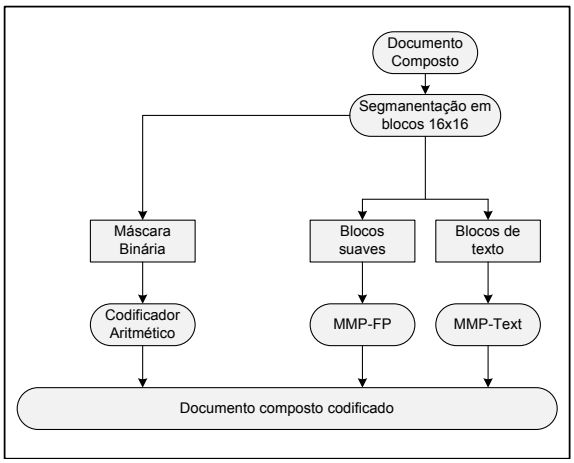


Figura 3.1: Arquitetura do MMP-*compound*.

prática, torna-se necessário preencher essas regiões nas várias camadas antes de proceder à sua codificação. Alguns algoritmos foram propostos para minimizar o custo de transmissão dessa informação redundante [69–71], mas tais métodos apenas fornecem soluções sub-ótimas do ponto de vista do desempenho taxa-distorção.

A Figura 3.1 ilustra a arquitetura do codificador desenvolvido, onde o documento composto digitalizado de entrada começa por ser submetido a um processo de segmentação, descrito em [41], que opera em blocos de 16×16 pixels. Este processo começa por aplicar um filtro *top-hat* e *bottom-hat* [72] à imagem original, com o intuito de atenuar as variações no fundo das regiões de texto, e ainda aumentar o contraste dos objetos do *foreground*. Para esse efeito, é utilizado um elemento estruturante com 7×7 pixels.

Este procedimento resulta em duas imagens processadas: uma gerada pelo operador *bottom-hat*, que permite identificar objectos de *foreground* escuros sobre fundos claros, e outra gerada pelo *top-hat* que permite identificar objetos claros sobre fundos escuros. É então aplicado um classificador de blocos a cada uma das imagens processadas, baseado no método apresentado em [73]. Para tal, é calculado o gradiente vertical e horizontal de cada bloco de 16×16 pixels de cada uma das imagens, e o bloco é classificado como tendo gradiente baixo (inferior a 10), médio (ente 10 e 35) ou alto (superior a 35).

Os blocos correspondentes a zonas de imagem natural tendem a ter gradiente baixos para médios em ambas as direções, enquanto que as regiões textuais tendem a ter gradientes médios a elevados. Os pixels de cada tipo presente no bloco são então contados, e o resultado é usado como entrada do diagrama de fluxo apresentado na Figura C.4, onde Th1 corresponde a 60% e Th2 a 1% dos pixels do bloco.

Este procedimento resulta em duas máscaras de segmentação binárias, que são combinadas através do operador OU lógico, ou seja, cada bloco da máscara final será classificado como texto se tiver sido classificado como texto em pelo menos uma das máscaras binárias. Este procedimento incorre no entanto na classificação errônea de alguns blocos correspondentes a zonas de imagens naturais que possuem elevadas variações, sendo este

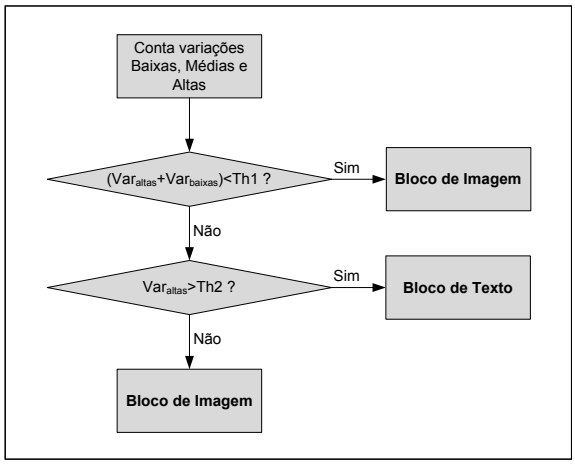


Figura 3.2: Diagrama de fluxo do algoritmo de segmentação.

problema atenuado com recurso a um operador morfológico de deteção de componentes conexas baseado em [74], que se encontra descrito em detalhe em [41].

A máscara binária é então transmitida em primeiro lugar, utilizando um codificador aritmético binário diferencial [48]. Em vez de transmitir diretamente o valor das *flags*, é transmitido o valor 0 sempre que o bloco é do mesmo tipo do anterior, e 1 quando se passa de um tipo de bloco para outro. Tal revelou-se mais eficiente, visto que os blocos do mesmo tipo tendem a ocorrer em grupos. Note-se que o tamanho da máscara é negligível do ponto de vista da imagem codificada, dado que mesmo sem compressão, apenas é necessário transmitir um único bit para cada bloco de 16×16 pixels.

Depois de codificar a máscara binária, o algoritmo efetua a codificação sequencial dos blocos de texto. O esquema de codificação proposto utiliza versões diferentes do MMP respectivamente para as componentes de texto e para as componentes de imagem natural. Tal deve-se ao fato da predição não ser eficiente para a codificação de imagens de texto, resultando frequentemente em blocos de resíduo com uma energia próxima ou mesmo superior à do bloco original. Deste modo, a informação adicional referente à sinalização da predição e respectiva segmentação contribui para que a utilização de um esquema preditivo não seja benéfica do ponto de vista do desempenho taxa-distorção, a somar a uma maior complexidade computacional.

Os blocos de texto são por isso codificados com um algoritmo apelidado de MMP-*text*, uma versão do MMP-FP que não utiliza esquema preditivo. Todos os parâmetros de codificação do algoritmo foram otimizados para operar com blocos de texto e com o esquema não preditivo, nomeadamente a gama dinâmica do dicionário, a distância euclideana mínima entre vetores do dicionário e a super-actualização, que deixou de contar com as simetrias aditivas dos blocos gerados por concatenação.

Após concluir a codificação dos blocos de texto, o algoritmo processa os blocos correpondentes às regiões suaves, que são codificados com recurso ao MMP-FP, descrito no Capítulo 2. Neste caso, os blocos de texto previamente codificados são utilizados como

referência para a predição dos blocos suaves de fronteira. Apesar do uso desta vizinhança poder parecer inapropriado, a verdade é que existe um alto grau de correlação entre os pixels na região de fronteira, ou porque o bloco de texto já contém uma parte que pertence à região suave, ou porque o bloco natural da fronteira ainda contém uma porção do fundo, que será eficientemente predita com base no bloco de texto vizinho.

O uso de dois algoritmos independentes para comprimir as duas componentes tem ainda a vantagem de gerar dois dicionários independentes altamente especializados, o que se revela benéfico do ponto de vista do desempenho de compressão. Os blocos gerados durante a codificação das imagens de texto, que apresentam uma baixa probabilidade de virem a ser utilizados na compressão das regiões suaves, não contribuem para aumentar a entropia dos índices correspondentes aos blocos suaves, e vice-versa. Adicionalmente, o uso de dicionários separados contribui também para a redução da complexidade computacional do algoritmo, já que menos blocos são testados em cada caso.

A segmentação permite ainda aplicar um filtro redutor do efeito de bloco apenas às regiões suaves, sem prejudicar os detalhes de alta frequência existentes nas regiões de texto. Para esse efeito, foi adotado o filtro redutor de efeito de bloco descrito no Capítulo 6.

Mais detalhes sobre o método podem ser encontrados no Apêndice C.

3.3 Resultados experimentais

O desempenho de compressão do algoritmo desenvolvido foi comparado aos de algoritmos que constituem o estado da arte dos codificadores baseados em transformadas: o JPEG2000 [50] e o H.264/AVC no seu perfil *high* [45, 54]. A comparação com o desempenho do H.264/AVC revela-se particularmente interessante, não só pelo excelente desempenho na codificação de imagens suaves [54], mas também pelo fato de ter servido de base ao desenvolvimento de vários esquemas de compressão vocacionados para a codificação de documentos compostos [75, 76].

Adicionalmente, os resultados do método proposto foram também comparados com uma implementação de um método baseado no modelo MRC, o *Lizardtech's Document Express with DjVu - Enterprise Edition* [77].

É importante salientar que para todos os resultados apresentados, o filtro de redução de efeito de bloco do H.264/AVC [78] foi ativado, bem como as ferramentas do DjVu que permitem maximizar a qualidade subjetiva das reconstruções, como o *subsample refinement* e o *background floss*.

Inicialmente, o mesmo valor para o multiplicador de Lagrange λ foi usado para a codificação das componentes de texto e suave. No entanto, verificou-se que a qualidade perceptual da componente texto se apresentava superior à da componente suave, o que motivou o uso de um coeficiente multiplicativo de 0.8 para o λ utilizado na codificação dos blocos de texto. Tal pemitiu uma melhor distribuição da qualidade perceptual em toda

a imagem, com custos no desempenho objetivo praticamente negligenciáveis.

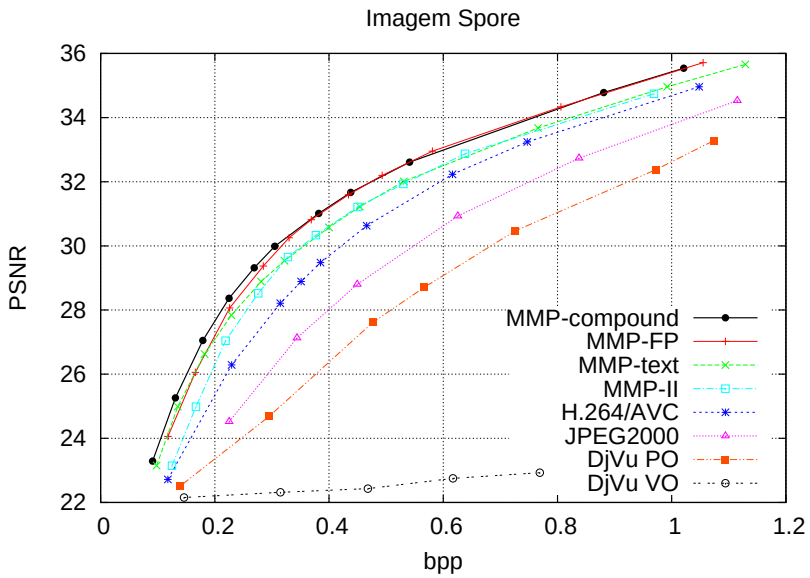


Figura 3.3: Resultados experimentais para o documento composto Spore (1024×1360).

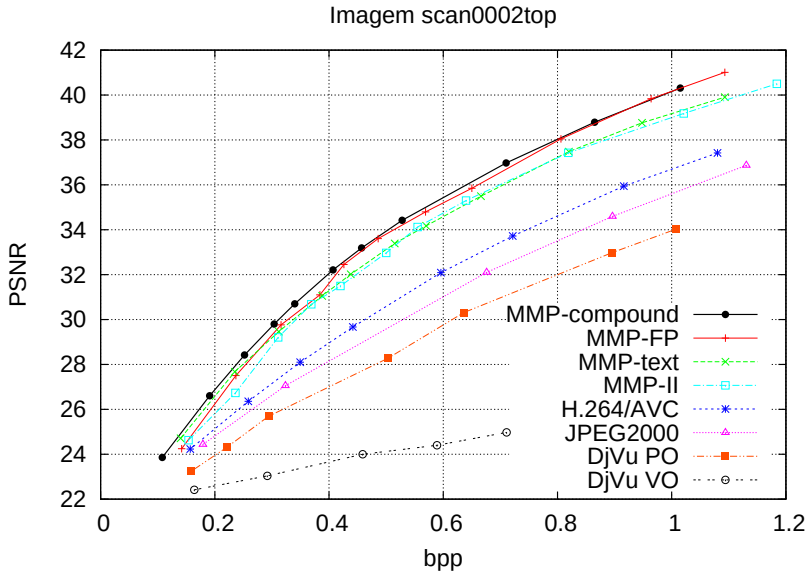


Figura 3.4: Resultados experimentais para o documento composto Scan0002 (512×512).

Os resultados obtidos são apresentados nas Figuras 3.3 e 3.4. Tendo em conta que o DjVu codifica as componente de texto e gráficos como objetos binários, as reconstruções tendem a apresentar uma boa qualidade subjetiva mas um baixo PSNR. Por este motivo, encontram-se representados dois conjuntos de resultados para o DjVu: para os assinalados como DjVu-VO, foi adotado o conjunto de parâmetros que maximiza a qualidade visual das reconstruções, enquanto que para os assinalados como DjVu-PO, foi adotado o conjunto de parâmetros que maximiza a qualidade objetiva da reconstrução.

A Figura 3.5 apresenta alguns detalhes da imagem Scan0002 codificada com os vários

algoritmos, a uma taxa de 0.3 bpp. Note-se que os resultados apresentados para o DjVu contemplam os parâmetros que maximizam a qualidade perceptual da reconstrução.

Nos documentos codificados utilizando o JPEG2000 e o H.264/AVC, são visíveis bastantes artefatos de *ringing* e *blurring* nas zonas de texto, o que compromete a legibilidade do documento para taxas de compressão elevadas. Na reconstrução obtida utilizando o DjVu, a degradação das transições abruptas dos bordos dos caracteres introduzida pelo processo de digitalização provocou a classificação errônea de algumas regiões de texto que ao serem codificadas conjuntamente com a componente suave, aparecem suavizadas e por isso ilegíveis. A reconstrução obtida com o algoritmo proposto não apresenta qualquer problema de legibilidade, mesmo para uma taxa de compressão tão elevada.

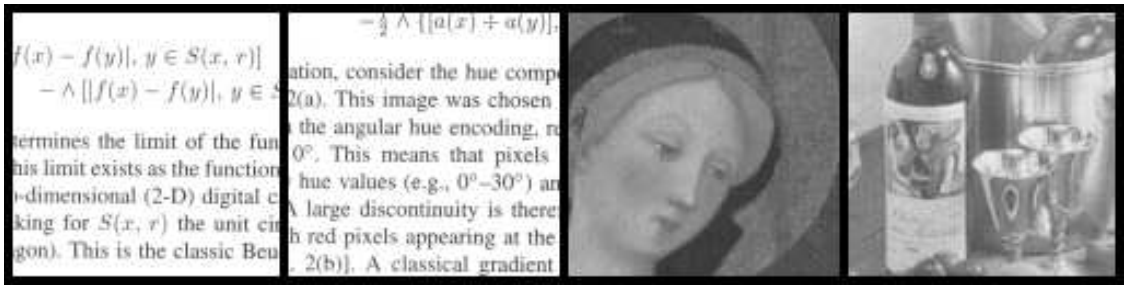
Nas regiões de imagem natural, também é notória a presença de artefatos de *ringing* e *blurring* nas imagens codificadas com o JPEG2000. Estes artefatos também aparecem bem visíveis na reconstrução obtida com o DjVu. Este último apresenta uma classe adicional de artefatos, originados pela classificação incorreta de regiões detalhadas das imagens naturais como *foreground* (por exemplo no rótulo da garrafa da Figura C.15d). As reconstruções obtidas com o método proposto e o H.264/AVC não apresentam nenhum dos artefatos mencionados. Tratando-se ambos de codificadores que processam a imagem bloco a bloco, seria de esperar a introdução de algum efeito de bloco na reconstrução, o que não acontece graças ao filtro de redução do efeito de bloco utilizado por ambos os métodos.

3.4 Conclusões

Neste capítulo, foi brevemente descrito um novo codificador de documentos compostos digitalizados baseado na recorrência de padrões multiescalas. Este algoritmo usa um classificador para decompor o documento nas suas componentes de texto e de imagem suave, sendo cada uma destas componentes codificadas com recurso a uma implementação do algoritmo MMP especificamente otimizada em função das características. O algoritmo MMP-FP, descrito no Capítulo 2, é utilizado para comprimir as regiões correspondendo a imagens naturais, enquanto que o MMP-*text*, uma variante deste algoritmo que não utiliza predição, é usado para comprimir os blocos de texto.

O resultado é um método de compressão de documentos compostos digitalizados robusto, com um desempenho que supera o dos algoritmos que compõem o estado da arte para esta aplicação. Esta robustez advém sobretudo da versatilidade do MMP, que lhe permite adaptar-se eficientemente às características do sinal de entrada, independentemente dos erros de classificação introduzidos na decomposição do documento, ao contrário de algoritmos como o DjVu, cujo desempenho é muito sensível a erros de classificação.

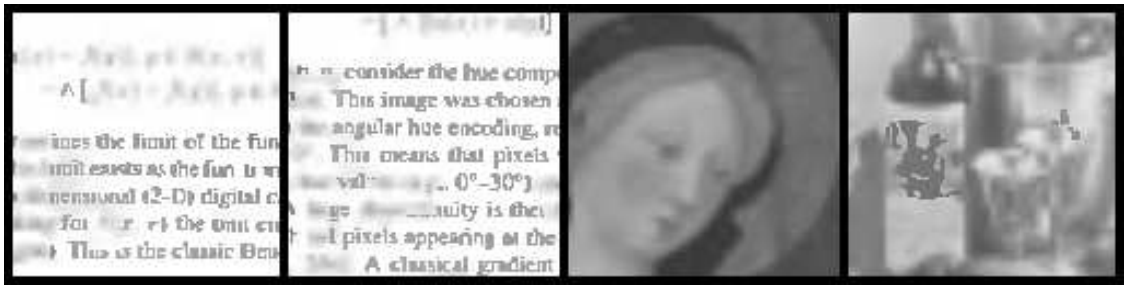
Mais detalhes relativos à implementação e características do MMP-*compound* são apresentados no Apêndice C.



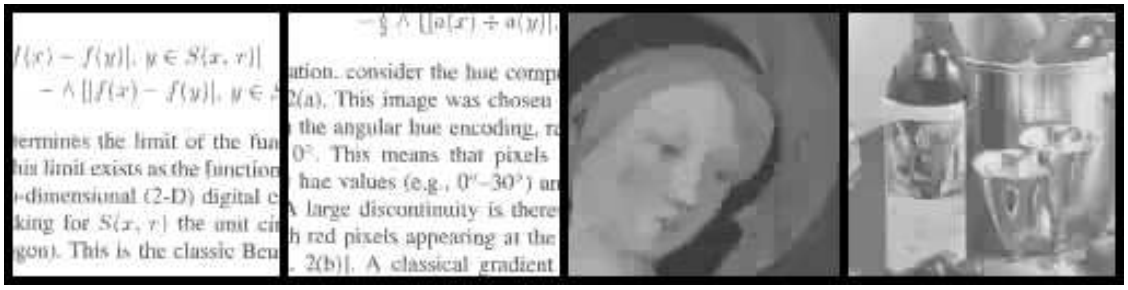
(a) Original a 8bpp

(b) JPEG2000 a 0.30bpp (24.44dB)

(c) H.264/AVC a 0.30bpp (27.11dB)



(d) DjVu a 0.31bpp (23.07dB)



(e) MMP-compound a 0.30bpp (29.98dB)

Figura 3.5: Detalhes da imagem composta Scan0002 a) Original; b) JPEG2000; c) H.264/AVC; d) DjVu; e) MMP-compound.

Capítulo 4

Compressão eficiente de vídeo usando o MMP

4.1 Introdução

Codificadores baseados na arquitetura híbrida têm sido, nas últimas décadas, dominantes na área da compressão de vídeo. Várias normas de sucesso, incluíndo o H.264/AVC [45], se baseiam nesta arquitetura. Esta arquitetura utiliza compensação de movimento e predição *Intra-frame*, para reduzir respetivamente a redundância temporal e espacial das sequências de vídeo, codificando seguidamente os resíduos resultantes recorrendo ao clássico paradigma da transformada, quantização e codificação entrópica. Deste modo, o acréscimo significativo de desempenho que o H.264/AVC [45] apresentou face aos seus antecessores não foi produto de uma mudança de paradigma, mas sim da utilização de um maior leque de ferramentas, que resultaram num algoritmo mais eficiente mas também computacionalmente mais complexo [51].

A elevada eficiência dos codificadores híbridos contribuiu para condicionar a investigação de arquiteturas alternativas para codificação de vídeo. Consequentemente, tal como para o caso das imagens estáticas, não houve muitas propostas de utilização de algoritmos baseados em casamento de padrões para compressão de vídeo. Algumas exceções podem no entanto ser encontradas em [36–39], mas nenhum destes métodos atingiu um desempenho próximo aos dos algoritmos estado da arte baseados no modelo híbrido.

Como referido no Capítulo 2, o MMP apresentou um elevado desempenho de compressão para vários tipos de sinais de entrada, e em [7, 14, 79], foi também proposta a utilização do MMP para a compressão do resíduo resultante da estimação de movimento, com resultados bastante promissores. No entanto, este método manteve a transformada usada pelo H.264/AVC para a codificação do resíduo de predição Intra, devido ao fato do MMP ser nessa altura consideravelmente menos eficiente que o H.264/AVC na codificação dos quadros de referência, o que só por si limitava o desempenho global do codifica-

dor. Mesmo sendo mais eficiente na codificação dos quadros temporalmente estimados, os ganhos aí obtidos não eram suficientes para compensar o pior desempenho na codificação dos quadros de referência, que são responsáveis por uma parte muito significativa da taxa de transmissão.

Com base nestes resultados promissores, um dos objetivos desta tese passava pelo desenvolvimento de um novo codificador de vídeo totalmente suportado pelo paradigma do casamento de padrões, competitivo com o H.264/AVC [45] do ponto de vista do desempenho taxa-distorção. Para tal, os módulos referentes às tranformadas, quantização e codificação entrópica dos coeficientes usados pelo H.264/AVC [45] deveriam ser substituidos pelo MMP [3]. Adicionalmente, o MMP deveria ser otimizado em função das características particulares dos sinais de vídeo, tendo sido introduzidas algumas melhorias que permitiram aumentar a eficiência de codificação para este tipo de sinais.

Neste capítulo, é descrita sucintamente a arquitetura geral do codificador de vídeo desenvolvido, bem como as melhorias introduzidas, que permitiram uma melhor exploração da redundância presente nestes sinais. Uma discussão mais aprofundada é apresentada no Apêndice D.

4.2 Fundamentos de compressão de vídeo

Uma sequência de vídeo é uma sucessão de imagens que apresentam geralmente um elevado grau de correlação espacial e temporal. O sucesso na exploração desta redundância é determinante do ponto de vista do desempenho de um algoritmo de codificação de vídeo.

Uma estratégia comum, usada por muitas normas de codificação, como o H.264/AVC [45], consiste em aplicar predição espacial e temporal às várias imagens, codificando posteriormente o resíduo resultante com o tradicional paradigma da transformada-quantização-codificação entrópica.

Os chamados quadros Intra (I) são codificados usando apenas uma predição espacial gerada a partir da vizinhança causal pertencente à própria imagem. Estes quadros são posteriormente usados para gerar uma predição temporal para os subsequentes (predição Inter), através da estimação de movimento (ME). Objetos se movimentando ao longo da cena aparecem em localizações espaciais diferentes nos vários quadros, pelo que um meio eficiente de os representar passa pela divisão da imagem em blocos, transmitindo para cada bloco um vetor de movimento (MV) que indica a posição do melhor casamento no quadro de referência. Esta abordagem permite substituir a transmissão dos valores de luminância para todos os píxels do bloco por um vetor bidimensional, resultando assim em elevadas taxas de compressão. Adicionalmente, o resíduo pode ainda ser codificado, como forma de reduzir a distorção resultante de variações de luminância ou alteração da forma dos objetos.

A ME pode ser feita utilizando apenas quadros de referência passados, ou também

quadros futuros, num esquema bi-preditivo, se a ordem de codificação dos quadros diferir da ordem de visualização. No primeiro caso, o quadro estimado é referido como quadro P, enquanto que no segundo caso é referido como quadro B. A utilização da bi-predição permite obter geralmente um melhor desempenho de compressão, às custas de uma maior complexidade computacional, visto implicar a necessidade de armazenar e testar mais quadros de referência.

Apesar de todos os quadros previamente codificados poderem ser usados como referências para a estimação de movimento, geralmente apenas os quadros I e P são usados. A forma como estes quadros são codificados é por isso determinante para o desempenho global do codificador, uma vez que a distorção neles introduzida será propagada para os subsequentes, através da ME.

Uma relação interessante pode ser estabelecida entre a estimação de movimento e alguns métodos de casamento de padrões, como os algoritmos LZ. As referências usadas na ME correspondem a porções previamente codificadas do sinal de entrada, tal como no LZ77 [17], sendo o ponteiro para essa informação definido pelo vetor de movimento e o tamanho do casamento implícito através do tamanho do bloco. O tamanho do *buffer* de pesquisa é ele próprio definido pelo número de quadros de referência e pelo tamanho da janela usados. Adicionalmente, a bi-predição permite que a estimação de movimento de um dado segmento do sinal de entrada seja realizada através da combinação de dois segmentos anteriores, numa melhoria relativamente ao LZ77.

Os segmentos de informação existentes nos quadros de referência podem também ser vistos como um dicionário adaptativo composto pela informação previamente codificada, sendo assim possível estabelecer um paralelismo com o LZ78 [18] ou métodos VQ [28]. Neste caso, os MV atuam como o índice que identifica o elemento do dicionário escolhido, que é composto por segmentos selecionados de acordo com a proximidade temporal (através da escolha do número de quadros de referência) e espacial (relacionada com o tamanho da janela de pequisa). O uso de bi-predição pode aqui também ser interpretado como uma extensão do algoritmo LZ78, onde é permitida a utilização de médias ponderadas de dois elementos do dicionário.

4.3 Compressão de vídeo usando casamento de padrões multiescalas - MMP-video

O codificador de vídeo proposto, apelidado de MMP-*video*, baseia-se na arquitetura da norma H.264/AVC, partilhando a mesma estrutura que a sua implementação de referência, o *software* JM [80]. A Figura 4.1 apresenta o diagrama de blocos do codificador, onde os blocos relativos à transformada e quantização foram substituidos pelo MMP. Os restantes módulos são comuns, incluindo a otimização RD [81]. Apesar de não serem

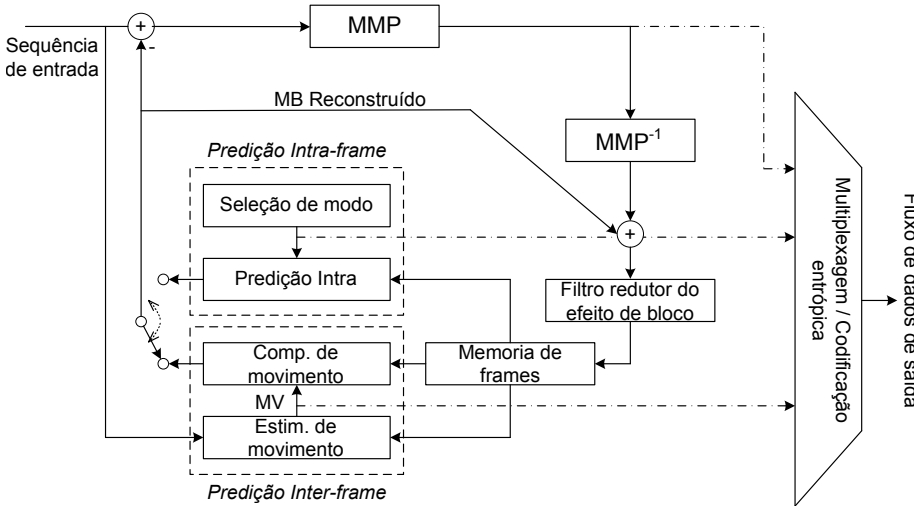


Figura 4.1: Arquitetura do codificador MMP-*video*.

usadas transformadas, existe uma correspondência direta entre o valor do parâmetro QP e do operador lagrangeano λ, usado pelo MMP na otimização RD.

A codificação dos macroblocos (MBs) Intra é uma adaptação direta do MMP-FP descrito no Capítulo 2. Deste modo, o MMP-*video* usa um esquema de predição hierárquico semelhante ao usado originalmente pelo H.264/AVC [45], com algumas modificações, nomeadamente a substituição do modo DC pelo MFV [49] e a introdução do modo LSP [46].

A codificação dos MBs Inter é semelhante à realizada no H.264/AVC, com a exceção da codificação do resíduo. É efetuada uma estimação/compensação de movimento com tamanho de bloco adaptativo (ABS), e o MB de resíduo resultante é codificado utilizando o MMP com segmentação flexível. No entanto, apesar da compensação de movimento com ABS significar que podem ser transmitidos vários vetores para cada MB, foi verificado ser vantajoso não impor a segmentação da predição temporal ao resíduo, otimizando a árvore de segmentação para o bloco de resíduo correspondendo a todo o MB.

A otimização da codificação de um MB implica que o algoritmo teste exaustivamente as várias possibilidades de codificação. No entanto, devido à elevada complexidade computacional do MMP, o custo de codificação do resíduo é estimado com base na sua distorção, tal como acontece no H.264/AVC, quer tendo como base a soma das diferenças absolutas (SAD) ou a soma das diferenças absolutas transformadas (SATD). No entanto, ao contrário do que acontece no H.264/AVC, o bloco de resíduo com menor SAD ou SATD não será necessariamente o que apresentará o menor custo de codificação com o MMP, o que resulta numa solução sub-ótima. Teste experimentais demontraram no entanto que esta abordagem não tem grande impacto no desempenho taxa-distorção do MMP-*video*, apresentando uma redução muito significativa da complexidade computacional, quando comparada com a otimização exaustiva.

Na codificação de quadros Inter, quando a estimação de movimento falha devido a oclusões ou mudanças de cena, o codificador resolve esse problema codificando o MB

em causa como sendo Intra. Essa opção é testada durante o ciclo de otimização e escolhida sempre que o seu custo é menor do que o resultante da sua codificação através da ME, tal como acontece no H.264/AVC [51]. No entanto, de modo a reduzir a complexidade computacional do codificador, o número de modos de predição usado na codificação destes MBs é limitado a apenas quatro modos (MFV, Vertical, Horizontal e LSP).

Para além dos resíduos que são codificados com o MMP, a restante informação é transmitida com recurso às técnicas usadas no H.264/AVC [45], incluindo a informação relativa à compensação de movimento e os cabeçalhos relativos ao quadro e à sequência. As opções do H.264/AVC que dizem respeito à codificação entrópica desta informação também foram mantidas, nomeadamente o uso do CAVLC ou do CABAC [82], dependendo do perfil de compressão utilizado. O filtro de redução do efeito de bloco também foi mantido, visto que resultados experimentais atestaram a sua eficiência também para o MMP-video.

4.3.1 Arquitectura do dicionário para o MMP-*video*

A arquitetura do dicionário foi um importante tópico de investigação para a compressão de imagens estáticas, tendo sido demonstrado possuir elevado impacto no desempenho de compressão do MMP [49]. Adicionalmente, as possibilidades relativas à arquitetura do dicionário aumentam significativamente num cenário de compressão de video a cores, com base na exploração de informação adicional existente neste caso.

A existência de quadros I, P e B, codificados com recurso a ferramentas distintas, resulta na criação de resíduos de características também elas distintas. A descorrelação resultante do uso do espaço de cor YUV também nos permite esperar que o resíduo gerado para cada componente venha a apresentar cacterísticas também elas diferentes. Adicionalmente, dependendo do número de quadros que compõem a sequência de vídeo, o dicionário poderá se beneficiar de um maior período de adaptação, o que também já provou ser benéfico para o desempenho do MMP [4].

Motivadas pelas novas possibilidades relativas ao dicionário e baseadas na experiência adquirida com a compressão de imagens estáticas em tons de cinza [49], foram levadas a cabo diversas experiências com o intuito de estudar o impacto de algumas técnicas de organização do dicionário no desempenho de compressão do MMP-video.

Em [49], foi proposta a utilização de um dicionário único, utilizando no entanto contextos separados em função da escala de origem de cada elemento. Esta solução permitiu explorar a probabilidade condicional dos índices sem com isso limitar a partilha de elementos entre escalas. Tal motivou a investigação de dois tipos de arquitetura para o caso do MMP-*video*, contemplando dicionários independentes por tipo de MB ou componente de cor e a utilização de contextos separados para explorar essa informação adicional.

O melhor desempenho global foi observado com a utilização de dicionários independentes para cada tipo de MB, mas comum às várias componentes de cor. Tal deve-se ao fato da compensação de movimento tender a gerar resíduos com uma energia muito menor que a predição Intra, pelo que os vetores gerados na codificação de um tipo de quadros apresentam uma baixa probabilidade de virem a ser usados em quadros de outro tipo. Esta abordagem tem a vantagem adicional de reduzir a complexidade computacional, dado que menos vetores precisam ser testados em cada casamento. Por outro lado, a utilização de dicionários comuns para as várias componentes de cor apresentou melhores resultados, visto que os dicionários independentes obtidos para as crominâncias apresentavam um crescimento muito limitado e consequentemente um fraco poder de aproximação, fruto da baixa energia que o resíduo resultante para estas componentes tende a apresentar.

As técnicas de controle de redundância e as restrições de escala propostas em [49] foram também elas otimizadas para o MMP-*video*. Verificou-se que o controle de redundância proporciona ganhos consistentes para todos os tipos de sequências, com resultados experimentais a demonstrarem que a regra proposta em [49] também se apresenta adequada para a codificação de sinais de vídeo. A restrição de escalas também demonstrou ter um impacto positivo no desempenho do algoritmo, não só do ponto de vista do desempenho de compressão como também da redução da complexidade computacional.

4.3.2 Uso de um símbolo CBP

Tal como no H.264/AVC, foi adotado um símbolo CBP (*Coded Block Pattern*) no MMP-*video*, para sinalizar a eventual transmissão de informação residual para cada bloco. Esta abordagem permite poupar o envio de índices para os casos em que a predição consegue por si só gerar uma representação eficiente do bloco a codificar.

Utilizando um codificador aritmético adaptativo, é transmitida uma *flag* para cada folha da árvore de segmentação. Caso o bloco de resíduo nulo tenha sido selecionado para representar essa folha, é transmitida a *flag* zero, omitindo-se o índice do dicionário, enquanto que para blocos de resíduo não nulo, é transmitida a *flag* um seguida do índice do dicionário escolhido para o bloco. Apesar de aumentar ligeiramente a taxa requerida para codificar blocos de resíduo não nulo, esta abordagem reduz consideravelmente a taxa dispendida na codificação de resíduos nulos, que ocorrem com frequência nos quadros codificados com ME, onde a predição tende a apresentar uma elevada qualidade.

A abordagem proposta difere da apresentada em [4], onde uma flag CBP era usada para sinalizar a ausência de resíduo mas apenas ao nível do MB. A nova abordagem proposta apresentou melhores resultados, visto que permite uma melhor adaptação às características locais do resíduo. No entanto, resultados experimentais demonstraram que o uso da flag CBP apenas se apresenta vantajoso para os MB Inter, prejudicando a eficiência do método quando usada também para MB Intra. A explicação para esta

constatação encontra-se no impacto global da otimização local. Tendo em conta que a escolha do uso ou não do CBP é feita com base no custo local do bloco, o codificador tende a decidir a favor da menor taxa proporcionada pelo CBP nulo, mesmo considerando a maior distorção da representação, que acaba posteriormente por se propagar aos quadros subsequentes. Por outras palavras, o gasto de alguma taxa adicional na codificação do bloco em causa teria sido vantajoso a longo prazo, visto que teria reduzido não só a distorção do bloco atual, como também de todos os que o utilizam como referência.

Além disso, a flag CBP acaba por limitar a inserção de novos elementos do dicionário próximos do padrão nulo, o que também contribui para reduzir a eficiência do dicionário a longo prazo.

4.4 Resultados experimentais

Com o intuito de comparar o desempenho taxa-distorção do método proposto com a implementação de referência da norma H.264/AVC, na sua versão JM17.1, foram codificadas diversas sequências de vídeo com características distintas. No entanto, verificando-se que a relação entre os resultados observados para as várias sequências se mantêm coerentes, são apresentados os resultados para apenas quatro sequências CIF representativas do conjunto de testes. Mais resultados poderão ser encontrados no Apêndice D.

Os resultados foram obtidos com base num conjunto de parâmetros frequentemente utilizados, nomeadamente um tamanho de GOP de 15 quadros com um padrão *IBBPBBP* a uma frequência de 30 fps. Esta configuração garante a transmissão de 2 quadros I por segundo, resultando num baixo tempo de sincronização para as sequências codificadas.

Foi usado o perfil *high*, otimização RD, e foi ativado o uso de MBs Intra nos quadros Inter. Não foram ativados os métodos de resiliência a erro, nem predição ponderada nos quadros B. Foi usado o CABAC na codificação entrópica dos símbolos gerados, e a estimação de movimentos foi efetuada usando uma janela de ± 16 pixels com 5 quadros de referência, utilizando o algoritmo *Fast Full Search*. As sequências foram codificadas em VBR, utilizando valores de QP diferentes para os quadros I/P e B [83]. Foram usadas quatro combinações de valores: 23-25, 28-30, 33-35 e 38-40.

Na Tabela 4.1, são apresentadas as médias de PSNR para os primeiros 120 quadros das várias sequências CIF, quando codificadas com o MMP-*video* e com o JM17.1. Para salientar a diferença entre o desempenho dos codificadores, foi calculado o delta de Bjøntegaard (BD) [84] para cada componente de cor. Esta medida reflete o ganho médio de PSNR do método proposto em relação ao JM17.1, na gama de sobreposição dos débitos de taxa. Esta medida tem vindo a ganhar uma crescente popularidade visto permitir visualizar de forma clara qual o método que apresenta em média o melhor desempenho.

Os resultados apresentados na Tabela 4.1 demonstram que o método proposto supera globalmente o desempenho de codificação da norma H.264/AVC, sendo os ganhos mais

Tabela 4.1: Comparativo do desempenho taxa-distorção global entre o MMP-*video* e o H.264/AVC JM 17.1. O BD-PSNR corresponde ao ganho de desempenho do MMP-*video* relativamente ao H.264/AVC.

		H.264/AVC				**MMP-Video**				**BD-PSNR**		
	QP [I/P-B]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**Y** [dB]	**U** [dB]	**V** [dB]
Bus	23-25	2223.56	39.07	42.52	44.28	1825.34	38.48	43.14	44.86	**0.54**	**0.47**	**0.50**
	28-30	1126.33	35.03	40.18	41.95	926.81	34.51	40.39	42.14			
	33-35	560.95	31.24	38.52	40.02	482.17	30.97	38.32	39.83			
	38-40	274.56	27.73	37.39	38.61	254.88	27.83	36.79	37.98			
Calendar	23-25	2384.86	38.53	39.32	39.95	2057.89	38.43	40.09	40.57	**0.77**	**0.72**	**0.67**
	28-30	1212.11	34.34	36.15	36.79	1087.08	34.45	36.77	37.32			
	33-35	606.44	30.22	33.46	34.05	559.36	30.54	33.72	34.29			
	38-40	298.52	26.46	31.60	32.17	277.89	26.78	30.71	31.36			
Foreman	23-25	700.09	40.22	43.26	46.08	667.82	40.49	43.90	46.60	**0.33**	**0.14**	**0.20**
	28-30	332.99	37.21	41.18	43.83	314.49	37.33	41.49	44.23			
	33-35	172.23	34.31	39.71	41.77	166.13	34.43	39.50	41.62			
	38-40	94.71	31.46	38.55	39.78	96.08	31.71	37.46	38.56			
Tempete	23-25	2121.89	39.11	40.27	41.84	1756.62	38.52	40.93	42.32	**0.41**	**0.32**	**0.20**
	28-30	897.94	34.66	37.47	39.54	808.16	34.63	37.81	39.68			
	33-35	403.09	31.16	35.31	37.73	390.02	31.39	35.18	37.60			
	38-40	188.55	27.91	33.80	36.39	186.79	28.17	33.05	35.85			

significativos para sequências que apresentam um grau elevado de movimento e detalhes de alta-frequência, como é o caso da Mobile&Calendar. Tal deve-se ao fato de, ao contrário dos métodos que utilizam transformadas, a eficiência de codificação do MMP-*video* não se basear em qualquer pressuposto relativamente às características espectrais das imagens a codificar.

4.5 Conclusões

Neste capítulo, foi apresentado o MMP-*video*, um codificador de vídeo baseado em casamento de padrões recorrentes multiescalas. O algoritmo desenvolvido adotou o MMP para codificar os resíduos resultantes tanto da estimação de movimentos como da predição Intra, subtituíndo por completo o uso das transformadas usadas nos algoritmos que compõem estado da arte. Com a abolição da transformada e quantização, o método proposto é totalmente baseado no paradigma do casamento de padrões, superando o desempenho atingido pela norma H.264/AVC, especialmente para taxas de compressão de médias a baixas.

Foram propostas diversas otimizações funcionais para o MMP, especificamente orientadas em função das características dos sinais de vídeo. Estas otimizações encontram-se descritas com maior detalhe no Apêndice D, onde são igualmente apresentados e discutidos mais resultados.

Capítulo 5

Técnicas de redução da complexidade computational

5.1 Introdução

A elevada adaptabilidade do MMP, verificada nos capítulos anteriores, permitiu-lhe atingir um desempenho de compressão superior aos de algoritmos que definem o estado da arte para vários tipos de aplicações. No entanto, tal como a generalidade dos métodos baseados em casamento de padrões, o MMP apresenta uma elevada complexidade computacional, um fator limitativo para a sua utilização prática na maioria das aplicações.

No caso de aplicações em que o sinal de entrada necessita ser codificado apenas uma vez para ser posteriormente decodificado em múltiplos receptores, uma complexidade computacional elevada do lado do codificador poderá ser justificada pela elevada eficiência de compressão. No entanto, a alta complexidade que o MMP apresenta também no decodificador acaba por tornar o seu uso proibitivo para aplicações que envolvam dispositivos de decodificação com baixos recursos computacionais.

Apesar do acréscimo do desempenho taxa-distorção ter constituído o principal foco de pesquisas anteriores, com a complexidade computacional do MMP sendo relegada para segundo plano, algumas exceções foram apresentadas em [49, 85]. No entanto, dada a natureza sub-ótima da técnica apresentada em [85], a redução da complexidade, que só foi conseguida do lado do codificador, foi atingida à custa de perdas de desempenho taxa-distorção que chegaram a 1 dB.

Neste capítulo, são identificadas as rotinas computacionalmente mais exigentes dos algoritmos de compressão baseados no MMP, sendo propostas duas novas técnicas que visam reduzir a sua complexidade. Estas técnicas têm uma natureza genérica, podendo não só ser usadas para generalidade dos algoritmos baseados em MMP, independentemente da sua aplicação, como também em outros algoritmos baseados em casamento de padrões.

5.2 Novos métodos de redução da complexidade computacional

Nesta seção, são propostas duas novas técnicas de redução da complexidade computacional para algoritmos baseados em casamento de padrões. Estas técnicas foram implementadas e testadas num codificador de imagens baseado no MMP-FP, permitindo reduzir a complexidade das duas rotinas computacionalmente mais exigentes deste algoritmo: a otimização da árvore de segmentação e o processo de busca no dicionário. Uma descrição mais detalhada detas técnicas é apresentada no Apêndice E.

5.2.1 Particionamento do dicionário por norma euclideana

A tarefa computacionalmente mais exigente levada a cabo pelos algoritmos de casamento de padrões em geral, e pelo MMP em particular, é a busca no dicionário pelo melhor vetor para representar cada bloco de entrada. O algoritmo precisa calcular a soma dos erros quadráticos (SSE) entre o bloco a codificar e cada um dos elementos do dicionário, num processo exaustivo e demorado. Para o caso dos algoritmos que utilizam dicionários adaptativos, como é caso do MMP, a atualização do dicionário contribui para aumentar ainda mais o número de buscas a realizar, visto que a inserção de novos padrões no dicionário requer uma busca para verificar a inexistência de outros padrões semelhantes, de modo a evitar redundância.

Uma organização criteriosa dos vetores no dicionário poderá dar uma contribuição importante na aceleração destas buscas. Por exemplo, se os vetores forem dispostos no dicionário seguindo um critério de norma euclideana crescente, a busca pelo elemento que melhor representa o bloco de entrada X^l poderá começar pelos elementos do dicionário com a norma mais próxima de $\|X^l\|$, visto que elementos apresentando uma norma muito distante irão certamente implicar uma grande distorção na representação. Assim, pensando numa otimização apenas baseada na distorção da representação, se o algoritmo começar a busca no dicionário pelos elementos com uma norma próxima de $\|X^l\|$, a menor distorção D encontrada num dado momento permitirá descartar todos os elementos cuja norma se situe fora do intervalo $[\|X^l\| - D; \|X^l\| + D]$, sem necessidade de testar estes elementos.

No entanto, reordenar o dicionário de cada vez que um novo elemento é inserido é uma tarefa também ela computacionalmente exigente, que facilmente anula os ganhos proporcionados pela busca mais eficiente. Tal poderia ser contornado, por exemplo, através de uma indexação com base na norma dos vetores do dicionário. Deste modo, os vetores permaneceriam dispostos arbitrariamente do ponto de vista da sua norma, com o campo de indexação dando a indicação de quais os elementos que necessitam ser testados em cada etapa da otimização. O problema desta abordagem reside no entanto na quanti-

dade de saltos de memória implicados, que demoram também eles bastante tempo a ser executados.

De modo a contornar estas duas limitações, foi desenvolvido um método que combina as duas abordagens discutidas. A gama dinâmica dos valores de norma foi dividida em N segmentos, com os vetores sendo dispostos sequencialmente dentro do seu respetivo segmento. Deste modo, cada segmento pode ser processado sequencialmente, o que minimiza o número de saltos de memória, preservando-se a capacidade de descartar os vetores contidos nos segmentos que correspondem a normas mais distantes em relação à do bloco a codificar. Com esta abordagem, se um casamento perfeito existir para o bloco de entrada X^l, este pertencerá ao segmento n que engloba o valor de norma $\|X^l\|$. Logo, o segmento n será o ponto de partida do processo de busca. No caso da otimização usar um critério de distorção, o melhor elemento deverá forçosamente pertencer ao intervalo $[\|X^l\|-D; \|X^l\|+D]$, onde D representa a distorção incorrida da aproximação de X^l com o melhor casamento obtido até ao momento. Seguidamente, o algoritmo procede para os segmentos $n-k$ e $n+k$, com valores crescentes de k, e de cada vez que se obtém um casamento melhor, o valor de D diminui, e por conseguinte também o intervalo de busca, aumentando assim o número de segmentos que pode ser descartado. Este procedimento acaba por convergir quando todos os segmentos que contemplam as normas contidas no intevalo $[\|X^l\| - D; \|X^l\| + D]$ forem integralmente testados.

A utilização de uma otimização taxa-distorção apenas implica alguns ajustes a esta abordagem. O intervalo de busca deixa neste caso de depender da distorção D para passar a depender do custo lagrangeano J. Consequentemente, a amplitude do intervalo cresce devido ao fator λR, que depende da taxa de compressão alvo. No entanto, um dado vetor situado na fronteira da região de busca poderá apenas constituir um casamento ótimo se for possível representá-lo através de uma taxa nula, o que é obviamente impossível. Sendo assim, o raio da região de busca pode ser reduzido para $J = D + \lambda(\Delta R)$, onde (ΔR) representa a diferença entre a taxa necessária para codificar o índice encontrado até ao momento que representa X^l com o menor custo J, e a taxa mínima requerida para codificar qualquer vetor pertencente a essa escala do dicionário.

Adicionalmente, usa-se um campo de indexação que indica a média de cada um dos elementos do dicionário, o que permite descartar de forma rápida vetores que pertencem ao mesmo segmento de norma, mas se situam noutro quadrante do espaço.

A Figura 5.1 representa a região de busca para um bloco X^l bidimensional. Cada segmento de norma corresponde a uma região concêntrica, sendo S_i^l o melhor casamento encontrado para X^l, dentro do segmento n. Os segmentos $n-1$ e $n+1$ são testados de seguida, com um novo ótimo sendo encontrado em $n-1$. Com esta abordagem, apenas os vetores assinalados como * necessitam ser testados (que correspondem aos segmentos de norma $n-1$, n e $n+1$). Todos os outros vetores pertencentes aos restantes segmentos de norma (representados como x) podem ser imediatamente descartados.

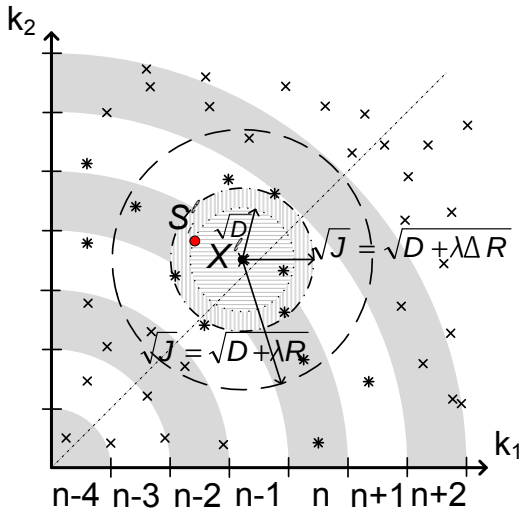


Figura 5.1: Região de busca para um bloco de entrada X^l bidimensional, utilizando um critério de otimização baseado no custo lagrangeano.

O número total de segmentos utilizados apresenta um impacto significativo no desempenho do método. Um elevado número de segmentos é mais eficiente do ponto de vista da redução da região de busca, mas impõe um maior número de saltos de memória. A relação entre o número de vetores testado em cada segmento e o número total de saltos de memória define o valor ótimo para N. Para o caso do MMP, que contempla casamento multiescalas, o valor de N foi otimizado para cada escala l do dicionário. Testes experimentais demonstraram que este valor pode ser satisfatoriamente aproximado através da expressão:

$$N(l) = \left\lceil \frac{\sqrt{GD^2 * Altura(l) * Largura(l)}}{4} \right\rceil , \tag{5.1}$$

onde GD representa a faixa dinâmica do sinal de entrada e $Altura(l)$ e $Largura(l)$ as dimensões dos blocos de escala l.

Foram inicialmente adotados segmentos de capacidade uniforme, somando até a máxima capacidade do dicionário (MDC). No entanto, tendo em conta que vetores necessitam ser descartados sempre que a capacidade máxima do segmento é atingida, esta abordagem apresentou algumas limitações em relação à abordagem tradicional. Os blocos de resíduo apresentam um histograma muito concentrado em torno de zero, pelo que os segmentos correspondentes às normas mais baixas são preenchidos muito mais rapidamente. Consequentemente, o algoritmo passa a ser obrigado a descartar vetores que estariam disponíveis no dicionário não segmentado. Esta restrição do processo de crescimento do dicionário demonstrou ser prejudicial do ponto de vista do desempenho taxa-distorção apresentado pelo MMP.

De modo a minimizar estas perdas de desempenho, a capacidade de cada segmento foi ajustada de acordo com a distribuição típica das normas dos vetores criados. Testes

experimentais demonstraram que quando o crescimento do dicionário não é sujeito a nenhum tipo de restrição, a norma dos vetores gerados tende a presentar uma distribuição que pode ser aproximada através de uma distribuição de Rayleigh, com duas particularidades interessantes. Primeiro, o uso de predição Intra tende a tornar a forma da distribuição praticamente independente das características do sinal de entrada. Segundo, a forma da distribuição depende da taxa de compressão alvo, e consequentemente do valor do operador lagrangeano λ. Baixas distorções correspondem a casos onde a predição é boa, pelo que os blocos de resíduo originados, e consequentemente os vetores criados apresentam uma distribuição de norma mais concentrada em torno de zero.

Com base nestas observações e no resultado dos testes experimentais, foi possível determinar uma expressão que permite calcular a capacidade desejável de cada segmento de norma, dada por:

$$C(n) = a\left(\frac{2n}{b}e^{\frac{-n^2}{b}}\right) + c. \tag{5.2}$$

O valor de b define a concentração da distribuição em torno de zero, e depende por isso de λ, podendo ser definido como:

$$b = \frac{0.2log_{10}(\lambda + 1) + 2}{2}.N(l). \tag{5.3}$$

O valor de c permite definir uma capacidade mínima para cada segmento, capacidade essa que aumenta com λ, pelos mesmos motivos apresentados. Uma relação logarítmica provou fornecer uma boa aproximação da dependência de c em relação a λ:

$$c = \left\lceil MDC\frac{log(\lambda + 1) + 1}{8}\right\rceil. \tag{5.4}$$

O valor de a corresponde ao número restante de elementos, em relação a MDC, resultando na expressão:

$$a = MDC - c.N(l). \tag{5.5}$$

Com esta nova abordagem, foi possível praticamente eliminar as perdas de desempenho taxa-distorção, em troca de uma redução da complexidade computacional um pouco mais modesta, quando comparada à obtida com a utilização de segmentos de capacidade uniforme.

É importante referir que este método permitiu reduzir não só a complexidade do processo de codificação mas também de decodificação, visto que as buscas por vetores similares no processo de atualização se tornam consideravelmente mais rápidas. Neste caso, apenas o segmento que deverá conter o padrão gerado precisa ser pesquisado, ou, no limite, apenas este e alguns segmentos vizinhos, em vez de todo o dicionário. Adicionalmente, este método é aplicável a outros modos baseados em casamento de padrões que implicam buscas no dicionário, tais como os codificadores baseados em VQ [28].

5.2.2 Análise da variação total para expansão da árvore de segmentação

No caso particular do MMP, uma das tarefas responsáveis pela alta complexidade computacional prende-se com a otimização da árvore de segmentação. De modo a determinar a árvore de segmentação ótima para cada bloco de entrada, o MMP precisa levar a cabo uma otimização hierárquica, que corresponde a uma busca por casamento em cada uma das escalas. No entanto, para o caso de blocos que apresentam uma textura muito heterogênea, a probabilidade de que o melhor casamento seja encontrado nas escalas mais elevadas é bastante reduzida.

De modo a evitar que padrões de segmentação com um elevado custo lagrangeano associado sejam testados, a variação total de cada bloco é calculada em ambas as direções, e a sua segmentação é interrompida na direção correspondente sempre que essa variação for inferior a um *threshold* pré-estabelecido τ. Uma relação de dependência pode ser definida entre τ e λ: valores altos de λ implicam um maior peso da taxa no critério de otimização, o que resulta em distorções tendencialmente maiores nas representações, pelo que τ poderá apresentar um valor maior sem comprometer a otimalidade da árvore de segmentação obtida. A expressão:

$$\tau = (0.001\lambda + 1.5) * dimens\tilde{a}o(l), \tag{5.6}$$

demonstrou ser adequada para descrever essa dependência, onde $dimens\tilde{a}o(l)$ representa o número de pixels do bloco numa dada direção. Note-se que esta técnica permite estabelecer um compromisso entre a eventual perda de desempenho de compressão e o ganho em tempo de computação. Definindo um valor de τ muito elevado permite no limite restringir a utilização de multiescalas no algoritmo, tornando-o muito mais rápido mas menos eficiente do ponto de vista do desempenho taxa-distorção.

5.3 Resultados experimentais

De modo a avaliar a redução da complexidade computacional resultante das técnicas propostas, bem como as eventuais perdas de desempenho taxa-distorção delas incorridas, foram codificadas várias imagens com características distintas, utilizando duas versões do MMP-FP semelhantes, com e sem as técnica propostas. A versão do codec que utiliza apenas o particionamento do dicionário em segmentos de norma variável será doravante referida como Enc. I, enquanto que a versão do codec que inclui ambas as técnicas propostas será designada como Enc. II. Considerando que o método da análise de variação total não impõe qualquer modificação ao decodificador, visto ter apenas impacto no ciclo de otimização RD, apenas são apresentados os resultados para o decodificador que

Tabela 5.1: Percentagem de tempo reduzida relativamente ao codificador de referência.

	Rate	0.25 bpp	0.50 bpp	0.75 bpp	1.00 bpp	Average
Enc. I	Lena	46%	53%	55%	57%	53%
	Barbara	51%	61%	69%	69%	63%
	PP1205	63%	72%	79%	84%	75%
	PP1209	50%	65%	66%	65%	62%
	Average	53%	63%	67%	69%	63%
Enc. II	Lena	56%	59%	63%	65%	61%
	Barbara	59%	65%	71%	72%	67%
	PP1205	73%	78%	82%	87%	80%
	PP1209	60%	68%	68%	70%	67%
	Average	62%	68%	71%	74%	69%
Decoder	Lena	73%	80%	84%	84%	80%
	Barbara	87%	81%	85%	87%	85%
	PP1205	94%	94%	92%	94%	94%
	PP1209	91%	87%	89%	90%	89%
	Average	86%	86%	88%	89%	87%

contempla ambas as técnicas.

Os resultados da Tabela 5.1 foram obtidos com um Intel(R) Xeon(R) CPU X5355 @ 2.66 GHz, com dois processadores de quatro núcleos e 8 GB de RAM. É possível verificar reduções médias de respetivamente 69% e 87%, nos tempos de codificação e decodificação, com a utilização das técnicas propostas. No entanto, apesar desta redução de complexidade computacional significativa, apenas foram identificadas perdas marginais de desempenho taxa-distorção, como se pode observar na Figura 5.2.

Para o caso de imagens de texto e compostas, a distribuição estatística da norma do resíduo tende a apresentar-se mais esparsa, dado o fraco desempenho da predição para esses casos, o que prejudica a precisão do modelo apresentado. Para estas imagens, as perdas chegam a rondar os 0.2 dB, para o pior caso. No entanto, esta corresponde também à situação em que a complexidade computacional é mais significativamente reduzida. É ainda importante salientar que mesmo com as perdas marginais apresentadas, o MMP continua a superar consideravelmente o desempenho do H.264/AVC e do JPEG2000.

5.4 Conclusões

Neste capítulo, foram apresentadas duas novas técnicas de redução da complexidade computacional especificamente desenvolvidas para algoritmos de compressão baseados no MMP, mas que podem ser aplicadas à generalidade dos algoritmos baseados em casamento de padrões. Estas técnicas permitiram obter uma redução considerável do tempo de codificação e decodificação, apenas com uma perda marginal de desempenho taxa-distorção.

As técnicas propostas podem ainda ser combinadas com outros métodos previamente

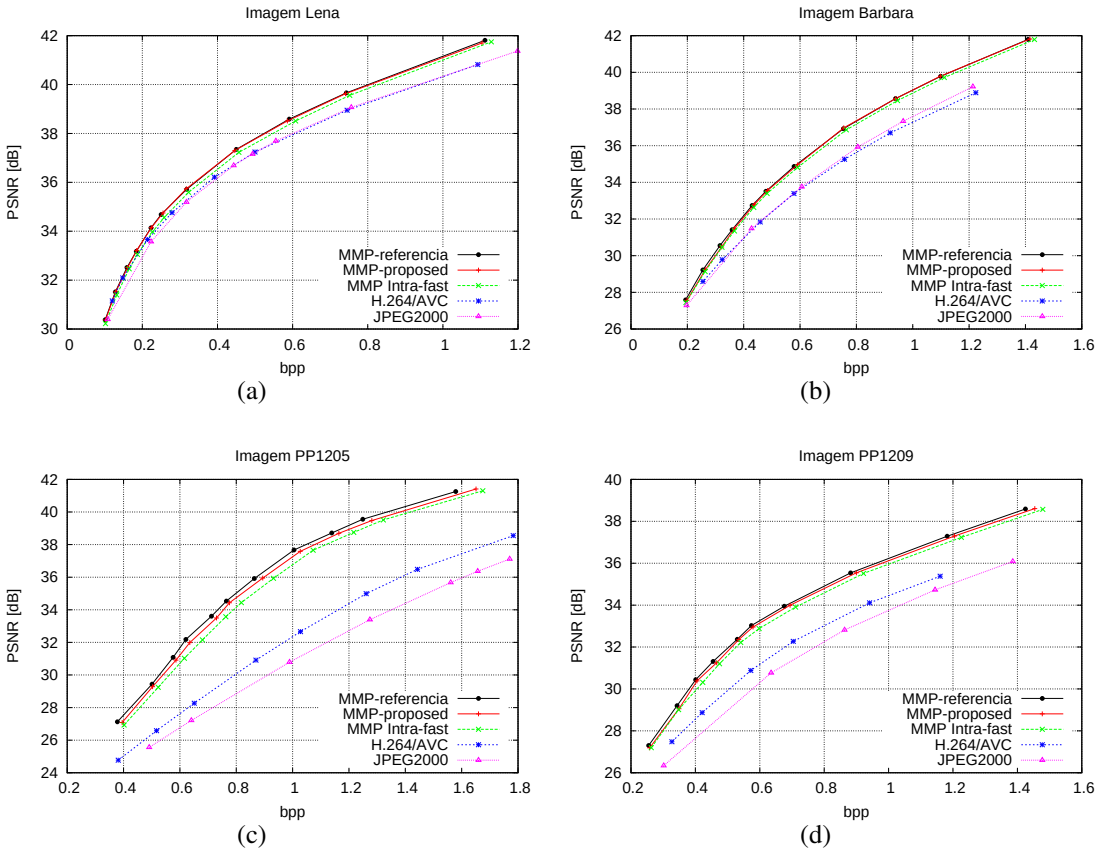


Figura 5.2: Gráficos taxa distorção para as quatro imagens de teste.

propostos, permitindo obter uma redução do tempo de codificação e descodificação que atingem os 90%, em relação à implementação de referência.

Tendo em conta que os algoritmos baseados em MMP apresentam atualmente um desempenho de compressão competitivo com o de algoritmos que constituem o estado-da-arte para um vasto leque de aplicações, a convergência entre a complexidade computacional destes métodos poderá constituir um fator importante na afirmação do casamento de padrões como uma alternativa viável ao paradigma atual da transformada-quantização-codificação entrópica. Apesar do MMP permanecer computacionalmente mais complexo, estas técnicas poderão revelar-se importantes do ponto de vista da aplicação prática do método, principalmente para cenários em que a informação apenas precisa ser codificada uma vez para ser decodificada múltiplas vezes. Neste caso, a complexidade computacional que permanece ainda elevada poderá ser justificada pelo desempenho de compressão superior, comparativamente aos algoritmos que constituem o estado da arte.

Várias melhorias neste campo poderão ser ainda levadas a cabo no futuro, nomeadamente através da paralelização de tarefas repetitivas, recorrendo por exemplo a *hardware* específico, como as GPU (*Graphic Processing Unit*), que sofreram grandes evoluções nos últimos anos. No entanto, essas melhorias referem-se à implementação, enquanto que os métodos propostos neste capítulo dizem respeito à arquitetura do próprio algoritmo.

Capítulo 6

Filtro genérico para redução de efeito de bloco

6.1 Introdução

À semelhança de várias normas de codificação de imagens e vídeo, como o JPEG [53] ou o H.264/AVC [45], o MMP é um algoritmo que processa o sinal de entrada bloco a bloco.

Apesar do elevado desempenho e popularidade apresentada por alguns destes algoritmos, a qualidade perceptual das imagens reconstruidas é frequentemente afetada pelo chamado efeito de bloco, resultante das descontinuidades induzidas nas fronteiras dos blocos, sobretudo a taxas de compressão elevadas. Tal motivou o desenvolvimento de vários métodos, propostos na literatura, que visam reduzir esses artefatos, baseando-se quer em filtragem espacial adaptativa [86, 87], wavelets [88], métodos que atuam no domínio da transformada [89, 90] ou métodos iterativos [91], entre outras propostas.

Alguns destes métodos foram desenvolvidos para funcionar como ferramentas *in-loop*, ou seja, como métodos que atuam diretamente no ciclo de otimização dos algoritmos. O filtro de redução de efeito de bloco adotado pela norma H.264/AVC [78] é um exemplo de um deste métodos, que apresentam no entanto a desvantagem de impor que todos os descodificadores compatíveis devam replicar essa mesma filtragem, de modo a manter o sincronismo com o codificador. Isto retira flexibilidade ao filtro, impedindo que este possa ser ligado e desligado como forma de trocar qualidade visual por uma redução da complexidade computacional.

Visando superar esta limitação, foram propostos alguns métodos de pós-processamento, tais como os descritos em [92, 93]. Neste caso, a filtragem é apenas realizada no final do processo de decodificação, não interferindo com o sincronismo codificador/decodificador. No entanto, estes métodos tendem a ser menos eficientes, visto não terem à sua disposição algumas informações disponíveis no codificador, que facilitam a localização destes artefatos.

Neste capítulo, propomos um novo método de pós-processamento para redução do efeito de bloco, que apresenta um desempenho competitivo com o de métodos *in-loop* que constituem o estado da arte nesta matéria, e que pode ser aplicado tanto a imagens estáticas como sequências de vídeo, comprimidas com recurso a vários codificadores, entre os quais o MMP [3], o H.264/AVC [45], o JPEG [53] ou mesmo a mais recente proposta de norma de codificação de vídeo, o HEVC [16].

6.2 Filtro de redução do efeito de bloco

Nesta seção, iremos descrever o método proposto. Este baseia-se em filtros FIR de resposta variável, que utiliza um número variável de coeficientes. Num primeiro passo, o método constroi um mapa que contém a informação relativa ao comprimento ótimo do filtro a aplicar a cada região da imagem. Seguidamente, é aplicada uma filtragem que utiliza a informação fornecida pelo mapa e parâmetros de forma, estimados em função das características globais da imagem a processar.

6.2.1 Construção do mapa de filtragem

A construção do mapa de filtragem é feita com base na análise da variação total da imagem reconstruída. O método começa por considerar que a imagem se encontra particionada em blocos de dimensão $N \times M$, e para cada bloco, é calculada a variação total respectivamente para as linhas e colunas, através das expressões:

$$\mathcal{A}_j^v = \sum_{i=1}^{N-1} |\hat{\mathbf{X}}_{(i+1,j)} - \hat{\mathbf{X}}_{(i,j)}|, \tag{6.1}$$

$$\mathcal{A}_i^h = \sum_{j=1}^{M-1} |\hat{\mathbf{X}}_{(i,j+1)} - \hat{\mathbf{X}}_{(i,j)}|. \tag{6.2}$$

Cada bloco será então segmentado na direção correspondente sempre que a variação total exceder um dado limiar τ. Deste modo, regiões que apresentam um elevado nível de atividade serão sucessivamente segmentadas, resultando em blocos pequenos, aos quais serão associados filtros com suporte estreito. As regiões com pouca variação, por outro lado, não serão segmentadas, associando-se a estas filtros com suportes largos, que resultarão numa filtragem mais agressiva.

Note-se que o impacto do valor de τ na filtragem final é reduzido, visto que poderá ser compensado com o formato do filtro. Por exemplo, um valor de τ mais baixo originará menos segmentações, o que resultará em blocos com dimensões médias maiores, mas tal poderá ser compensado através da utilização de um formato de filtro que atribua mais peso aos coeficientes correspondentes aos pixels mais próximos, e menos aos mais distantes, o que resultará na diminuição do poder da filtragem. Assim, o valor de τ pode ser fixo,

sem prejuízo para o desempenho final do método, visto apenas se destinar a estabelecer uma relação comparativa da atividade verificada nas várias regiões da imagem. O poder do filtro será então controlado através dos parâmetros que determinam o seu formato.

É igualmente importante salientar que a construção do mapa é realizada apenas com base em informação disponível na imagem, sendo assim independente do algoritmo de codificação que a gerou.

6.2.2 Adaptação dos parâmetros de forma do filtro

Para o método proposto em [94], foram testados diversos formatos de filtro. Resultados experimentais demonstraram a eficiência dos filtros guassianos tanto do ponto de vista do aumento da qualidade objetiva como subjetiva das imagens reconstruídas. Deste modo, os filtros guassianos foram igualmente adotados no método proposto, com o seu suporte estabelecido em $l_k + 1$ amostras, onde l_k refere o tamanho do bloco obtido pelo processo de mapeamento, na respetiva direção. O efeito de filtragem é assim controlado através do ajuste da variância $\sigma^2 = \alpha L$, onde $L = l_k + 1$ é o comprimento do suporte, o que resulta numa resposta ao impulso do filtro dada por:

$$g_L(n) = e^{-\frac{\left(n-\frac{L-1}{2}\right)^2}{2(\alpha L)^2}}, \tag{6.3}$$

com n variando desde 0 a $L-1$. Assim, variando o parâmetro α, é possível ajustar a resposta do filtro desde sendo praticamente rectângular até a um simples impulso (quando α tende para zero), para os casos em que a filtragem não se revela benéfica.

No caso em que a diferença entre os tamanhos dos blocos contíguos implique que o suporte usado na filtragem do bloco maior utilize pixels para além do bloco adjacente, o tamanho do suporte é restringido de modo a apenas usar pixels do bloco adjacente. Esta modificação visou evitar que se utilizem pixels de suporte localizados numa região que se situa para além de uma zona detalhada, o que em algumas situações resultava na introdução de alguns artefatos de filtragem.

Adicionalmente, com o intuito de evitar a filtragem de arestas naturais, o filtro é desativado sempre que a diferença entre a intensidade dos dois pixels que definem a aresta excede um determinado limiar s, analogamente ao que acontece no filtro adaptativo da norma H.264/AVC [78].

Assim, o funcionamento do esquema de filtragem proposto depende de três parâmetros: α, s e τ. Por outro lado, já foi referida a correlação entre os parâmetro τ e α, dado que valores de τ elevados poderão ser compensados diminuindo α, e vice-versa. Consequentemente, o funcionamento do filtro poderá ficar restrito a apenas dois parâmetros.

Em [94], os parâmetros de filtragem eram exaustivamente otimizados de modo a maximizar o PSNR da imagem filtrada, e transmitidos no final do fluxo de dados. Esta abor-

dagem não só implicava um acréscimo marginal na complexidade computacional, como também uma mudança na estrutura do arquivo com a imagem codificada, o que impossibilitava a sua utilização em outros codecs que possuem formatos de arquivo normalizados. De modo a contornar estas limitações, propomos estimar os parâmetros em função da imagem reconstruída.

Testes experimentais demonstraram que o desempenho de método depende muito mais do parâmetro α do que do parâmetro s. Assim, começamos por estudar a relação entre o valor ótimo de α e as características da imagem a processar, fixando para isso $\tau = 32$ e $s = 100$. O parâmetro s foi então posteriormente usado para realizar o ajuste fino do método.

Foi codificado um número elevado de imagens a taxas de compressão diferentes, e usando algoritmos de compressão diferentes, nomeadamente o MMP, o H.264/AVC e o JPEG. Seguidamente, cada uma das imagens reconstruídas foi sujeita a uma filtragem com o método proposto, utilizando diversos valores de α, de modo a determinar aquele que maximiza o PSNR para cada situação. A análise destes resultados e de diversas características estatísticas das imagens permitui determinar uma dependência de α relativamente a essas características. Estas incluíram o tamanho médio do suporte obtido no processo de mapeamento e o desvio padrão desta distribuição, e a variação entre pixels vizinhos e o desvio padrão da distribuição dessas variações. Foi observado que o valor ótimo de α varia de uma forma diretamente proporcional ao tamanho médio dos suportes, e inversamente proporcional à variação média entre pixels, tal como seria de esperar.

Os desvios padrão dessas medidas revelaram-se úteis para caracterizar a distribuição dos detalhes na imagem. Desvios padrão elevados indicam que o detalhe se encontra concentrado numas poucas regiões, enquanto que desvios padrão baixos indicam que estas estatísticas se apresentam homogêneas ao longo de toda a imagem.

O tamanho médio obtido para o suporte demonstrou ser um estimador simples e eficiente para α. Através da criação de um gráfico onde o valor ótimo de α é apresentado em função do produto desta medida calculada em ambas as direções, para todas as imagens de teste, verificamos que a equação:

$$\alpha = 0.0035 \times v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}, \tag{6.4}$$

onde $v_{\text{size}_{\text{avg}}}$ e $h_{\text{size}_{\text{avg}}}$ representam a média dos suportes obtidos respetivamente na vertical e horizontal, permite obter uma boa estimativa de α. Verificou-se ainda que a combinação das características de ambas as direções tende a apresentar um desempenho mais consistente do que a otimização independente em cada uma das direções. De modo a evitar que as imagens sejam excessivamente suavizadas, o valor máximo de α foi limitado a 0.21.

Apesar dos bons resultados obtidos para a generalidade das imagens, o modelo apresentado demonstrou algumas limitações quando aplicado a imagens que contêm um ele-

vado grau de detalhe muito concentrado nalgumas regiões, como é o caso das imagens de texto. Foi então usado o desvio padrão da distribuição do tamanho do bloco para detectar estes casos, e desabilitar o filtro sempre que o produto dos desvios padrões correspondentes a ambas as direções (σ_{size_v} e σ_{size_h}) excede o produto dos tamanhos médios dos blocos numa determinada quantidade: :

$$\frac{\sigma_{\text{size}_v} \times \sigma_{\text{size}_h}}{v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}} > 25. \tag{6.5}$$

O valor de s é então adaptado em função de α, visto que um valor de α elevado identifica a necessidade de aplicar uma filtragem agressiva, enquanto que um valor baixo de α resulta de imagens com elevado detalhe. Assim, e de modo a preservar o detalhe nessas situações, o valor de s deverá também ele ser baixo nesses casos. A equação:

$$s = 50 + 250\alpha, \tag{6.6}$$

demonstrou estabelecer uma relação apropriada entre s e α.

Através do uso das Equações 6.4, 6.5 e 6.6, o método proposto tem a capacidade de se ajustar em função das características da imagem, ou de se ajustar quadro a quadro se usado para processar uma sequência de vídeo.

O impacto do tamanho inicial de bloco usado no processo de mapeamento também foi estudado, sendo que blocos de 16×16 se revelaram os mais vantajosos do ponto de vista da melhoria da qualidade objetiva e subjetiva das imagens reconstruídas. O uso de blocos iniciais grandes não prejudica o desempenho do método, visto estes serem segmentados na presença de zonas de elevado detalhe. No entanto, a sua utilização torna-se muito rara, e quando utilizados, não trazem nenhum acréscimo de desempenho significativo, contribuindo assim apenas para o aumento da complexidade computacional. Por outro lado, a utilização de blocos iniciais menores limita o desempenho máximo do algoritmo, visto limitarem o tamanho máximo do suporte do filtro, e consequentemente, o seu poder de reduzir os artefatos de blocagem.

6.3 Resultados experimentais

O desempenho do método proposto foi avaliado não só para processar imagens estáticas como também sequências de vídeo.

Para o caso das imagens estáticas, o método foi testado em imagens comprimidas utilizando o MMP, o JPEG e o H.264/AVC, com o intuito de demonstrar a sua versatilidade. Os resultados obtidos para as imagens codificadas com os três codificadores são sumarizados na Tabela 6.1.

Para as imagens comprimidas com recurso ao MMP, o método de referência conside-

Tabela 6.1: Comparativo dos resultados obtidos com os vários métodos de filtragem para imagens estáticas [dB].

		MMP				**H.264/AVC**				**JPEG**			
Lena	Rate (bpp)	0.128	0.315	0.442	0.600	0.128	0.260	0.475	0.601	0.16	0.19	0.22	0.25
	Sem filtragem	31.38	35.54	37.11	38.44	31.28	34.48	37.20	38.27	26.46	28.24	29.47	30.41
	Referência	31.49	35.59	37.15	38.45	31.62	34.67	37.24	38.27	**27.83**	**29.55**	**30.61**	**31.42**
	Proposto	**31.67**	**35.68**	**37.21**	**38.48**	**31.63**	**34.72**	**37.31**	**38.31**	27.59	29.32	30.46	31.29
Peppers	Rate (bpp)	0.128	0.291	0.427	0.626	0.144	0.249	0.472	0.677	0.16	0.19	0.22	0.23
	Sem filtragem	31.40	34.68	35.91	37.10	31.62	33.77	35.89	37.09	25.59	27.32	28.39	29.17
	Referência	31.51	34.71	35.92	37.10	**32.02**	33.99	35.90	37.02	**27.33**	**28.99**	**29.89**	**30.54**
	Proposto	**31.73**	**34.77**	**35.95**	**37.11**	31.98	**33.99**	**35.95**	**37.11**	26.64	28.14	29.10	29.74
Barbara	Rate (bpp)	0.197	0.316	0.432	0.574	0.156	0.321	0.407	0.567	0.20	0.25	0.30	0.38
	Sem filtragem	27.26	30.18	32.39	34.43	26.36	29.72	31.13	33.33	23.49	24.49	25.19	26.33
	Referência	27.26	30.18	32.39	34.43	26.54	**29.87**	**31.28**	**33.45**	**24.39**	**25.26**	**25.89**	**26.86**
	Proposto	**27.38**	**30.31**	**32.51**	**34.52**	**26.59**	29.84	31.25	33.42	24.18	25.03	25.52	26.42

rado é o apresentado em [94]. Considerando que o MMP não é um codificador normalizado, os parâmetros do filtro foram exaustivamente otimizados no codificador, do ponto de vista da maximização do PSNR, e transmitidos no arquivo codificado. Estes resultados demonstram que o método proposto proporciona para todos os casos, um maior ganho de PSNR quando comparado ao método apresentado em [94]. O método de mapeamento mais eficiente permite detectar com sucesso as regiões de elevado detalhe, o que permite aplicar uma elevada filtragem às regiões que sofrem de efeito de bloco sem degradar os detalhes originais da imagem.

Os resultados apresentados para o H.264/AVC foram obtidos com recurso ao JM 18.2, ativando e desativando o filtro *in-loop*. O método de referência diz assim respeito aos resultados correspondentes à utilização do método *in-loop* do H.264/AVC [78], e os resultados do método proposto contemplam a aplicação deste às imagens reconstruídas obtidas com o filtro *in-loop* do H.264/AVC desativado. De modo a preservar a concordância com a norma, é utilizada a estimação de parâmetros proposta na obtenção dos resultados apresentados. Os resultados demonstram que o método proposto apresenta resultados que superam os do próprio filtro *in-loop* do H.264/AVC, para algumas situações.

Para os resultados respeitantes ao JPEG, o método de referência diz respeito ao apresentado em [87]. Neste caso, o método proposto não superou o desempenho do método apresentado em [87], mas é importante salientar que este último, ao contrário do proposto, é especifico para o JPEG, utilizando assim muita informação adicional que lhe permite localizar os artefatos introduzidos (o JPEG usa uma transformada de tamanho fixo de 8×8) bem como a sua intensidade (a partir da informação presente na tabela de quantização). Ainda assim, o método proposto revelou-se capaz de melhorar consideravelmente a qualidade das imagens reconstruídas, com resultados próximos dos do método específico.

No Apêndice F, são apresentadas algumas imagens obtidas utilizando os vários mé-

Tabela 6.2: Comparativo dos resultados obtidos com os vários métodos de filtragem para sequências de vídeo [dB].

	H.264/AVC						HEVC					
		In-Loop [78] ON		In-Loop [78] OFF		Proposed		In-Loop [78] ON		In-Loop [78] OFF		Proposed
	QP	Bitrate	PSNR	Bitrate	PSNR	PSNR	QP	Bitrate	PSNR	Bitrate	PSNR	PSNR
	[I/P-B]	[kbps]	[dB]	[kbps]	[dB]	[dB]	[I/P-B]	[kbps]	[dB]	[kbps]	[dB]	[dB]
Rush Hour	48-50	272.46	30.62	288.24	29.99	30.69 (+0.70)	48	214.14	34.41	216.27	34.00	34.27 (+0.26)
	43-45	478.65	33.62	500.79	32.92	33.67 (+0.75)	43	404.23	36.69	410.19	36.25	36.53 (+0.28)
	38-40	865.48	36.38	903.19	35.73	36.42 (+0.69)	38	785.83	38.75	798.08	38.32	38.60 (+0.28)
	33-35	1579.27	38.76	1636.38	38.23	38.80 (+0.57)	33	1655.04	40.67	1683.73	40.28	40.53 (+0.25)
Pedestrian	48-50	409.69	28.68	420.79	28.18	28.75 (+0.56)	48	322.32	32.68	321.74	32.25	32.49 (+0.24)
	43-45	711.64	31.93	730.58	31.43	32.01 (+0.58)	43	576.87	35.20	575.34	34.76	35.00 (+0.24)
	38-40	1216.63	34.89	1243.96	34.44	34.83 (+0.39)	38	1040.16	37.50	1036.70	37.11	37.28 (+0.17)
	33-35	2080.58	37.43	2107.30	37.09	37.17 (+0.08)	33	2003.40	39.68	1995.11	39.36	39.39 (+0.02)
Blue Sky	48-50	572.47	26.74	583.90	26.40	26.71 (+0.30)	48	414.47	32.10	422.10	31.52	31.54 (+0.02)
	43-45	912.33	30.25	924.21	29.99	30.28 (+0.29)	43	715.92	35.01	727.25	34.45	34.44 (-0.01)
	38-40	1557.29	33.77	1566.44	33.54	33.73 (+0.19)	38	1261.90	37.80	1284.25	37.26	37.20 (-0.05)
	33-35	2737.99	37.10	2740.06	36.90	36.82 (-0.08)	38	2327.27	40.41	2369.07	39.95	39.78 (-0.17)

todos, de modo a demonstrar a qualidade perceptual das reconstruções obtidas com o método proposto.

A avaliação do desempenho do método proposto foi também realizada para sequências de vídeo, codificadas com a norma H.264/AVC [51] e com a nova proposta de norma HEVC [16]. O processamento de sequências de vídeo apresenta alguns desafios adicionais. Em primeiro lugar, os artefatos de efeito de bloco podem encontrar-se em qualquer posição dos quadros codificados com recurso a ME, ao contrário do que acontecia para os quadros I, onde os artefatos apareciam ao longo de uma grelha definida pelo tamanho da transformada utilizada. Segundo, o fato da ME ser realizada com base em quadros de referência não filtrados e consequentemente de pior qualidade, degrada a predição temporal e por conseguinte o desempenho global do algoritmo de codificação. Por estes motivos, quando se desliga a filtragem *in-loop*, a obtenção de resultados competitivos implica ganhos maiores no estágio de filtragem, de modo a compensar a perda de eficiência introduzida na estimação de movimento.

Na Tabela 6.2, encontram-se sumarizados os resultados obtidos para os primeiros 128 quadros de três sequências de alta-definição (1920×1080 pixels). Apenas se apresentam os resultados relativos à luminância, visto serem representativos do desempenho global.

Os resultados apresentados para o H.264/AVC foram obtidos recorrendo ao *software* de referência JM18.2, funcionando no perfil *high*, com o filtro redutor de efeito de bloco *in-loop* [78] respectivamente ativo e inativo. A sequência não filtrada foi então pós-processada com o método proposto, usando os parâmetros estimados, conforme descrito na seção anterior. Consequentemente, a taxa de débito apresentada para a sequência não filtrada e a filtrada com o método proposto é a mesma. Foram utilizados parâmetros de codificação usuais, nomeadamente um tamanho de GOP de 15 quadros, com um padrão IBBPBBP, a uma frequência de 25 fps. Para a ME, foi utilizado o algoritmo *Fast Full*

Search, com uma região de busca de 32 pixels e 5 quadros de referência.

Para o caso do HEVC, os resultados foram obtidos recorrendo ao *software* HM5.1, com a maioria dos parâmetros usado por omissão, nomeadamente uma estrutura B hierárquica, com um período de quadros Intra de 8 e um incremento gradual do QP de 1 para cada camada. A ME foi realizada utilizando o algoritmo EPZS, com uma janela de busca de 64 pixels. Note-se que a utilização deste codificador permite avaliar o desempenho do algoritmo para imagens utilizando blocos iniciais de 64×64, em vez dos 16×16 utilizados pelo H.264/AVC e pelo MMP e dos 8×8 utilizados pelo JPEG.

Os resultados apresentados na Tabela 6.2 demonstram que, apesar de não superar a eficiência de compressão obtida ativando os filtros *in-loop*, o método proposto resulta em incrementos consistentes da qualidade das sequências reconstruídas. Tal corrobora a elevada versatilidade do método, que apresentou ganhos significativos de qualidade nas imagens reconstruídas, independentemente das ferramentas utilizadas na sua compressão.

É ainda importante salientar que o desempenho superior dos filtros *in-loop* resulta de alguns inconvenientes práticos, nomeadamente a impossibilidade de ser desativado como forma de trocar qualidade por uma redução da complexidade computacional. Contrariamente, o método proposto apenas impõe que a filtragem seja realizada no decodificador, o que possibilita que esta seja desligada sempre que necessário, sem prejuízo de perda de sincronismo.

6.4 Conclusões

Neste capítulo, foi apresentado um método de pós-processamento para redução de efeito de bloco, baseado em filtros bilaterais adaptativos. O método proposto realiza uma análise da variação total das imagens reconstruídas de modo a elaborar um mapa no qual se define a forma e comprimento do filtro a aplicar a cada zona da imagem. Regiões de baixa variação são filtradas agressivamente, enquanto se assume que regiões que apresentam vaiações maiores possuem um maior grau de detalhe, sendo por isso filtradas mais moderadamente, ou no limite, não filtradas. Esta capacidade de ajuste do grau de suavização proporcionado pelo filtro permite evidar degradar as arestas naturais das imagens, sem com isso deixar de filtrar as regiões que apresentam efeito de bloco.

Ao contrário de outras abordagens, o método proposto é universal, não tendo sido especificamente desenvolvido para funcionar com um único algoritmo de codificação. Tal fica demonstrado pelas melhorias objetivas e perceptuais observadas em imagens codificadas com algoritmos que vão desde o MMP, JPEG, H.264/AVC ou mesmo o HEVC. Tratando-se de um método de pós-processamento, o esquema de filtragem proposto apresenta as vantagens adicionais de não requerer a transmissão de informação adicional, o que o torna compatível com as várias normas, e permite que este seja desativado sempre que necessário, sem o risco de perder o sincronismo entre codificador e decodificador.

Capítulo 7

Compressão de sinais volumétricos utilizando o MMP

7.1 Introdução

Os esquemas de compressão de imagens e vídeo baseados em casamento bidimensional de padrões recorrentes multiescalas foram alvos de uma investigação aprofundada ao longo dos últimos anos. Várias pesquisas culminaram com o desenvolvimento de codificadores que apresentaram resultados que definem o estado da arte para uma vasta gama de aplicações, como foi referido no Capítulo 2. Tal demonstrou o potencial deste paradigma, motivando novas pesquisas por esquemas cada vez mais eficientes e para um número cada vez maior de aplicações. Chegou o tempo de procurar novas abordagens para o MMP, explorando novas ferramentas e arquiteturas de codificação direcionadas aos sinais multimédia. Neste capítulo, é proposta uma nova arquitetura de compressão baseada numa extensão tridimensional do MMP.

Uma adaptação do MMP a blocos tridimensionais já foi anteriormente proposta em [95], num esquema de codificação de sinais provenientes de radares meteorológicos. No entanto, e apesar do elevado desempenho de compressão verificado para esta aplicação particular, o algoritmo proposto em [95] era baseado numa versão obsoleta do MMP, que ainda não dispunha de algumas das técnicas de codificação que permitiram aumentar significativamente o desempenho de compressão do MMP bidimensional. De entre essas técnicas, merecem destaque o uso do esquema preditivo hierárquico [15], a segmentação flexível [5] e algumas técnicas de desenho do dicionário apresentadas em [49].

O principal interesse relativamente ao desenvolvimento de algoritmos de compressão tridimensionais recai na vasta gama de aplicações que poderão beneficiar desta abordagem. Muitos sinais são tridimensionais por natureza, tais como os provenientes de radares meteorológicos, de ecografias ou imagens multiespectrais. Muitos outros, cuja codificação é normalmente feita com base em técnicas bidimensionais, são na realidade também

eles sinais tridimensionais, como é o caso dos sinais de vídeo. Estes não são mais do que uma sucessão de imagens, podendo por isso ser concebidos como sinais tridimensionais, apresentando uma dimensão temporal e duas espaciais.

Como foi referido no Capítulo 4, a maioria dos algoritmos de compressão de vídeo baseiam-se numa arquitetura híbrida, mas vários autores já sugeriram abordar a sua compressão utilizando técnicas inerentemente tridimensionais, como fractais tridimensionais [96], ou extensões tridimensionais da transformada DCT [97–100] e DWT [101–104].

No campo das técnicas que adotaram transformadas tridimensionais, as primeiras propostas sugeriram aplicar diretamente a transformada ao sinal de vídeo [97–99, 101]. Apesar do bom desempenho obtido em sequências com pouco movimento, para as quais a energia se concentra nos coeficientes temporais de baixa frequência, esta abordagem apresenta um fraco desempenho na presença de movimento complexo e não uniforme. Tal motivou o desenvolvimento de outra classe de algoritmos, onde é realizado algum tipo de compensação de movimento antes de aplicar a transformada [100, 102, 104]. No entanto, apesar da excelente relação de desempenho relativamente à complexidade computacional, nenhum destes algoritmos se revelou numa alternativa competitiva aos codificadores híbridos.

Neste capítulo, é proposto um novo algoritmo de compressão de sinais tridimensionais, baseado no MMP e modos de predição tridimensionais, entre os quais, um modo baseado no critério dos mínimos quadrados [46, 47]. O algoritmo proposto destina-se a ser usado para a codificação de diversos sinais tridimensionais, pelo que começamos por descrevê-lo como um método genérico. Seguidamente, são propostas algumas otimizações para adequar o funcionamento do algoritmo à compressão de sinais de vídeo, sendo então avaliado para esta aplicação específica. Uma descrição mais aprofundada do método proposto, assim como uma discussão mais detalhada das várias opções tomadas na sua implementação, são apresentadas no Apêndice G.

7.2 Arquitetura de compressão volumétrica

A arquitetura adotada para o método de compressão proposto utiliza um esquema preditivo hierárquico [15], e uma extensão tridimensional do algoritmo MMP para a codificação do resíduo resultante, operando com um esquema de segmentação flexível tridimensional [5].

7.2.1 3D-MMP

Em [95], a utilização de uma extensão tridimensional do MMP foi proposta para codificação de sinais provenientes de radares meteorológicos. No entanto, o algoritmo proposto

em [95] apresenta algumas diferenças relativamente aos objetivos definidos para esta tese, visto se basear numa versão mais antiga do MMP, que não dispunha de algumas das técnicas que permitiram aumentar o seu desempenho de compressão. Deste modo, pretende-se começar por estudar a influência dessas técnicas numa arquitetura tridimensional, que incluem a segmentação flexível [5] ou as técnicas de desenho eficiente do dicionário propostas em [49].

Quando comparado ao MMP bidimensional, descrito no Capítulo 2, a primeira grande diferença reside no fato das unidades básicas deixarem de ser blocos rectângulares $X^l_{m,n}$ com $N \times M$ pixels, para passarem a ser paralelepípedos $X^l_{m,n,o}$ com $N \times M \times K$ pixels. Como consequência direta, ocorre um aumento das possibilidades de segmentação dos blocos. Um bloco genérico de escala l com $N \times M \times K$ pixels, onde $N \neq 1$, $M \neq 1$ e $K \neq 1$, passa a poder ser dividido segundo cada um dos três eixos que definem o espaço.

Esta alteração tem implicação direta no número total de *flags* utilizadas para indicar a segmentação do bloco de predição ou dos blocos de predição e resíduo segundo o eixo adicional. Passa assim a existir um total de sete *flags* de segmentação, indicando a não segmentação do bloco, e respectivamente a segmentação apenas do resíduo e simultaneamente do resíduo e da predição segundo os três eixos agora existentes.

O aumento de possibilidades de segmentação tem igualmente impacto no número total de escalas, que passa a ser definido pela expressão:

$$N_{scales} = (1 + \log N) \times (1 + \log M) \times (1 + \log K), \tag{7.1}$$

Onde N, M e K deverão ser potências de dois, definindo o tamanho inicial do bloco usado pelo MMP.

Este aumento no número de escalas tem um impacto significativo em vários aspetos do MMP, como por exemplo o codificador aritmético, que utiliza histogramas independentes para cada escala do dicionário, e a limitação de escalas de inserção de novos elementos no dicionário. Adicionalmente, o número de escalas influi na complexidade computacional, visto que o aumento das possibilidades de segmentação aumenta também o número de buscas a realizar. Estes aspetos são discutidos com maior detalhe no Apêndice G.

7.2.2 Predição tridimensional

Foram desenvolvidos alguns modos de predição tridimensionais, de modo a explorar a redundância existente ao longo das várias dimensões. Um destes modos baseia-se no critério dos mínimos quadrados, proposto em [47], e que foi adaptado para um funcionamento bloco a bloco em [46]. Quando a imagem de entrada é processada bloco a bloco, alguns pixels da vizinhança que pertencem ao mesmo bloco que o pixel que está sendo predito ainda não se encontram totalmente codificados, ao contrário do que acontecia em [47]. De modo a contornar esta não causalidade, foi proposto em [46] utilizar os va-

lores preditos para estes pixels, tanto no suporte como na área de treino do filtro usado na predição. Adicionalmente, na fronteira direita do bloco, os pixels da linha acima situados à direita do que está sendo predito, pertencem ao bloco seguinte e como tal, ainda não se encontram disponíveis. Foi proposta uma modificação tanto ao suporte como à área de treino para solucionar esta situação, que se encontra ilustrada nas Figuras B.6-b e B.7-b.

No entanto, os problemas de causalidade inerentes ao processamento da informação de entrada bloco a bloco ficam ainda mais evidentes num esquema tridimensional. O método proposto em [47] utiliza pixels do quadro anterior para gerar uma predição espaciotemporal, no que pode ser visto como um caso genérico de um esquema de compressão tridimensional em que o eixo temporal define a terceira dimensão. No entanto, tendo em conta que o algoritmo apresentado em [47] processa a sequência de vídeo quadro a quadro, os pixels do quadro anterior se encontram sempre disponíveis quando um dado pixel é predito, o que nem sempre acontece numa arquitetura tridimensional bloco a bloco. Por este motivo, propomos estender as adaptações apresentadas em [46] à dimensão adicional, quer através do uso dos valores preditos para os pixels ainda não codificados, necessários ao suporte do filtro, quer adaptando o suporte e a área de treino de acordo com a causalidade verificada em cada caso.

O suporte usado por omissão é apresentado na Figura 7.1a, sendo semelhante ao usado em [47], que utiliza quatro vizinhos espaciais e nove temporais. Os suportes apresentados nas Figuras 7.1b e 7.1c são usados no limite direito do bloco, respectivamente para a primeira e para as restantes camadas do bloco. Na primeira camada do bloco, os vizinhos temporais pertencem ao bloco anterior, estando por isso disponíveis. Nas camadas seguintes, não existem vizinhos temporais à direita do pixel que está sendo codificado, uma vez que pertencem ao bloco vizinho. Neste caso, as referências temporais são deslocadas para a esquerda. Uma situação similar ocorre nos pixels da última linha do bloco, onde não existem referências abaixo do pixel que está sendo codificado. Nesses caso, as referências temporais são delocadas para cima, como ilustrado nas Figuras 7.1d e 7.1e. Para os pixels da primeira camada do sinal de entrada, a ordem do preditor é reduzida a quatro, compreendendo apenas as referências temporais.

A predição para $X(\vec{n}_0)$, onde $\vec{n}_0 = (k_1, k_2, k_3)$, é então calculada através da expressão:

$$\hat{X}(\vec{n}_0) = \sum_{i=1}^{N} a_i X(\vec{n}_i), \tag{7.2}$$

onde $\vec{n}_i$ são os pixels que compõem a vizinhança apresentada na Figura 7.1, e $\vec{a} = [a_1, ..., a_n]^T$ são os coeficientes que minimizam o erro de predição na vizinhança de treino. A vizinhança de treino proposta é um extensão tridimensional da apresentada em [46]. A Figura G.3a apresenta o caso genérico, enquanto que a Figura G.3b representa a vizinhança usada para pixels na fronteira direita do bloco ou do sinal de entrada, para os quais não existem vizinhos para a direita, relativamente a $\vec{n}_0$. Note-se que para a primeira ca-

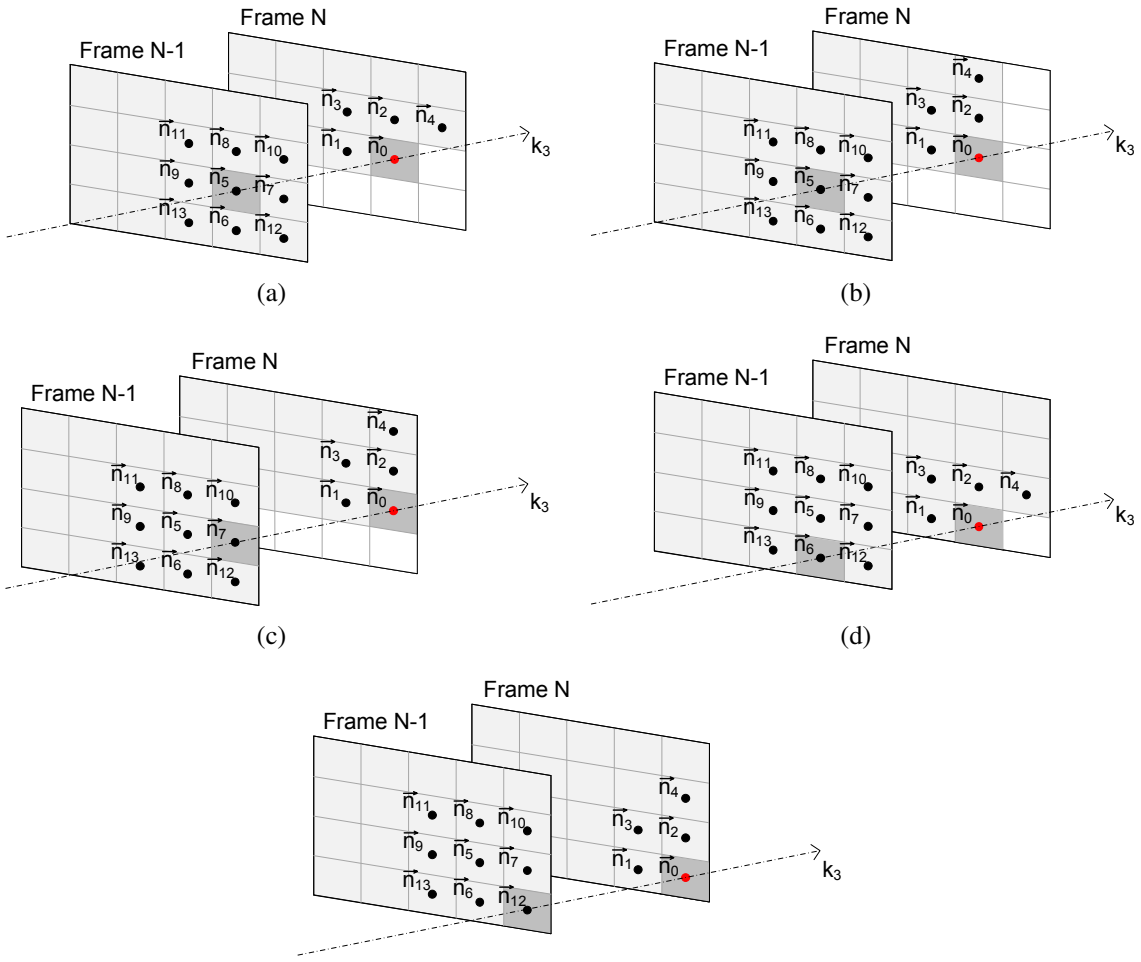


Figura 7.1: Vizinhança tridimensional usada (a) por omissão (b) coluna da direita (c) linha de baixo (d) canto inferior direito.

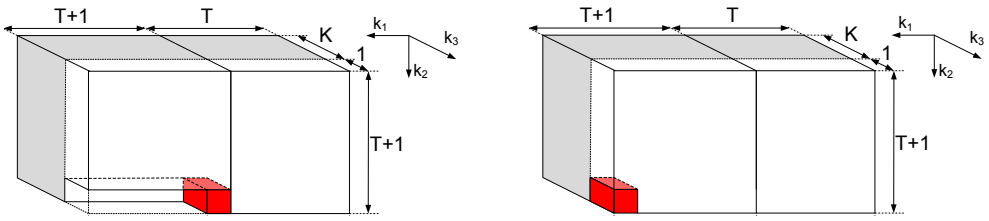
mada, é usado $K = 1$, incrementando até ao valor máximo definido à medida que a codificação do sinal procede. Os M pixels da região de treino são usados para formar um vetor coluna $\vec{y}$ de dimensão $M \times 1$. Se colocarmos os N vizinhos que compõem o suporte do filtro num vetor linha $1 \times N$, é possível formar uma matriz C, de dimensões $M \times N$. Os coeficientes de predição $\vec{a}$ podem então ser obtidos resolvendo:

$$\min(||\vec{y} - C\vec{a}||^2), \tag{7.3}$$

que tem como solução:

$$\vec{a} = (C^T C)^{-1}(C^T \vec{y}). \tag{7.4}$$

Adicionalmente, foi desenvolvido um novo modo de predição direcional, adaptado à nova arquitetura tridimensional. Os modos direcionais bidimensionais foram explorados com sucesso em codificadores híbridos, como o H.264/AVC [51, 51], e mais recentemente no HEVC [16].



(a) (b)

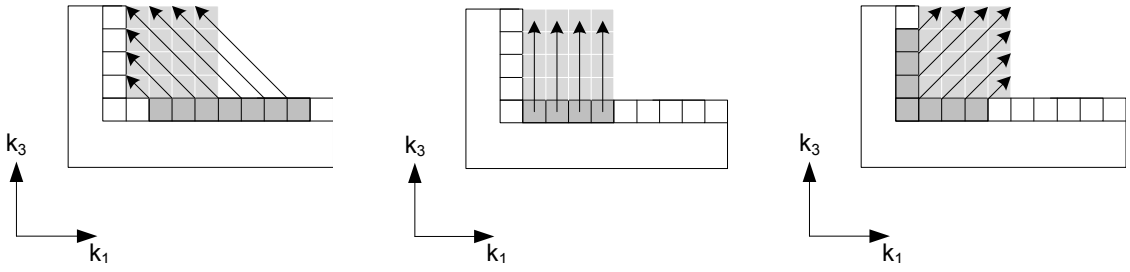
Figura 7.2: Vizinhança de treino usada (a) por omissão (b) coluna da direita do bloco.

Para o caso de blocos tridimensionais, considerando um bloco genérico $X^l(k_1, k_2, k_3)$ com $N \times M \times K$ pixels, uma predição para cada camada ao longo de k_3 pode ser gerada através da expressão:

$$\hat{X}^l(k_1, k_2, k_3) = X^l(k_1 - v_1, k_2 - v_2, k_3 - 1), \tag{7.5}$$

onde v_1 e v_2 são componentes de um vetor direcional $\vec{v}$ bidimensional. Esta predição pode ser encarada como uma generalização da ME usada nos codificadores de vídeo híbridos, em que vários quadros podem ser preditos através do mesmo vetor.

O esquema de codificação bloco a bloco impede no entanto que se use simplesmente uma versão deslocada da camada anterior, dado que esta poderá não ser causal. Assim, são sempre usados os pixels mais próximos para gerar a predição. A Figura 7.3 ilustra a predição direcional gerada ao longo de uma única coordenada, para facilitar a visualização, bem como a respectiva referência usada. Inicialmente, todas as direções



(a) (b) (c)

Figura 7.3: Predição direcional ao longo de uma coordenada (a) $v_1 < 0$ (b) $v_1 = 0$ (c) $v_1 > 0$.

contempladas eram exaustivamente testadas, escolhendo-se a que minimizava a energia do resíduo gerado. Note-se que esta abordagem nem sempre corresponde ao menor custo de codificação do resíduo por parte do MMP, mas permite reduzir muito significativamente a complexidade computacional, sem comprometer o desempenho de compressão. O vetor direcional obtido era então codificado com recurso a um codificador aritmético adaptativo, usando como contexto a escala do bloco que estava sendo predito.

No entanto, esta abordagem não tirava partido da correlação do vetor com o compor-

tamento da vizinhança do bloco. Como tal, o vetor direcional passou a ser estimado com base no comportamento dos blocos vizinhos, escolhendo-se entre o vetor estimado e o calculado com base no custo lagrangeano por eles apresentados. Caso o vetor estimado seja escolhido, é apenas transmitida a *flag* 0, sendo que o decodificador consegue estimar o mesmo vetor com base na mesma vizinhança causal decodificada. Caso contrário, é enviada a *flag* 1, seguindo-se do vetor direcional calculado.

Adicionalmente, foram adotados os mesmos modos de predição do H.264/AVC [51, 51], apresentados na Figura 2.2, aplicados camada a camada, segundo a coordenada k_3.

7.3 3D-MMP para compressão de vídeo

A codificação de sinais de vídeo começou por ser testada processando a informação sequencialmente em blocos $N \times N \times N$. Tal corresponde a definir k_1 e k_2 como as coordenadas espaciais, e atribuir o eixo temporal a k_3. A semelhança entre cada conjunto de N quadros e o GOP definido nos codificadores híbridos torna-se notória, sendo que ambos correspondem à mínima unidade temporal que pode ser descodificada independentemente.

Tal levou-nos a uma segunda abordagem, na qual, em vez de codificar os quadros sequencialmente, estes passam a ser codificados alternadamente, como ilustrado na Figura 7.4.

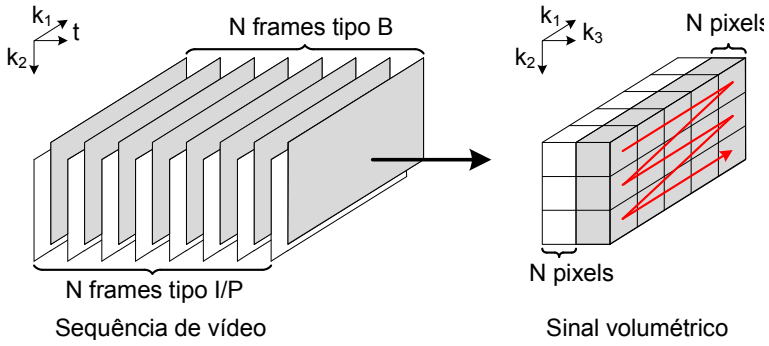


Figura 7.4: Arquitetura hierárqica para codificação de vídeo.

Neste caso, começa-se por retirar um grupo de quadros alternados, que sendo codificado em primeiro lugar, será usado para efetuar uma predição bidirecional para os quadros intermédios, codificados num segundo grupo. Esta abordagem assemelha-se aos quadros I/P e B dos codificadores híbridos, e permitiu obter um acréscimo de desempenho para a codificação de sinais de vídeo.

Com esta abordagem, foi possível alterar a predição baseado no critério dos mínimos quadrados, de modo a incluir pixels do quadro futuro no suporte do filtro, tornando este num modo de predição implícito bidirecional. Para tal, foram incluídos no suporte nove

pixels do quadro futuro, situados em posições espaciais equivalentes às dos pixels do quadro passado.

Do mesmo modo, as referências intermédias passaram a ser usadas no modo de predição direcional, de modo a tirar proveito das referências mais próximas agora existentes, como ilustrado na Figura 7.5.

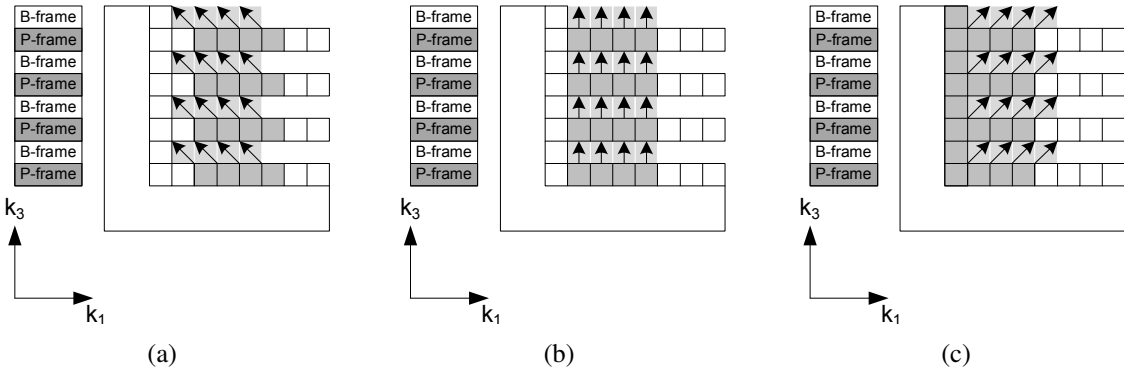


Figura 7.5: Predição direcional ao longo de uma coordenada para os quadros tipo B (a) $v_1 < 0$ (b) $v_1 = 0$ (c) $v_1 > 0$.

7.4 Resultados experimentais

O desempenho de compressão do método proposto para a codificação de sinais de vídeo foi avaliado por comparação com a versão JM17.1 da norma H.264/AVC.

O mesmo conjunto de parâmetros de configuração usados no H.264/AVC para gerar os resultados apresentados no Capítulo 4 foi igualmente usado para obter os resultados apresentados neste capítulo, incluindo o uso do perfil *high* com otimização RD, o tamanho de GOP de 15 com um padrão *IBBPBBP* a uma frequência de 30 fps, e o uso de MBs Intra nos quadros Inter. As ferramentas de resiliência a erro foram desativadas, assim como a predição ponderada nos quadros B. A codificação entrópica foi deixada a cargo do CABAC, e a ME utilizou o algoritmo *Fast Full Search*, com uma janela de ± 16 pixels e 5 quadros de referência. As sequências foram codificadas em VBR, utilizando os mesmos valores de QP usados no Capítulo 4, sendo eles 23-25, 28-30, 33-35 e 38-40.

Para o 3D-MMP, foram usados blocos com $8 \times 8 \times 8$ pixels, de modo a limitar a complexidade computacional. Foi utilizado o esquema de codificação hierárquica, alternando um quadro B e um quadro I/P, sequencialmente. Tendo em conta que, tal como no H.264/AVC, é vantajoso codificar os quadros de referência com uma menor distorção [83], o valor de λ usado nos blocos B foi definido como sendo 50% maior que o usado nos P. De modo a obter uma gama de taxas equivalente à obtida com os valores de QP usados no H.264/AVC, foram usado quatro pares de valores de λ, nomeadamente 20-30, 75-112, 200-300 e 500-750, respectivamente para os quadros I/P e B. Foi utilizado

Tabela 7.1: Comparativo do desempenho taxa-distorção global entre o 3D-MMP e o H.264/AVC JM 17.1. O BD-PSNR corresponde ao ganho de desempenho do 3D-MMP relativamente ao H.264/AVC.

		H.264/AVC				**3D-MMP**				**BD-PSNR**		
	QP [I/P-B]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**Y** [dB]	**U** [dB]	**V** [dB]
Akiyo	23-25	256.12	43.39	45.46	46.68	272.66	42.95	46.01	47.22	**-0.91**	**0.27**	**0.42**
	28-30	140.84	40.64	42.69	44.12	144.35	39.81	43.16	44.77			
	33-35	81.06	37.65	40.07	41.82	98.56	37.59	41.28	43.00			
	38-40	48.22	34.47	38.29	40.47	58.69	35.23	38.71	40.97			
Coastguard	23-25	2335.07	38.78	45.79	46.88	2220.32	36.09	45.13	46.26	**-2.03**	**-0.17**	**-0.14**
	28-30	987.19	34.19	44.16	45.10	969.46	31.97	43.72	44.65			
	33-35	431.83	31.11	42.59	43.49	507.94	29.54	42.80	43.76			
	38-40	172.47	28.34	40.50	41.31	208.84	27.66	41.61	42.50			
Container	23-25	576.35	40.38	45.00	45.09	505.28	39.92	45.65	45.73	**0.17**	**1.05**	**0.96**
	28-30	286.29	37.02	42.26	42.31	247.71	36.58	42.85	42.90			
	33-35	146.99	33.99	39.82	39.79	145.82	34.08	40.89	40.57			
	38-40	76.99	30.94	38.32	37.98	85.78	31.52	38.97	38.58			

um tamanho máximo de dicionário de 5000 elementos por escala, e o método de controle de redundância proposto em [49], que se revelou adequado também para o 3D-MMP. A atualização é feita apenas para escalas cujas dimensões correspondam de metade ao dobro das dimensões da escala original.

Após terminar a codificação de cada grupo de 8 quadros, é aplicada a filtragem de redução de efeito de bloco proposta no Capítulo 6. Os valores dos parâmetros τ e s foram fixados respectivamente em 32 e 100, e o parâmetro α é otimizado exaustivamente de entre um conjunto de oito valores possíveis (0, 0.05, 0.08, 0.10, 0.12, 0.15, 0.17 e 0.20), de modo a maximizar o PSNR da reconstrução, sendo transmitido no fluxo de dados após ser codificado com recurso a um codificador aritmético adaptativo.

Na Tabela 7.1, é apresentado o PSNR médio correspondente às três componentes de cor dos primeiros 64 quadros de 3 sequências de teste. É ainda apresentado o BD-PSNR, que corresponde ao ganho de desempenho do 3D-MMP relativamente ao H.264/AVC.

É possível observar que o desempenho do método proposto ultrapassa o do H.264/AVC para a sequência Container, onde a existência de movimento uniforme permite a estimação simultânea de vários quadros através do modo de predição direcional. Adicionalmente, a homogeneidade do movimento permite em muitos casos estimar eficientemente os vetores, contribuindo para uma baixa entropia associada à sua transmissão.

A sequência Coastguard apresenta um cenário bem diferente, com diversos objetos se movimentando em várias direções, resultando num movimento mais errático e difícil de prever. Desde modo, não só se torna pouco provável a predição simultânea de vários quadros, como também diminui a eficiência da estimação dos vetores, o que tem por consequência o aumento da quantidade de informação respeitante aos vetores, que precisa

ser transmitida, chegando esta a ser responsável por quase metade da taxa. O H.264/AVC consegue ser mais eficiente neste caso, apoiado na ME com precisão fracionária e com uma janela de busca maior.

Para o caso da sequência Akiyo, verifica-se um bom desempenho dos esquemas de predição propostos, mas o fato da sequência possuir um fundo estático, que é codificado muito eficientemente com recurso aos modos *skip* e *copy* do H.264/AVC, faz com que o desempenho deste último supere o do algoritmo proposto em quase 1 dB nesta situação.

É importante salientar que o desempenho do H.264/AVC supera o do 3D-MMP para os casos em que a informação respeitante aos vetores é mais significativa. Assim, o desenvolvimento de técnicas mais eficientes de predição e transmissão destes vetores poderá contribuir para que o desempenho de compressão do algoritmo proposto supere o do H.264/AVC em todas as situações.

7.5 Conclusões

Neste capítulo, foi proposto um esquema de compressão de sinais tridimensionais baseado no MMP. Este esquema contempla a utilização de uma predição hierárquica tridimensional, conjuntamente com uma extensão tridimensional do MMP usada para codificar o resíduo resultante.

Foram propostas várias técnicas que permitiram aumentar o desempenho do MMP, e foram avaliados os diversos parâmetros do algoritmo com impacto no seu desempenho. Adicionalmente, foram propostos alguns novos métodos de predição, entre os quais um modo tridimensional baseado no critério dos mínimos quadrados e um modo direcional.

O desempenho do algoritmo proposto foi avaliado para a compressão de sinais de vídeo, apresentando-se próximo ao do H.264/AVC. No entanto, acreditamos ser ainda possível levar a cabo diversas otimizações do algoritmo proposto, visando um aumento de desempenho da predição, nomeadamente através de uma estimação e codificação mais eficiente dos vetores direcionais, que chegam em alguns casos a contribuir com cerca de metade da taxa usada pelo codificador.

No futuro, planeamos testar o método proposto para outros tipos de sinais de entrada, entre os quais sinais provenientes de radares meterológicos, de ressonância magnética, imagens multiespectrais ou mesmo imagens e vídeos multivistas. Estes vários tipos de sinais caracterizam-se por apresentar uma elevada correlação ao longo da múltiplas dimensões, que poderá vir a ser explorada com sucesso recorrendo a este tipo de técnicas.

Adicionalmente, no futuro também pretendemos levar a cabo a substituição do MMP [3] por outras técnicas de compressão de resíduo tridimensional, tais como transformadas [97–104], visando o desenvolvimento de algoritmos de compressão de elevado desempenho e baixa complexidade computacional, que poderão revelar-se numa alternativa viável aos codificadores híbridos.

Capítulo 8

Conclusões e perspectivas

8.1 Considerações finais

Nos capítulos anteriores, foram descritos os tópicos principais sobre os quais se baseou o trabalho desenvolvido neste tese. Conclusões específicas, visando cada um dos tópicos abordados e os respectivos resultados, são apresentados na última seção do capítulo correspondente.

O paradigma do casamento de padrões recorrentes multiescalas foi estudado em detalhe, e foram propostas diversas otimizações, focando o aumento do desempenho taxa-distorção e da qualidade percetual das imagens reconstruídas, bem como a redução da complexidade computacional dos algoritmos propostos. Como resultado, foram desenvolvidos novos esquemas de codificação para imagens estáticas, documentos compostos digitalizados ou sinais de vídeo. Cada um dos algoritmos propostos atingiu um nível de desempenho competitivo com o dos algoritmos que constituem o estado da arte dessa aplicação específica.

Adicionalmente, foi iniciado outro tópico de pesquisa, resultante da combinação de uma extensão volumétrica do MMP com um esquema preditivo hierárquico tridimensional. A nova arquitetura de codificação foi testada para compressão de sinais de vídeo, apresentando resultados promissores para esta aplicação em particular. Tal demonstrou as potencialidade da nova arquitetura de codificação, justificando pesquisas futuras, nomeadamente respeitantes à sua aplicação para outros tipos de sinais volumétricos.

8.2 Contribuições da tese

Nesta seção, é apresentado um resumo das contribuições mais importantes desta tese. Essas contribuições dizem principalmente respeito ao algoritmo MMP, mas algumas delas são extensíveis a outros algoritmos baseados em casamento de padrões, ou mesmo a qualquer algoritmo de compressão que utilize uma abordagem bloco a bloco.

A validação do trabalho desenvolvido junto à comunidade científica foi considerada fundamental como meio de aferição da sua relevância. Consequentemente, a maioria dos resultados obtidos foi submetido para publicação em revistas e congressos internacionais. A lista completa das publicações resultantes do trabalho desenvolvido no âmbito desta tese pode ser encontrada no Apêndice J.

As contribuições mais importantes desta tese podem ser sumarizadas nos seguintes tópicos:

- **O codificador MMP-*compound*: um codificador de documentos compostos digitalizados baseado no MMP.**

  As investigações relativas à otimização do desempenho do MMP para imagens naturais e imagens de texto, deram origem a dois novos codificadores, respectivamente o MMP-FP e o MMP-*text*. Estes codificadores revelaram-se capazes de superar a eficiência dos algoritmos que constituem o estado da arte nessa área de aplicação. Combinando ambos os algoritmos num novo método que efetua a segmentação dos documentos compostos respectivamente nas suas componentes suaves e de texto, foi criado um novo codificador de documentos compostos, o MMP-*compound*, descrito no Capítulo 3.

  Os resultados experimentais demonstraram que o desempenho do algoritmo desenvolvido superou consideravelmente o dos métodos que constituem o estado da arte na compressão de documentos e o de codificadores de imagens convencionais, tanto do ponto de vista objetivo como perceptual.

  O trabalho desenvolvido para este tópico de pesquisa resultou no artigo: "Scanned Compound Document Encoding Using Multiscale Recurrent Patterns", publicado na revista *IEEE Transactions on Image Processing*.

- **O codificador MMP-*video*: um algoritmo de compressão de vídeo totalmente baseado no paradigma do casamento de padrões.**

  O desenvolvimento de um codificador de vídeo baseado no MMP era igualmente um dos objetivos principais desta tese.

  A investigação conduzida resultou no algoritmo MMP-*video*, um codificador híbrido que utiliza o MMP na compressão dos resíduos resultantes tanto da predição Intra como da estimação de movimento. O uso do casamento de padrões recorrentes multiescalas foi otimizado para a codificação de sinais de vídeo, com base nos conhecimentos obtidos de estudos anteriores, e foram adicionalmente propostas algumas novas técnicas, especificamente orientadas para as características particulares dos sinais de vídeo.

  O codificador de vídeo desenvolvido é assim totalmente suportado pelo paradigma do casamento de padrões, atingindo um desempenho de compressão superior ao do

H.264/AVC, que constitui o estado da arte para esta aplicação. Estes resultados ajudaram a demonstrar que o casamento de padrões poderá constituir uma alternativa viável ao paradigma dominante das transformadas.

Estes resultados validaram a utilização do MMP também para a compressão de sinais de vídeo, e deram origem ao artigo: "Efficient Recurrent Pattern Matching Video Coding", publicado na revista *IEEE Transactions on Circuits and Systems for Video Technology*. Este tópico específico será ainda alvo de investigações futuras, de modo a estender o leque de aplicações do algoritmo desenvolvido a sequências de alta resolução ou mesmo a sinais de vídeo multivistas.

- **Estudo de técnicas de redução da complexidade computacional para os codificadores baseados no MMP.**

  A maior limitação à utilização prática do MMP prende-se com a sua elevada complexidade computacional. Esta limitação surge ainda mais agravada pelo fato de se verificar também uma complexidade considerável do lado do decodificador, tornando pouco viável a utilização prática do MMP até para aplicações nas quais o sinal de entrada apenas precisa de ser codificado uma vez, para ser decodificado em múltiplos recetores.

  Foram desenvolvidas técnicas de redução da complexidade computacional para o MMP que permitiram diminuir o tempo necessário para a codificação e decodificação, respectivamente em 86% e 95%, sem afetar significativamente o desempenho de compressão dos algoritmos. As técnicas desenvolvidas podem ser usadas conjuntamente com outras técnicas propostas anteriormente, permitindo ganhos ainda maiores nos tempos de computação.

  No entanto, este tópico de investigação será alvo de trabalho futuro, dado que a complexidade computacional dos codificadores baseados no MMP ainda é muito elevada quando comparada à de algoritmos baseados em transformadas.

  Os resultados obtidos neste tópico de trabalho foram descritos no artigo: "Computational Complexity Reduction Methods for Multiscale Recurrent Pattern Algorithms", apresentado no congresso *Eurocon2011 - IEEE International Conference on Computer as a Tool*, que decorreu em Lisboa, e publicado nos anais do congresso.

- **Melhorar a qualidade perceptual das imagens codificadas com o MMP, com recurso a técnica de pós-processamento.**

  Tendo em conta que o MMP é um algoritmo que processa as imagens de entrada bloco a bloco, é comum serem introduzidos alguns artefatos nas imagens reconstruídas, especialmente a taxas de compressão elevadas. Tal motivou o estudo prévio

de técnicas de filtragem para redução do efeito de bloco, mas essas técnicas revelaram no entanto algumas ineficiências.

Nesta tese, foi proposto um novo método de filtragem para redução do efeito de bloco, de modo a ultrapassar as limitações dos métodos anteriores e melhorar a qualidade percetual das imagens reconstruídas.

O método proposto utiliza um filtro FIR adaptativo para processar cada bloco da imagem de entrada. A resposta do filtro é adaptada a cada região da imagem, com base nas suas características locais. Para tal, é efetuada uma análise da variação total de cada bloco, de modo a ajustar iterativamente o comprimento do suporte a utilizar no filtro. O método proposto é assim uma técnica de pós-processamento, que pode ser aplicado em qualquer imagem reconstruída, independentemente do algoritmo usado na sua codificação.

O filtro proposto pode assim ser usado tanto como um método iterativo, otimizado para cada tipo específico de imagens, ou pode operar com um conjunto pré-estabelecido de parâmetros, de modo a contornar a necessidade de ter que enviar esses parâmetros para o descodificador. Tal permite a utilização do filtro como uma técnica de pós processamento, tornando viável a sua aplicação conjunta com normas de codificação que possuem um fluxo de dados normalizado. O método proposto demonstrou bons resultados quando usado em imagens codificadas com vários algoritmos diferentes, incluíndo a norma H.264/AVC, a proposta de norma HEVC e o JPEG.

Os resultados obtidos foram apresentados no artigo: "A Generic Post Deblocking Filter for Block Based Image Compression Algorithms", publicado na revista *Elsevier Signal Processing : Image Communications*.

- **Desenvolver um codificador de sinais volumétricos baseado em casamento de padrões recorrentes multiescalas.**

  Com o intuito de investigar a aplicabilidade do casamento de padrões recorrentes multiescalas para diversos tipos de sinais volumétricos, tais como sequências de vídeo, vídeos tridimensionais, imagens multiespectrais e sinais provenientes de ressonâncias magnéticas ou radares meteorológicos, foi desenvolvido um novo algoritmo de compressão de sinais volumétricos tridimensionais, baseado em predição hierárquica e numa extensão tridimensional do MMP, usada para comprimir o resíduo resultante.

  Foram propostos vários modos de predição tridimensionais, incluindo extensões das técnicas utilizadas pelo H.264/AVC, um modo baseado no critério dos mínimos quadrados e um modo direcional tridimensional. Adicionalmente, foi levada a cabo

uma avaliação extensiva de cada um dos parâmetros com impacto no desempenho do MMP, de modo a verificar a sua influência na nova arquitetura volumétrica.

O algoritmo desenvolvido foi testado para a compressão de sequências de vídeo monoscópicas. No entanto, serão futuramente investigadas outras modificações que visam o aumento do desempenho do algoritmo, e este será avaliado para outros tipos de sinais de entrada.

8.3 Perspectivas futuras

O trabalho apresentado nesta tese, assim como trabalhos anteriores com ele relacionados, demonstraram as potencialidade do algoritmo MMP para codificação de imagens, ao atingir resultados que competem com os do estado da arte para várias aplicações. No entanto, no estado atual de desenvolvimento, a elevada complexidade computacional do MMP torna-o ainda num algoritmo proibitivo para a maioria das aplicações práticas. Consequentemente, inúmeras questões permanecem abertas. Porquê investir tempo no desenvolvimento de um algoritmo de compressão tão complexo? Precisamos realmente de esquemas de compressão de imagens e vídeos alternativos, ou os algoritmos existentes são suficientes para suprir a demanda?

A busca por soluções alternativas para problemas existentes é no entanto a melhor forma de chegar a soluções que rompem com as abordagens e conceitos pré-estabelecidos relativos a esses problemas, resultando numa capacidade acrescida de olhar o problema de pontos de vista distintos. Deste modo, o conhecimento adquirido com a investigação relativa ao MMP poderá vir a tornar-se útil também para outros paradigmas de compressão, nomeadamente para os algoritmos baseados em transformadas. Entender o MMP pode ajudar a entender também a natureza das imagens, permitindo desenvolver novas formas de as representar. Consequentemente, a investigação de algoritmos fora das tendências gerais, tais como os abordados nesta tese, têm o potêncial para alargar as fronteiras do conhecimento da compressão de imagens, e devem por isso continuar.

Para além disso, a complexidade computacional tem vindo a se tornar um problema cada vez menos relevante com o passar do tempo, dado o aparecimento de máquinas cada vez mais poderosas e com maior capacidade de computação. Soma-se a isto também o desenvolvimento de *hardware* específico para manipulação de imagens, tais como as GPUs, que poderão contribuir para o uso generalizado de algoritmos como o MMP. Tal leva-nos ainda a outra questão em aberto, que diz respeito ao impacto das crescentes capacidades do *hardware* no desempenho dos algoritmos propostos. Como poderá o MMP ser melhorado de modo a melhorar o seu desempenho de compressão é outra interessante pergunta cuja resposta continua em aberto.

De entre as propostas apresentadas nesta tese, vários tópicos poderão ainda originar

outras linhas de pesquisa. Os novos desafios do ponto de vista da compressão de imagens e vídeo, dos recursos de *hardware* e do aparecimento de novos tipos de conteúdos tornam a compressão de dados multimídia um tópico de investigação permanentemente aberto.

No futuro, esperamos estender a aplicação do filtro de redução do efeito de bloco descrito no Capítulo 6, a uma arquitetura volumétrica, no desenvolvimento de uma técnica de filtragem espaciotemporal. Esta abordagem poderá permitir a atenuação simultânea de dois dos artefatos mais incomodativos em sequências de vídeo codificadas a elevadas taxas de compressão: o efeito de bloco e o *flickering* visível em zonas uniformes muito quantizadas. A informação conjunta de tempo e espaço poderá ser útil na identificação das bordas dos blocos introduzidas na codificação, relativamente aos bordos reais dos objetos e às mudanças de cena.

O tópico de trabalho abordado no Capítulo 7 é no entanto aquele que apresenta mais linhas com potencial para investigações futuras. Várias melhorias poderão ainda ser levadas a cabo para aumentar o desempenho da predição espaciotemporal, e a estimação e codificação dos vetores direcionais ainda apresenta algumas margens para melhoramentos. Adicionalmente, esperamos vir a desenvolver um esquema de compressão alternativo baseado nesta arquitetura, onde o MMP dará lugar a uma transformada tridimensional para codificação do resíduo espaciotemporal, permitindo assim desenvolver codificadores de vídeo de baixa complexidade.

Appendix A

Introduction

A.1 Motivation

Digital multimedia contents have experienced an accelerated dissemination over the past years. Several advances in consumer electronics resulted in a rapid proliferation of digital cameras and scanning devices, with increasing resolutions and capabilities. As a consequence, the amount of information that needs to be handled and stored as video, images and digital media libraries is increasing everyday.

Digital video is now ubiquitous: the traditional analog television broadcasting is being replaced by a new digital video service, and we are facing an explosion of digital video applications and providers, such as *Youtube*, where the users can share video contents with other users from all around the world. Video and image became usual in web pages, and many of us are just as likely to catch the latest news on the web as on TV, either in our computers or mobile handsets.

At the same time, digital media libraries also experimented an increasing popularity. Many international newspapers made their editions available in digital format, and an increasing number of libraries are creating digital copies of their collections, making sensitive and historic contents available for a larger number of users, without concerns on preservation issues.

Furthermore, some emerging multi-client applications, such as cloudset-screen computing [105], virtualized screen systems [106] or deep-shot systems [107], also rely on the transmission of visual information across networks.

This massive amount of information that needs to be stored and transmitted imposes the need for efficient image and video compression algorithms, as the increase in storage capacities and the ever growing available bandwidth and network speeds are not enough to satisfy this demand.

Over the last decades, the transform-quantisation-based encoding methods have been dominant in this area, either using the traditional discrete cosine transform (DCT) and

discrete wavelet transform (DWT), or the integer transforms proposed on recent encoding standards. However, despite being particularly efficient for smooth, low-pass images, poor rate-distortion (RD) performance and highly disturbing visual artifacts frequently appear in other image types, such as text images, computer generated images, compound documents (text and graphics) or textures, among others.

The efficiency of these methods rely on the energy compaction achieved by the transform, when a high spatial correlation actually exists on the image. In this case, the transform coefficients representing the highest frequencies tend to be of little importance, or even negligible, and can be subjected to a coarse quantization or simply discarded. This allows to achieve a high compression of the input signal without compromising the visual quality of the reconstruction. In some cases, the coding efficiency can be further improved with predictive schemes that efficiently reduce the spatial and temporal correlation of the input signal. At a final stage, entropy coding is commonly used to reduce the statistical correlation still present on the generated information [1], improving the overall compression efficiency.

However, when the input signal does not present a low-pass nature, as for the case of text and graphics images, or synthetic and computer generated images, transform-based algorithms present a poor compression efficiency. If a coarse quantization is applied to the high-frequency coefficients, highly disturbing visual artifacts may appear. On the other hand, if these coefficients are not coarsely quantized, in order to maintain a suitable visual quality, high compression ratios may not be achieved.

In this sense, several hybrid algorithms have been proposed to address this issue. Their approach involves the segmentation of the input signal into high-pass (text) and low-pass (smooth) regions, in order to process each one with an optimized algorithm. However, the success of such methods has a strong dependence on the performance of the segmentation step, that is not able to provide satisfactory results in all situations.

All these limitations motivated the on-going research on alternative compression paradigms for image and video signals, but the quest for a universal method, that works for all input sources, has proved to be a challenging task.

The investigation described in this thesis relies in a promising algorithm, that already proved its versatility for a wide range of input signal types. The multidimensional multiscale parser (MMP) [2, 3] was originally proposed as a lossy data compression algorithm. It has been successfully used either for lossy and lossless compression of several data types, with state-of-the-art results on many applications. The compression of lossless [6] and lossy still images [4, 5], video sequences [7], stereoscopic images [8], touchless multiview fingerprints [9] or even ECG's [10–12] are examples from such applications.

A.2 Main objectives

The previously identified research issues provide an opportunity for exploiting alternative algorithms, in order to fulfill the need for efficient and versatile data compression methods. Among the wide variety of proposals, MMP assumes a privileged position, due to its already proven versatility and excellent rate-distortion performance in many coding applications.

The work described in this thesis investigates efficient MMP-based compression frameworks, in order to exploit the potential of such paradigm for digital visual data compression. The research goals are the optimization of the algorithm for still images and video signals, as well as the development of specifically orientated architectures for compound documents and video compression. The development of a three-dimensional framework is also a goal to achieve, in order to exploit a joint spatiotemporal decorrelation using MMP with a volumetric hierarchical prediction scheme.

This way, the main research topics of this thesis can be summarized as follows:

- **Improve the efficiency of MMP for image coding.**

  The focus will be on optimizing the algorithm to improve its efficiency either for smooth, as well as text and graphics image compression, in order to develop a competitive compound scanned document encoder. The high heterogeneity verified on this type of input sources is an important obstacle when designing efficient encoding algorithms. Thus, sufficiently robust and reliable compression methods to respond to this increasingly relevant application, have not been presented yet.

  The improvements target both the objective and visual quality of the reconstructed images, in order to affirm MMP as a viable alternative to other state-of-the-art encoders. The results of the proposed schemes will be compared with the compression performance of state-of-the-art encoders, as well as those from the previous versions of the MMP algorithm.

- **Investigating the efficiency of the MMP paradigm for video coding applications.**

  Preliminary tests on using MMP to compress time estimated residues on a hybrid coding framework showed promising results [7, 13, 14]. However, this previous work was supported by an obsolete version of MMP [15], that still used transforms to encode the reference frames.

  In order to allow a complete substitution of transforms on the proposed encoder, a fully pattern matching based algorithm was developed.

  The experimental results will be assessed by comparison with the current state-of-the-art video compression standard: the H.264/AVC high profile video encoder.

The most recent video standard proposal, the HEVC [16], was not adopted for comparison purposes, because it was not still implemented at the time the work presented in this thesis was accomplished.

- **Address the computational complexity issue**
  MMP already proved its high coding efficiency and versatility, but still presents an important drawback which limits its practical use on most applications: a high computational complexity.

  The reduction of MMP's computational complexity can be a decisive step while affirming MMP as a viable alternative to the transform-quantization paradigm.

  The experimental results will be assessed by comparison with other benchmark versions of the MMP algorithm and previous works on this area.

- **Develop a volumetric multiscale recurrent pattern based compression framework.**

  The investigation of a three-dimensional prediction based compression scheme is worth of investigating. By combining a volumetric extension of the MMP algorithm with a 3D hierarchical prediction scheme it is possible to develop a volumetric data compression method applicable to a multitude of input data sources, such as weather radar data and video sequences.

  This research topic includes the development of three-dimensional prediction modes and optimized architectures to take advantage of the spatiotemporal redundancy.

  The experimental results will be evaluated against those from previous versions of MMP-based encoders and those from other state-of-the-art compression algorithms, applicable for such types of input data.

A.3 Outline of the thesis

This thesis is organized as follows. Chapters 1 through 8 provide an overview of the work developed in the scope of this PhD thesis, and are written in Portuguese. These chapters are complemented by a more extensive and detailed description, presented on appendices A through H, which are written in English.

The current appendix presents an introduction related to the research area topics of this PhD thesis. The motivation for the developed work is presented, as well as a list of the main objectives and goals to achieve with this work.

Appendix B presents a detailed description of the most important aspects and features of the multidimensional multiscale parser algorithm. Experimental results are presented and evaluated against that of state-of-the-art transform-based image encoders.

Appendix C presents the optimizations performed in the MMP algorithm in order to tune its performance specifically for smooth and text and graphics images. The description of a scanned compound document encoder framework, based on the two previously described algorithms is presented. The experimental results are evaluated against that of state-of-the-art compound document encoders.

Appendix D investigates the use of MMP in a fully pattern matching video compression algorithm. Following the good performance previously achieved by MMP when encoding still images and motion-compensated residues, a video compression framework which totally substitutes the transforms used on H.264/AVC by MMP was developed. An additional prediction mode specifically oriented to the chrominance components was also included on the proposed codec, and is described on this chapter. The results of the proposed method are evaluated against that of JM17.1 H.264/AVC reference software.

In Appendix E, two computational complexity reduction methods are presented. One of these techniques is specially oriented towards the MMP algorithm, while the other can be easily adapted to other pattern matching algorithms. Both present considerable gains on the computation time both on the encoder and the decoder side. The computational complexity reduction is evaluated comparing the computation time with that from a benchmark version of the MMP algorithm.

Appendix F presents an improved post-processing deblocking method. This method was originally developed targeting the increase on the subjective quality of reconstructed images encoded using MMP. The proposed method overpassed some implementation issues from the existing deblocking filter. The nature of the proposed method allied to an extensive optimization, allowed its successful application on images and videos encoded using other block based compression algorithms. Thus, the results from the proposed post-processing deblocking algorithm are evaluated not only on MMP coded images, but also on images compressed using transform-based codecs such as JPEG, H.264/AVC, or even the upcoming standard HEVC.

In Appendix G, a joint spatiotemporal volumetric framework is proposed. This framework adopted a 3D hierarchical prediction scheme, with a 3D extension of the MMP algorithm being used to compress the generated residue. Several volumetric prediction modes are investigated and optimized for the proposed framework. The proposed algorithm is evaluated for video compression applications, but can also be applied to other volumetric-like signals, such as meteorological radar signals, tomographic scans or multispectral/multiview images.

Conclusions regarding the work developed and the achieved results are summarized on Appendix H, as well as the list of the contributions from this thesis. This appendix also presents the main topics considered for future work.

Appendix I presents an overview of the test images and video sequences used throughout this work. Still images and video sequences with different characteristics were con-

sidered, in order to evaluate the versatility of the proposed algorithms. Still images vary from smooth, natural images to text and compound images, and video sequences vary from slow motion to video signals with high complex motion.

Appendix J presents a summary of the submitted and published papers, which were used to propagate the contributions of the research work presented on this thesis among the research community.

Appendix B

Multiscale recurrent patterns: The MMP algorithm

B.1 Introduction

Pattern matching algorithms have been well investigated over the past decades, to compress several types of data sources. Their approach comprises the division of the input data into segments, in order to approximate each one by blocks (code-vectors) chosen from a dictionary, also referred to as a codebook. This approximation can be performed following different criteria, either in a lossy or lossless approach. In a lossless compression, an exact match is required between the input data segment and the code-vector chosen for its representation, while in lossy compression schemes, a certain degree of distortion is allowed for the match, in order to increase the compression ratio.

Among pattern matching algorithms, two classes of methods are arguably the most popular and well known: the Lempel-Ziv (LZ) [17–27] algorithms and vector quantization (VQ) [28] based encoders.

The LZ class of compression algorithms emerged from the work of Abraham Lempel and Jacob Ziv, and rely on two different paradigms. The LZ77 [17] algorithm is commonly designed as a *sliding window* compression method, as it tries to perform the match between the data segment being encoded and the data previously processed, which is contained inside a search buffer. For that purpose, a pair of numbers called a *length-distance pair* is used, indicating the offset to the previously decoded match and the length of that match, respectively. Several variations of these method were proposed [19–22], improving the encoding of *length-distance pair* with variable length codes. Several lossless compression methods, such as Zip, Gzip and Arj rely on these improved versions of the LZ77 algorithm.

The LZ78 algorithm [18] uses an adaptive dictionary to perform the matches, in order to overcome the intrinsic locality of the *sliding window* approach. The input data is

forward scanned, and the algorithm tries to perform a match with the codewords stored in the dictionary. At this point, the algorithm outputs the index that identifies the longest match in the dictionary, if one is available, as well as the match length and the first character that caused a match failure. The resulting pattern is then added to the dictionary, and will be available to compress subsequent data. The popular Unix Compress and GIF use an improvement of LZ78: the LZW algorithm [23]. Other variations from the LZ78 algorithm have been proposed in the literature [23–27]. Several lossy image compression algorithms are also known as lossy Lempel-Ziv algorithms [34, 108–110].

Vector quantization is another popular pattern matching algorithm. In the traditional VQ coders, the input signal is segmented into blocks or vectors, and each of these blocks are approximated by a codevector of the dictionary, that contains a representative set of the input data source patterns. A certain distortion is generally allowed for the match, and the index of the chosen codevector is transmitted to the decoder, that is able to fetch the same pattern from a local, synchronized, copy of the dictionary.

Despite the success achieved by pattern matching methods for applications like lossless image compression [29] and binary image coding [30, 31], this paradigm, in general, has not yet produced efficient alternatives to transform-based encoders for lossy images [32–35] or video coding [36–39].

An exception can be found on the Multidimensional Multiscale Parser [2, 3]. This pattern matching algorithm can be seen as a combination of the LZ methods and vector quantization. The input data is partitioned into blocks that are approximated using codevectors from a codebook, such as in VQ coders and an adaptive codebook updated with previously processed patterns is used to perform those matches, which can occur for variable dimensions, such as in the LZ methods.

Furthermore, MMP has another feature that distinguishes it from previous algorithms, and that is the key feature for its high degree of adaptiveness: it allows scale adaptive pattern-matching [3]. Instead of restricting the match to blocks of a constant size, MMP waives this restriction by performing contractions and expansions of the codewords, in order to allow to match blocks with different dimensions. This concept exploits the self-similarity present on natural images, a property exploited for example by fractal encoders [40].

The high degree of adaptiveness allowed MMP to outperform state-of-the-art compression methods for a wide range of applications, from lossy [5] and lossless [6] still images, to video sequences [7], compound documents or stereoscopic images [8, 41]. Good results were also achieved while encoding audio signals [42, 43], touchless multiview fingerprints [9] or even ECG's [10, 11].

In this appendix, we describe the most relevant aspects of the MMP algorithm, focused on image compression.

B.2 The MMP algorithm

As a pattern matching block based compression algorithm, MMP first divides the input data source into fixed dimensions, non-overlapping adjacent blocks, to be sequentially processed on a raster scan order. Each of these blocks is individually processed, resulting in an optimized segmentation tree, which is represented in the bitstream send to the decoder. The patterns learned while coding a given block become available to approximate the subsequent ones, conferring MMP the ability to adapt to the input signal characteristics.

B.2.1 Optimizing the segmentation tree

For each of the fixed dimensions input blocks with $M \times N$ pixels, belonging to scale l, X^l, MMP starts by finding the best code-vector S_i^l of the dictionary $\mathcal{D}^l$ to represent X^l, based on an R-D optimization function J, given by:

$$J = D(X^l, S_i^l) + \lambda R(S_i^l), \tag{B.1}$$

where λ is a *lagrangian multiplier* [44], that weights the relative importance of the rate R required for the representation over its resultant distortion D. The distortion D is computed as:

$$D(X^l, S_i^l) = \sum_{x=1}^{M} \sum_{y=1}^{N} (X^l(x,y) - S_i^l(x,y))^2, \tag{B.2}$$

and the rate R is estimated through the probability of occurrence [1] of the code-vector indices. As separate probability models are used for each dictionary scale (l), the probability of each index is conditioned according to its scale, so:

$$R(S_i^l) = -\log_2(Pr(i|l)). \tag{B.3}$$

In other words, $Pr(i|l)$ depends on the quotient between the number of times the index i from scale l was previously used, and the total utilization of indices from scale l.

After selecting the code-vector that minimizes the lagrangian cost function, the algorithm segments the original block, X^l, into two new blocks at a lower scale, X_1^{l-1} and X_2^{l-1}, each with half the pixels of the original block. The same matching procedure described is then recursively applied to each sub-block, down to elementary 1×1 sub-blocks (scale 0).

The sum of the representation cost of the two halves is then compared with the cost of representing X^l with a single code-vector, in order to decide whether or not to segment the original block. As the decision of segmenting or not the original block needs to be signalled to the decoder, a segmentation flag is transmitted for that purpose. Thus, the

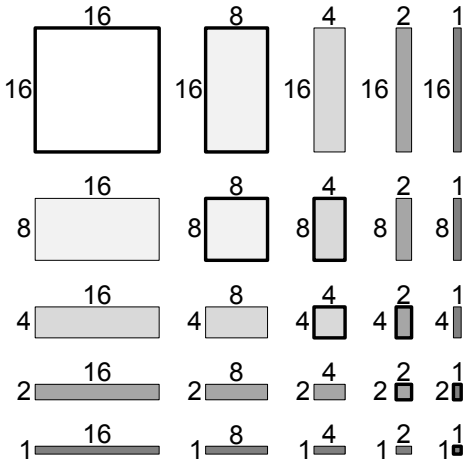


Figure B.1: Possible block dimensions using the flexible and the dyadic partition schemes, for initial block size of 16×16 pixels.

original block will be segmented if:

$$J(X^l) + \lambda R_{\text{nseg}} > J(X_1^{l-1}) + J(X_2^{l-1}) + \lambda R_{\text{seg}}, \tag{B.4}$$

that is, its associated cost plus the rate required to transmit the non-segmentation flag, multiplied by λ, is greater than the sum of the cost of representing the two halves, plus the rate required for the segmentation flag, also multiplied by λ.

Originally [3], the MMP algorithm segmented each block in a pre-established direction for each scale, alternating horizontal and vertical directions. In [5], a new segmentation scheme was proposed, where segmentations both in the horizontal and vertical directions are tested at each scale, with the one with the lowest lagrangian cost being selected for each case.

This flexible partition scheme increases considerably the number of different block dimensions, or scales, used by the MMP algorithm. Generically, for initial $M \times N$ pixels blocks, the diadic segmentation schemes resulted in a total number of dictionary scales $\mathcal{N}_{scales}$ translated by the following equation:

$$\mathcal{N}_{scales} = 1 + \log_2 (M \times N), \tag{B.5}$$

Where M and N are powers of two. When the flexible partition mode is adopted, $\mathcal{N}_{scales}$ becomes:

$$\mathcal{N}_{scales} = (1 + \log_2 M) \times (1 + \log_2 N). \tag{B.6}$$

Figure B.1 presents the possible block dimensions when the flexible partition is used when compared with the block dimensions of the original dyadic segmentation method (at bold), for initial blocks with 16×16 pixels.

As referred, each different block dimension has an associated dictionary scale. Fig-

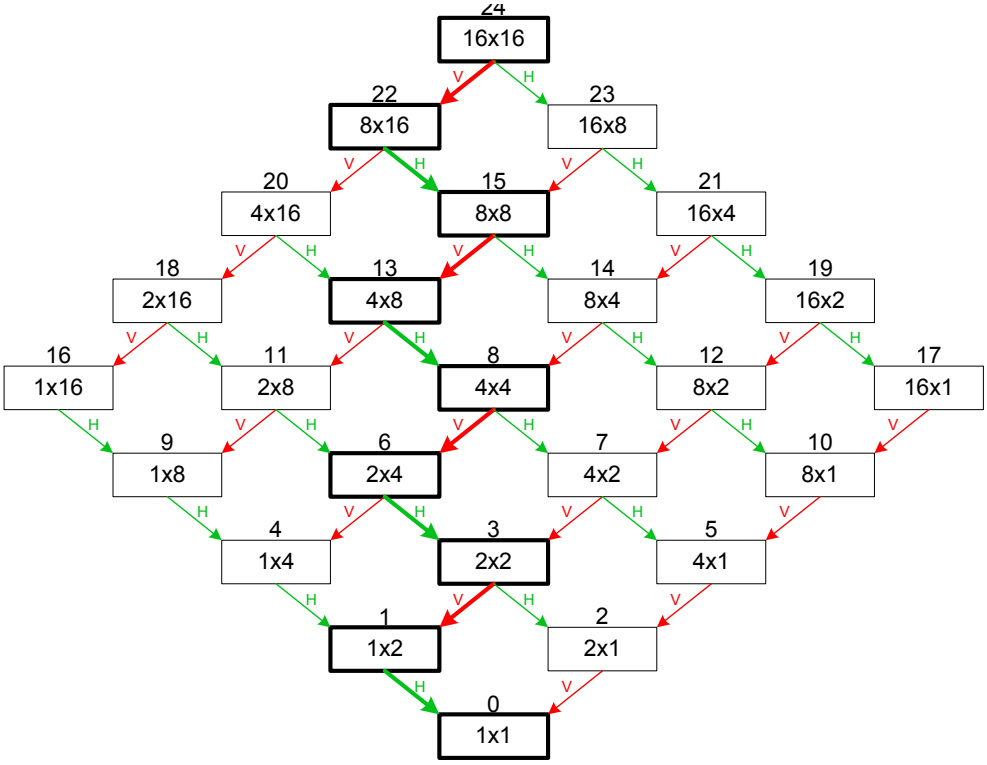


Figure B.2: Level diagram for flexible segmentation *vs.* the original segmentation (at bold).

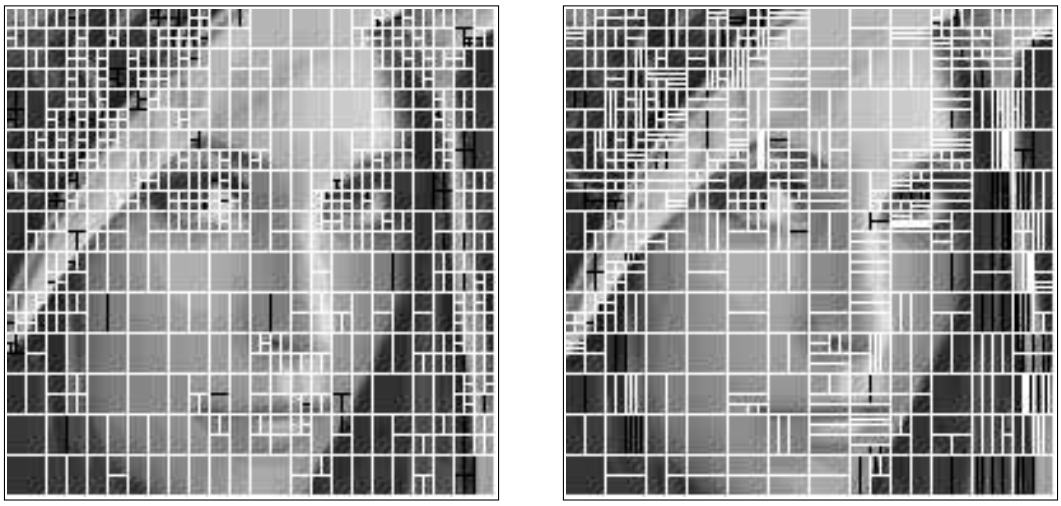
ure B.2 represents the different dictionary scales and their corresponding block dimensions, for the case of flexible partition scheme *vs.* the original dyadic segmentation method (in bold).

The increase in the number of possible block dimensions resulted in a more adaptive algorithm, as it allows MMP to exploit more efficiently the image's structure, with considerable gains for all tested image types.

Figure B.3 shows the image Lena compressed with the original dyadic segmentation scheme and with the flexible partition scheme, for the same target bitrate. It becomes clear from the image that blocks from the new scales are frequently used, specially on more detailed regions. As a result, the algorithm is able to represent those regions with lower distortions.

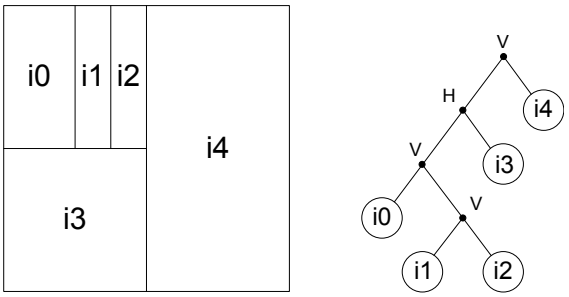
The segmentation pattern used for each block can be represented by a binary tree, $\mathcal{T}$, as shown on Figure B.4 for a case where the original block size is 4×4 pixels. Each leaf of $\mathcal{T}$ corresponds to a non-segmented block, X^l, which is approximated by a single code-vector, S_i^l, identified by its index, i. Each node n_i^l corresponds to a segmented block, which is approximated by the concatenation of two codewords, represented by the child nodes of n_i^l. Each level of $\mathcal{T}$ has a direct correspondence with the scale of the block that it corresponds to. While using the flexible segmentation scheme, the nodes can correspond either to vertical or horizontal segmentations, if both are defined at scale l.

A very important feature of MMP is the ability of using scale adaptive pattern match-



(a) (b)

Figure B.3: Comparison between the resulting segmentation, obtained using a) dyadic scheme and b) flexible scheme, for image LENA.



(a) (b)

Figure B.4: Segmentation of an image block (a) and the corresponding segmentation tree (b).

ing. An original vector X^l, from dictionary scale l can be approximated using one vector S_i^k of scale k with different dimensions, through the use of a 2D separable scale transformation, T_k^l. The scale transformation converts S^k into a scaled version S^l to allow for the match to be performed, so that the codewords from every scale of the dictionary can be used to approximate blocks of any dimensions.

B.2.2 Combining MMP with predictive coding

In [15], a combination of the original MMP algorithm and intra-frame prediction techniques was proposed. This new feature allowed MMP-based encoders to outperform state-of-the-art transform-based encoders for natural image coding.

Predictive coding techniques have the well known property of generating residue samples with highly peaked probability distributions, centered around zero, which favors the adaptation of the arithmetic coder's statistics to the source [4]. The concept is to use the

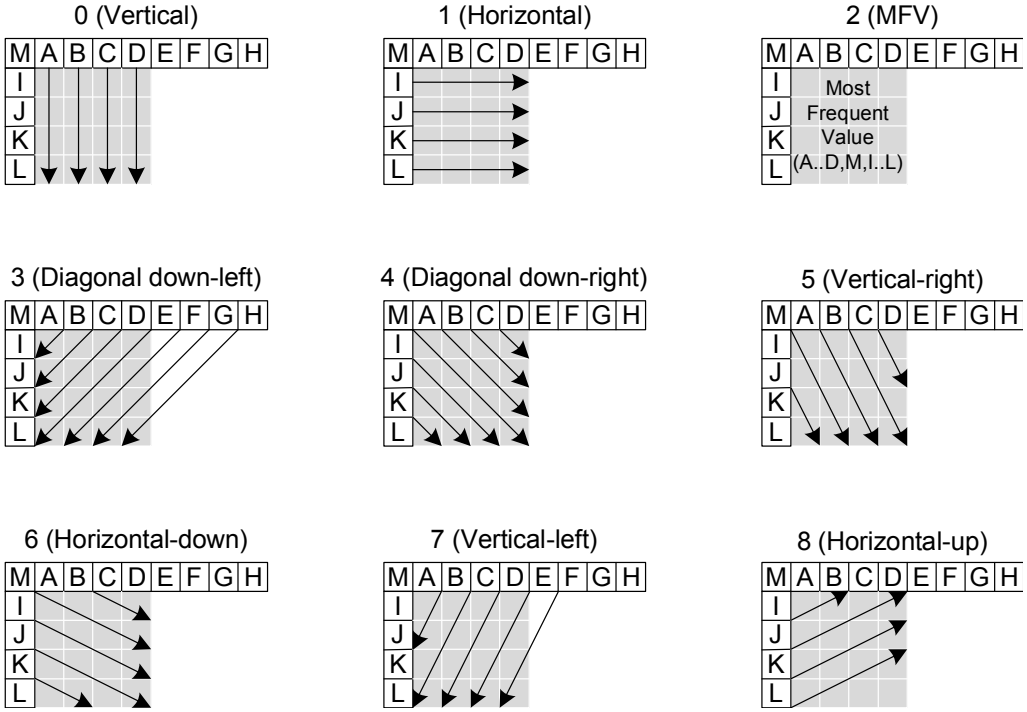


Figure B.5: MMP prediction modes.

previously encoded neighboring samples of the block to generate a prediction block, P_M^l, to be subtracted from the original input block:

$$X^l - P_M^l = R_{P_M}^l. \tag{B.7}$$

This originates a residue block, $R_{P_M}^l$, which is encoded instead of the original block. The residue blocks, $R_{P_M}^l$, tend to have a much lower energy than the original block [4], due to the high degree of spatial correlation that usually exists in natural images, and for this reason, the residue signal is, generally, more efficiently encoded.

The original prediction modes adopted are inspired by those of H.264/AVC standard [45, 111], with only one exception: the DC mode was substituted by the most frequent value (MFV) [15]. In both cases, the prediction mode returns an homogenous block, but in the DC mode, the used value corresponds to the average of the neighboring samples, and in the MFV value, the intensity is equal to the most frequent value among the pixels on the causal neighborhood. This mode revealed to be advantageous over the DC mode when used with MMP, specially for text images. Figure B.5 graphically represents the prediction modes proposed in [15].

In [46], an additional prediction mode based on Least Square Prediction (LSP) was proposed, to complement the existing ones. This extra mode uses the blocks' causal neighborhood (Figure B.6-a) to compute linear prediction coefficients for a given pixel $X(\vec{n}_0)$, located at position $\vec{n}_0$=(x,y), according to an Nth order Markovian model:

$$\hat{X}(\vec{n}_o) = \sum_{i=1}^{N} a_i X(\vec{n}_i), \tag{B.8}$$

where $\vec{n}_i$, with $i = 1, 2, ..., N$, are the spatial causal neighbors presented on Figure B.6.

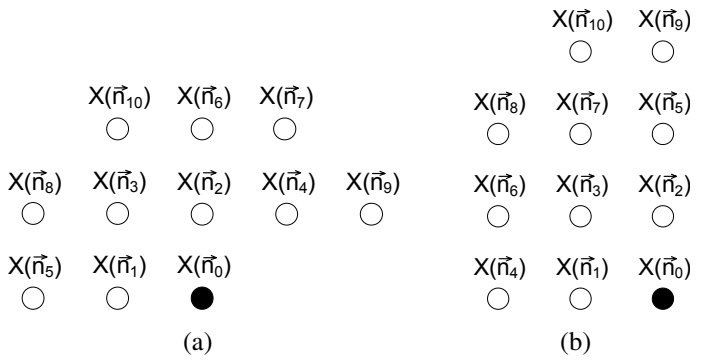


Figure B.6: Original (a) and modified (b) causal pixel neighborhoods.

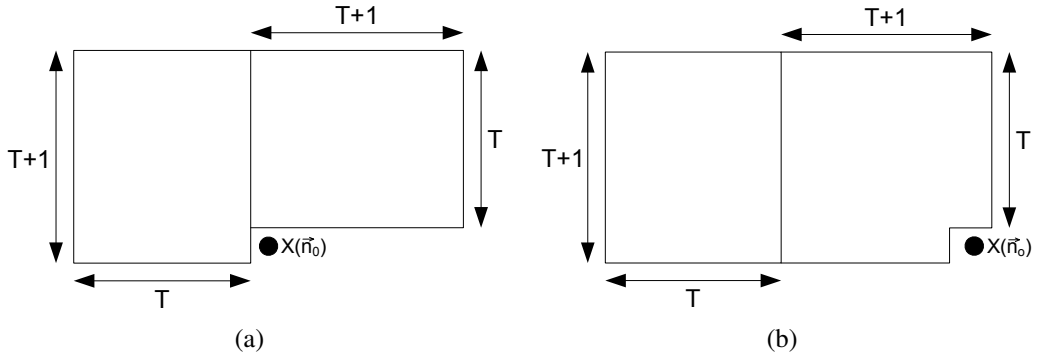


Figure B.7: Original (a) and modified (b) causal training windows.

Thus, the pixel prediction is calculated as a weighted average of the neighbor pixels represented on Figure B.6-a. However, since the encoding is block-based, only pixels from the previous blocks are available to be used by the predictor. When reconstructed pixel values inside the block being predicted are not yet available, their predicted values are used instead, in order to maintain the prediction on a pixel-by-pixel basis.

Under the assumption of the Markov property, the weighted average coefficients a_i, can be trained on a local causal neighborhood. A convenient choice of the training window is the double-rectangular window that contains $M = 2T(T + 1)$ elements, as shown on Figure B.7-a. However, in a block-based prediction approach, pixels in the right of the predicted position may not be available for training, since some of them may belong to a block that still needs to be encoded. For these cases, both the pixel neighborhood and the training window are modified, in order to include only causal elements (see Figures B.6-b and B.7-b, respectively).

In order to express the training procedure using matrix notation, let us define two indicator functions, $g(k)$ and $f(j)$. The function $g(k)$ provides the delta displacement

between the position of a pixel inside the training window of size M and the position of the pixel being filtered, indexed by k. The function $f(j)$ provides the delta displacement between the adjacent neighboring pixels and the pixel to be predicted, in the Nth order markovian model, indexed by j.

The training sequence can then be arranged in an $M \times 1$ column vector $\vec{y} = [X(n - g(1)) \dots X(n - g(M))]^T$. As the prediction window slides through the M positions, the arrangement of the N adjacent neighbors of the local prediction support region in vector forms an $M \times N$ matrix, **C**:

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

This way, the prediction process can be expressed using matrix notation as:

$$\mathbf{C}\vec{a} = \vec{y}. \tag{B.9}$$

Since **C** is an $M \times N$ matrix, with the size of the training window M being defined to be larger than N, the least squares solution of the problem $\min(||\vec{y} - \mathbf{C}\vec{a}||^2)$, can be obtained through the left pseudo-inverse [112], given by:

$$\vec{a} = (\mathbf{C}^T\mathbf{C})^{-1}(\mathbf{C}^T\vec{y}). \tag{B.10}$$

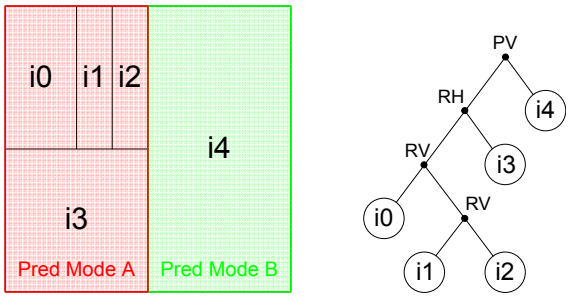
Finally, the obtained prediction coefficient are used in Equation B.8. In [46], it was suggested to use a filter support $N = 10$ and a training window with $M = 112$ pixels.

In the optimization process, all prediction modes are exhaustively tested, and the generated residues are coded using MMP, in order to determine which one achieves the best RD trade-off. Note that unlike other algorithms, the mode that generates the residue with lower energy is not necessarily the best one for encoding with MMP. Depending on the available code-vectors and on their statistical distribution, it can be more efficient to encode a high energy residue block than another of lower energy, for which a proper match cannot be found. The prediction mode is chosen based on the lagrangian cost of the reconstruction, weighting the distortion of the reconstructed block over the rate required to transmit the prediction mode, plus all the information regarding the MMP encoded residue. Once the most efficient prediction mode is determined, it is transmitted to the decoder by using a prediction mode flag.

In other words, each prediction mode P_M will have a lagrangian cost associated:

$$J_{P_M}(X^l) = J(R^l_{P_M}) + \lambda Rate(P_M), \tag{B.11}$$

which depends on the lagrangian cost of its associated residue $J(R^l_{P_M})$ encoding, and



(a) (b)

Figure B.8: Segmentation of an image block with predictive scheme(a) and the corresponding binary segmentation tree (b).

on the rate required to signalize this prediction mode. The mode with the lowest costs $J_{P_M}(X^l)$ associated will be chosen by the optimization process.

The prediction scheme is hierarchically applied across the segmentation tree, down to a scale where blocks still have more than 16 pixels. Furthermore, when a given block is encoded using a segmented prediction, it is considered that the residue is also segmented, and the two halves are individually optimized. Thus, the prediction segmentation always implies its residue segmentation, and the residue can be further segmented to achieve its optimal representation [49].

This results in two different classes of tree nodes, that either correspond only to a residue block's segmentation, or to both the prediction and the residue blocks' segmentation. Each of these two classes further comprehend two different segmentation directions, resulting in a total of four different types of nodes.

Figure B.8 represents an example of a segmentation for an image block and its corresponding segmentation tree, $\mathcal{T}$. The block prediction is segmented into two halves, each one using a different prediction mode. This way, the root node corresponds to a segmentation of both the prediction and the residue on the vertical direction. The residue of the sub-block on the left is further segmented, to obtain an optimal representation, so the remaining tree nodes correspond only to the residue blocks' segmentation.

The use of a hierarchical prediction scheme, allied to RD optimization techniques, allows MMP to determine a good trade-off between the prediction accuracy and the allocated rate. The use of the flexible partition scheme also favors the prediction process. For example, the use of very thin blocks (*e.g.* 16×1) in regions with thin vertical detail, may generate a more accurate prediction than, for example, 4×4 prediction blocks (which have the same number of pixels).

Experimental results have shown that the use of a predictive scheme significantly improves the performance of MMP for smooth images, while maintaining the performance advantage for text and compound images over state-of-the-art transform-based encoders [5, 49]. These good all-round results demonstrate the versatility of the MMP paradigm.

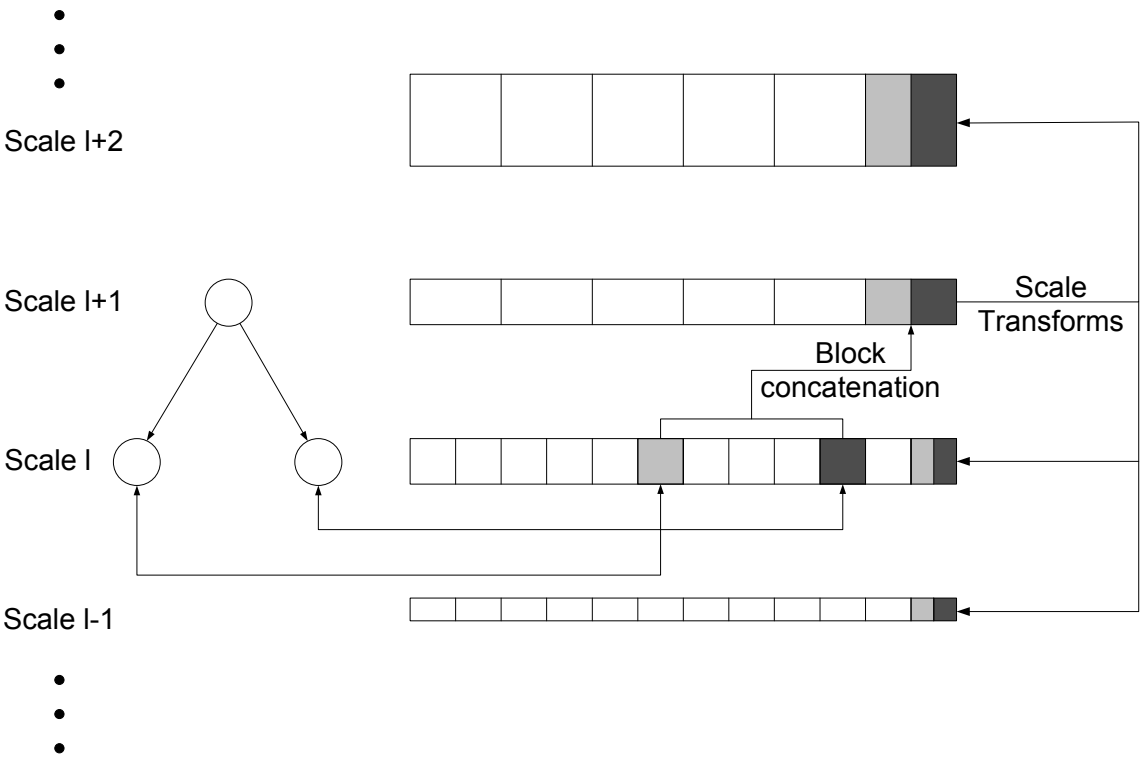


Figure B.9: Dictionary update scheme.

B.2.3 Dictionary update

The MMP's initial dictionary is very sparse, composed only by a set of homogenous blocks. The increase of its approximation power depends on the ability to generate new patterns.

Each time a block X^l of scale l is segmented, a new pattern is originated by concatenating the two dictionary blocks of scale $l-1$ used to represent the two halves, X_1^{l-1} and X_2^{l-1}. The new pattern is then used to update the dictionary at every scale, through the use of a separable scale transformation, T_l^s, that adjusts the dimensions of the generated block to those from each of the dictionary scales. The used scale transformation, T, is a straightforward implementation of traditional sampling rate conversion methods [113].

Figure B.9 represents the dictionary update procedure, resulting from the concatenation of two code-vectors of scale l. A new code-vector is originated on scale $l+1$, and the scale transformations are used to adjust its dimensions to that of the other scales. The new pattern thus becomes available on every scale from the adaptive dictionary.

It is important to notice that with this approach the dictionary adaptation process does not require extra overhead in the bitstream, as the decoder is able to keep a synchronized copy of the dictionary, based only in the information regarding the segmentation flags and the dictionary indices.

This scale adaptive dictionary update procedure is the key feature that distinguishes MMP from other pattern matching encoders. However, a careful analysis of experimental tests led to the development of several dictionary adaptation techniques that further

improved the performance of the algorithm. These originated an algorithm referred to as MMP-II [49, 114].

The first technique proposed in [49] targets the statistical probability distribution of the dictionary elements. Experimental tests demonstrated that the probability of a given dictionary block to be used on scale l depends on the scale where the block was originally created. A code-vector is most likely to be a good match for blocks with dimensions closer to that of the scale where the block came from, than for blocks with very different dimensions. From this observation, two considerations can be made:

- The inclusion of a scaled version of each new block in every dictionary scale conditions the statistical distribution of the dictionary indices, as they are unlikely to be useful and contribute to increase the entropy of all dictionary indices. Thus, the limitation of blocks insertion into scales with dimensions close to that of the original block is advantageous, as the lower dictionary approximation power is compensated by a lower entropy for the dictionary indices.

- Classifying the codewords in accordance to the scale where they were originally created can take advantage of this particular distribution, if one uses this information as a context in the arithmetic coder, thus reducing the overall entropy of the indices.

To take advantage from these considerations, the dictionary elements were organized into partitions, and each of these partitions only received code-vectors created on a particular scale. Additionally, the insertion of new blocks was restricted to scales whose dimensions in both directions are half or double those from the original scale. Each code-vector is then identified using its partition (context) followed by an index within that partition.

The second technique targets the improvement of the dictionary approximation power. For that purpose, geometric transforms and translations of the original block are created, and inserted into the dictionary. This includes 90^o, 180^o and 270^o rotations (see Figure B.10), symmetries relatively to the vertical and horizontal axis (see Figure B.11), the additive symmetric of the original block (see Figure B.12) and translations of half and a quarter block (see Figure B.13) [4, 49].

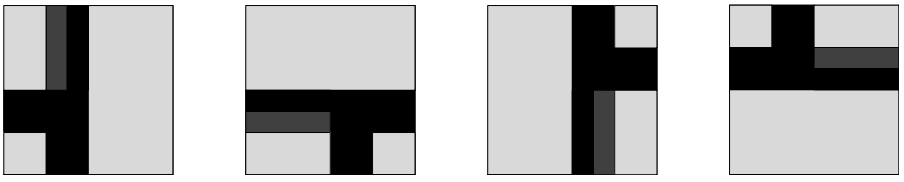
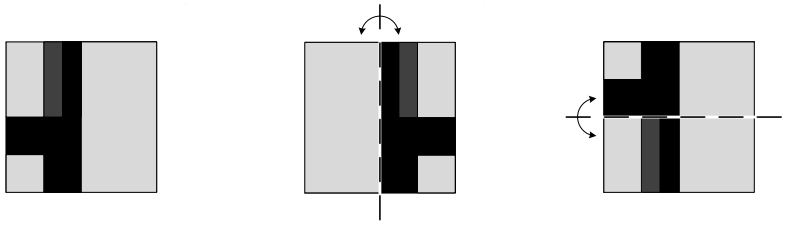


Figure B.10: New patterns created by rotations of the original block: (a) original, (b) 90^o, (c) 180^o and (d) 270^o rotations.



(a) (b) (c)

Figure B.11: New pattern created by using symmetries of the original block: (a) original, (b) vertical symmetry and (c) horizontal symmetry.

0	0	1	2
1	0	-1	1
0	1	0	2
-1	-1	-2	-2

(a)

0	0	-1	-2
-1	0	1	-1
0	-1	0	-2
1	1	2	2

(b)

Figure B.12: New pattern created by using the additive symmetric of the original block: (a) original and (b) additive symmetry.

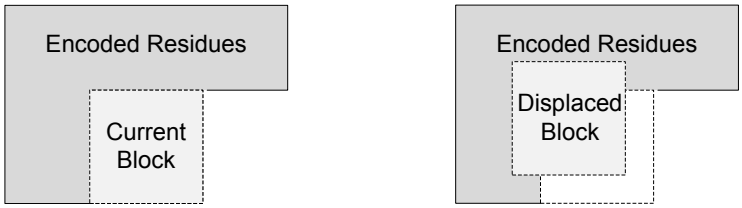
The main idea was to provide a richer set of patterns to the dictionary, but this approach has the important drawback of increasing the average entropy of its indices. Therefore, it is important to ensure that the generated patterns are likely to be useful, or otherwise, they will only contribute to increase the entropy of other indices.

For this reason, a third technique was proposed, consisting in a redundancy control scheme for the dictionary elements. The insertion of any new blocks in the dictionary is only allowed if its distance relatively to an existing code-vector is inferior to a given threshold d. This avoids the creation of new dictionary indices that bring little distortion gains relatively to other existing ones, which will increase the average entropy of the other symbols.

Figure B.14 graphically illustrates a generic case, with five code-vectors (S_1^l to S_5^l) present in the dictionary. A redundancy free region with radius d is created around each of these code-vectors, and new dictionary elements will not be inserted if they fall into any of these regions. This is the case for X^l, which falls into the region defined around S_4^l, that is then not inserted in the dictionary.

The threshold d was optimized using experimental tests. A direct dependency on the lagrangian operator λ was observed [4, 49], where the Equation B.12 was proposed to determine the threshold d as a function of the lagrangian operator λ:

$$d(\lambda) = \begin{cases} 5, & \text{if} \quad \lambda \leq 15; \\ 10, & \text{if} \quad 15 < \lambda \leq 50; \\ 20, & \text{otherwise.} \end{cases} \tag{B.12}$$



(a) (b)

Figure B.13: New patterns created by using displaced versions of the original block: (a) original and (b) quarter block diagonal translation.

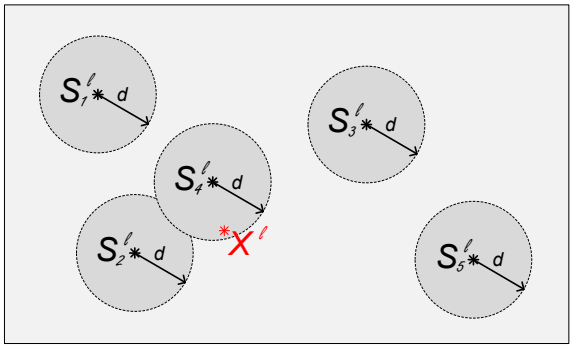


Figure B.14: Dictionary redundancy control technique.

Higher λs mean that the rate becomes more relevant than the distortion, and for that reason, the redundancy control tool needs to be more restrictive (higher values of d are used). As a higher distortion is tolerated and the rate is critical, less patterns are inserted in the dictionary to preserve a low average entropy for the indices. On the other hand, low λs correspond to low distortions, as higher bitrates are available. Thus, the redundancy control reduces the value of d, allowing more blocks to be included in the dictionary, that will improve the matching accuracy.

The fourth technique is a norm-equalization procedure that allows the algorithm to adapt the new code-vector patterns to the statistical distribution of the residue signal. When a block of scale l is subjected to a scale transformation that increases its dimensions, its norm is generally also increased. As the use of the predictive scheme has the particularity of generating residues highly peaked around zero, the scale transformations to higher scales and its consequent increase in the norm usually result in blocks that fall apart from this peaked distribution. This way, when a block is expanded, a norm equalization procedure allows to better fit its norm to the statistical distribution, resulting in a more accurate model for the existing code-vectors.

A detailed description of the MMP-II algorithm can be found in [49], together with a discussion of its computational complexity.

B.2.4 The MMP bitstream

Once the optimal segmentation tree $\mathcal{T}$ is obtained, it is converted into a string of symbols, using a top-down approach.

The hierarchical prediction scheme used in MMP allows to segment the prediction of a given block. This enables the use of different prediction modes on each of the resulting sub-blocks, as represented in B.8. Each of these independently predicted blocks will originate a corresponding residue block, which will be encoded using MMP. This procedure can originate further segmentations of the residue blocks, represented by a specific set of tree nodes.

In other words, two type of nodes exist in the segmentation tree, either indicating that the prediction is segmented, or indicating that only the residue block is segmented.

Therefore, five different flags are used to identify the different nodes that may occur in the segmentation tree:

- NS - The node is a tree leaf (the original block is not segmented);
- PV - The node corresponds to a vertical segmentation of both the residue and the prediction blocks;
- PH - The node corresponds to a horizontal segmentation of both the residue and the prediction blocks;
- RV - The node corresponds to a vertical segmentation of only the residue block;
- RH - The node corresponds to a horizontal segmentation of only the residue block.

When a vertical segmentation occurs, the subtree that corresponds to the left branch is first encoded, followed by the right branch sub-tree. Similarly, in case of horizontal segmentation, the algorithm starts by the upper branch sub-tree and follows to the lower branch sub-tree.

This way, the algorithm starts on the tree root, and keeps transmitting the segmentation flags that correspond to the successive tree nodes. When a node, where only the residue is segmented is reached, a RV or RH flag is transmitted, followed by the flag indicating the used prediction mode. The decoder is able to identify that the prediction needs to be reconstructed for the entire block before proceeding, in order to stay synchronized with the encoder. The algorithm then proceeds for the remaining sub-tree.

When a tree leaf is reached, the flag indicating that the block is no further segmented (NS) is transmitted, with the only exception for scale 0 (1×1 elementary blocks). In this case, there is no need to send this flag, as the node is obviously a tree leaf. After the non segmentation flags, there are two possibilities:

- If the prediction flag has not been sent for the pixels of the block, it will be transmitted at this point, followed by the index of the code-vector that should be used to approximate the corresponding block.
- If the prediction was already transmitted for these pixels, only the index needs to be transmitted.

As an example, the tree represented on Figure B.8 is encoded using the following string of symbols:

$$PV \quad RH \quad PredModeA \quad RV \quad NS \quad i_0 \quad RV \quad NS \quad i_1 \quad NS \quad i_2 \quad NS \quad i_3 \quad NS \quad PredModeB \quad i_4.$$

The generated symbols are then entropy coded using an adaptive arithmetic encoder [48, 115]. Independent probability models are used for each symbol type and segmentation tree level. Note that the segmentation tree level can be inferred directly from the tree node it corresponds to, both in the encoder and decoder.

For the case of the dictionary indices, the probability model of each symbol depends not only on the dictionary scale, but also on the original scale where each codeword was created. Thus, instead of simply transmitting the codeword's index using the probability of the index, conditioned to the knowledge of the block level (as on the original MMP algorithm [3]), we first transmit the scale where the codeword was created, conditioned to the knowledge of the block level. It is later used as a context for the index, jointly with the block level [49].

B.2.5 Computational complexity

Similarly to full search VQ algorithms, the biggest computational burden of the MMP algorithm is the optimal codeword index determination. This operation is similar to a full search vector quantization, whose complexity is typically given by $(2^m \times 2^n)S$, where $(2^m, 2^n)$ is the block dimension, and S is the number of elements present on the codebook.

In [116], the number of multiplications required by the matching procedure performed on a non-predictive MMP algorithm was derived for the case where a dyadic block segmentation is used. The same approach was adopted in [6] to derive the computational complexity of the MMP algorithm using both the dyadic and the flexible segmentation scheme. The number of multiplication operations necessary to encode one given block, using the original MMP algorithm (dyadic segmentation), was shown to be:

$$\mathfrak{C}_{\mathrm{MMP}}(2^m, 2^n) = (2^m \times 2^n) \times S \times (m + n + 1). \tag{B.13}$$

This equation is based on the fact that for an initial block size of $2^m \times 2^n$ pixels, the total number of dictionary scales was shown in Equation B.5 to be $1 + \log_2(2^m \times 2^n)$,

which is equal to $(1+m+n)$, the computational complexity derived on [116] is no more than the product of the complexity from a full search VQ algorithm for $2^m \times 2^n$ pixels blocks, by the number of different scales used on MMP.

For the case where the flexible partition scheme is used, a similar derivation also presented in [6] suggests that the computational complexity to encode one block, using the flexible segmentation scheme can be determined as:

$$\mathfrak{C}_{\mathrm{MMP-FP}}(2^m, 2^n) = \sum_{i=0}^{\max(m,n)} \sum_{j=0}^{i} \binom{i}{j} (2^m \times 2^n) \times S \times f(i,j), \tag{B.14}$$

where the function f is:

$$f(n) \triangleq \begin{cases} 1 & \text{if } m-(i-j) \geq 0 \text{ and } n-j \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{B.15}$$

Note that by the simple relaxation of the dyadic block division criterion, the computational complexity is severally increased. Nevertheless, since Equation B.14, which was presented on [6], only provides a pessimistic estimation of the algorithm's computational complexity, we will derive in this section the actual computational complexity of the MMP-FP algorithm.

In this new analysis, we note that successive segmentations frequently result in blocks with similar dimensions, which correspond to the same nodes of the segmentation tree. Thus, there is no need to perform several optimizations for these redundant nodes. For example, the vertical segmentation of a given 16×16 pixels block, followed by an horizontal segmentation of each of the halves, results in four 8×8 pixels blocks. This is also the case when a 16×16 pixels block is first segmented in the horizontal direction, and each half is then vertically segmented. As no dependencies exist while encoding a residue block, the four 8×8 pixels block resulting from the second situation do not require any extra computation, as their optimization was already performed earlier on the segmentation tree optimization process. This phenomenon becomes more evident at lower scales, which can be reached by a larger number of alternative paths across the segmentation tree.

Thus, each sub-block from the initial block only needs to be optimized once for each dictionary scale, as is the case for the original MMP algorithm (see Equation B.13). However, the flexible partition results in the increase of the total number of different scales, which was defined in Equation B.6 as $(m+1)\times(n+1)$, for an initial block size of $2^m \times 2^n$ pixels. Replacing the number of different scales possible on MMP-FP in Equation B.13, one may obtain the computational complexity of the MMP-FP algorithm without redundant nodes:

$$\mathfrak{C}_{\mathrm{MMP-FP}}(2^m, 2^n) = (2^m \times 2^n) \times S \times ((m+1)(n+1)). \tag{B.16}$$

The proof of Equation B.16 can be done by induction, similarly to the approach adopted on [6]. The formula holds for blocks of size (1×1), since the elements of the dictionary will be tested only once, that is:

$$\begin{aligned} \mathfrak{C}_{\mathrm{MMP}}(2^0, 2^0) &= (2^0 \times 2^0) \times S \times ((0+1)(0+1)) \\ &= S. \end{aligned} \tag{B.17}$$

Using the inductive hypothesis, the formula holds for blocks of dimension $(2^m, 2^n)$. For blocks of dimension $(2^{m+1}, 2^n)$, the algorithm needs to perform the extensive optimizations of the two $(2^m, 2^n)$ blocks which compose the original block, plus the optimization of all the non-redundant nodes, which correspond to those from dictionary scales with dimensions $(2^{m+1}, 2^i)$, with $(i = 0...n)$. Thus:

$$\begin{aligned} \mathfrak{C}_{\mathrm{MMP-FP}}(2^{m+1}, 2^n) &= 2 \times \mathfrak{C}_{MMP-FP}(2^m, 2^n) + \sum_{i=0}^{n} 2^{n-i} \times (2^{m+1} \times 2^i) \times S \\ &= 2 \times ((2^m \times 2^n) \times S \times ((m+1)(n+1))) \\ &\quad + \sum_{i=0}^{n} (2^{m+1} \times 2^n) \times S \\ &= (2^{m+1} \times 2^n) \times S \times (m \times n + m + n + 1) \\ &\quad + (2^{m+1} \times 2^n) \times S \times (n+1) \\ &= (2^{m+1} \times 2^n) \times S \times ((m \times n + m + n + 1) + (n+1)) \\ &= (2^{m+1} \times 2^n) \times S \times (m \times n + m + 2n + 2) \\ &= (2^{m+1} \times 2^n) \times S \times (((m+1)+1)(n+1)). \end{aligned} \tag{B.18}$$

The induction procedure when one considers the other coordinate is entirely analogous. It is important to notice that the computational complexity calculated using Equation B.18 is considerably lower than the value obtained using Equation B.14, for a given initial block size.

If a prediction scheme is adopted, and MMP is used to compress the generated residues, the residue optimization tree reuse is not possible between prediction modes, since for each prediction the residue might be different. Thus, if M prediction modes are used, and considering only prediction at the highest block scale, the computational complexity from Equation B.16 becomes:

$$\mathfrak{C}_{\mathrm{MMP-FP}}(2^m, 2^n) = M \times (2^m \times 2^n) \times S \times ((m+1)(n+1)). \tag{B.19}$$

If a hierarchical prediction is used, all the prediction modes will be tested for each of the block dimensions used for prediction. In this case, there are no redundant nodes in the prediction level, as each path across the segmentation tree imposes a different encoding for the block's neighborhood, and consequently, the prediction for the block may differ.

Thus, all the combinations of block partitions may be optimized on the hierarchical prediction stage. Nevertheless, when encoding resulting residues, the redundant node does not need to be encoded, such as considered on Equation B.16.

B.3 Experimental results

In this section, we present a performance evaluation of the MMP algorithm, that use all the techniques described on the previous sections. This algorithm will be referred to as MMP-FP. The results are evaluated against that of two well known state-of-the-art image encoders, namely JPEG2000 [50] and H.264/AVC high profile intra frame encoder [45, 51].

JPEG2000 [50] is a lossy and lossless image coding standard based on wavelet transforms. It uses the CDF 9-7 transform for lossy compression, and the CDF 5-3 transform for lossless compression. The simple Mallat structure is used for the subband decomposition. After the transformation, the wavelet coefficients are quantized using a scalar dead-zone quantization, and then compressed by an arithmetic entropy encoding, using the binary MQ-coder. The JPEG2000 is commonly used as a state-of-the-art reference for the performance of wavelet-based encoders. The results presented in this thesis for JPEG2000 were obtained using the KAKADU software [117].

Despite being conceived as a video compression standard, many of the coding advances brought into H.264/AVC [45] have made this method not only a new benchmark for video compression but also a very efficient compressor for still images [54, 118]. The still image coding is performed on a block-based approach, using a predictive scheme and a discrete cosine transform to encode the generated residue. It also incorporates a deblocking loop filter, that can minimize undesired blocking artifacts, which was enabled in our experimental tests. We adopted the FRExt high profile configuration in our experimental tests, as it achieves the best coding performances [54]. The results presented in this thesis were obtained using the JM reference software [80].

B.3.1 Objective performance evaluation

For the objective performance evaluation, we considered the peak signal-to-noise ratio (PSNR) as a function of the compression ratio, measured in bits-per-pixel (bpp) [119]. This quality evaluation measure is given in decibels (dB), and can be defined as:

$$PSNR = 10log\frac{(2^n - 1)}{MSE}, \tag{B.20}$$

where n represents the number of bits used to represent each sample, and MSE is the mean squared error between the two signals. When working with images, the MSE can

be defined as:

$$MSE = \frac{1}{M \times N} \sum_{j=1}^{M} \sum_{i=1}^{N} (X_{(i,j)} - \hat{X}_{(i,j)})^2, \tag{B.21}$$

where M and N are the dimensions (in pixels) of original image, X is the original image, and $\hat{X}$ its noisy reconstruction.

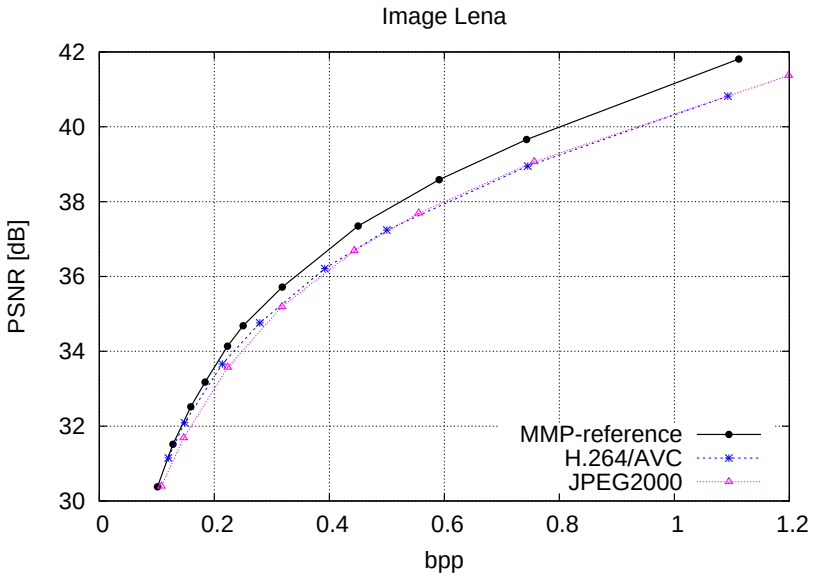


Figure B.15: Experimental results for natural image Lena (512×512).

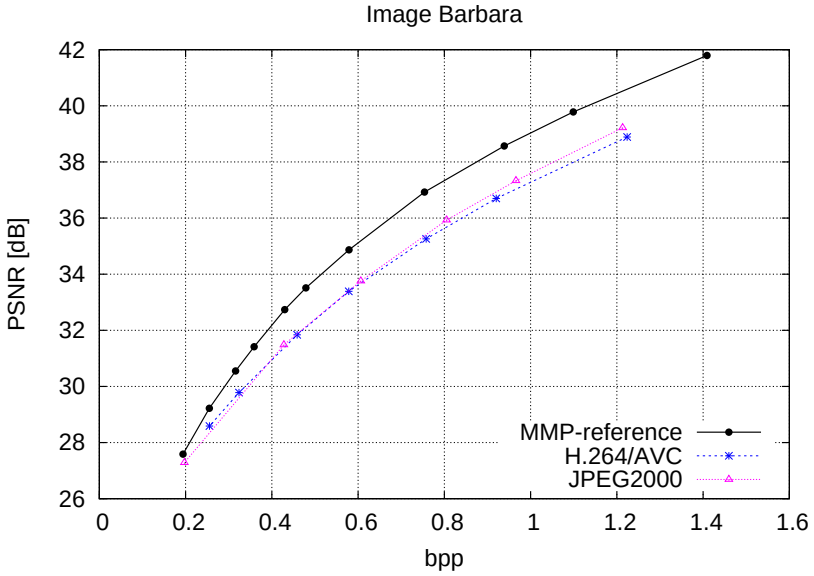


Figure B.16: Experimental results for natural image Barbara (512×512).

A set of four images with different characteristics was selected for the objective quality comparison. Natural images Lena and Barbara have been extensively used on the image processing and compression literature, and were chosen due to their particular features. Image Lena has a strong low-pass nature, with only few details concentrated on

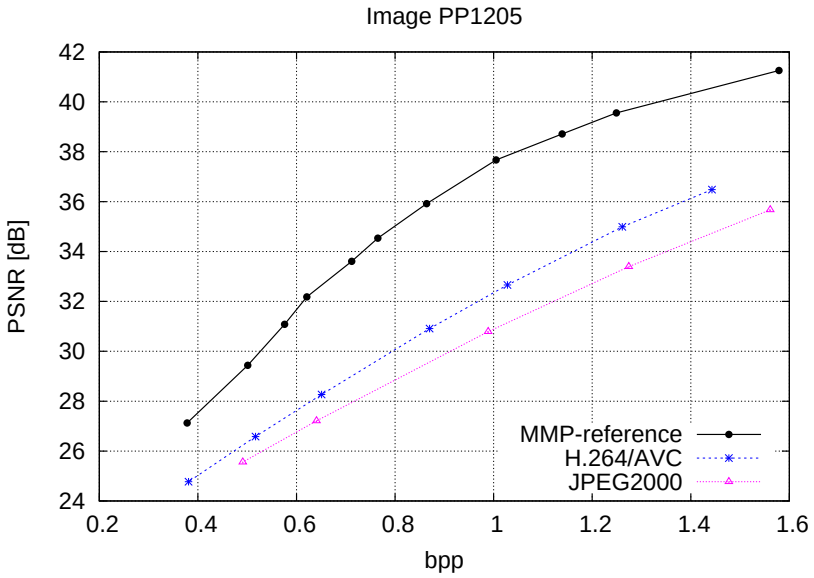


Figure B.17: Experimental results for text image PP1205 (512×512).

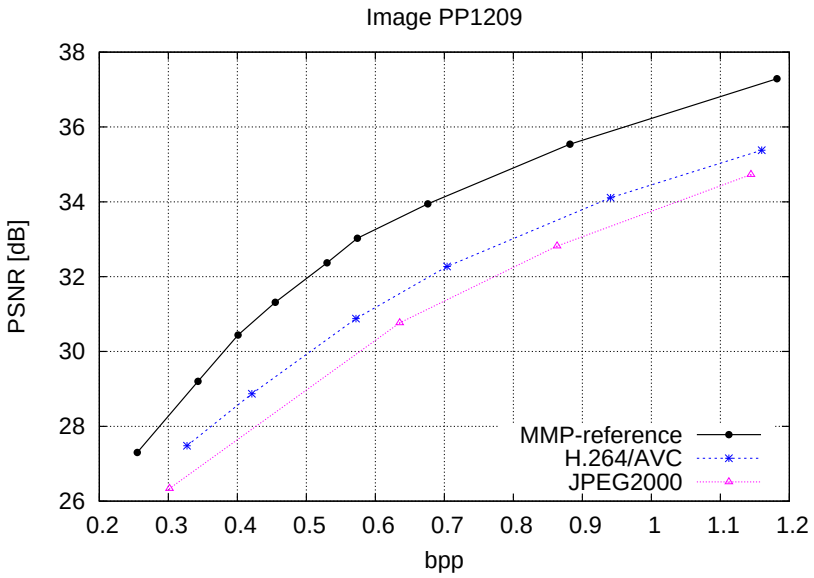


Figure B.18: Experimental results for compound image PP1209 (512×512).

limited regions. For this reason, the transform-based encoders tend to be particularly successful in its compression. Image Barbara presents relevant high-frequency components, with more detailed regions across the entire image, which are less efficiently exploited by transform-based encoders. We also include a scanned text image (PP1205) and a scanned compound image (PP1209), to further evaluate the algorithms for other image types. These images were scanned respectively from pages 1205 and 1209 of the *IEEE Transactions on Image Processing*, volume 9, number 7, from July of 2000, and were chosen because they have been used on several MMP-related previous publications. All this images are presented on Appendix I, and have 512×512 pixels.

Figure B.15 shows the rate-distortion performance comparison for image Lena. It can

be seen that the H.264/AVC coder is able to outperform JPEG2000 by up to 0.35dB at high compression ratios, with the two encoders presenting an equivalent performance for lower compression ratios. MMP-FP is able to outperform both algorithms by up to 0.9dB. The performance of MMP is close to that of H.264/AVC at high compression ratios, but the gains increase for lower compression ratios.

For image Barbara, the gains presented by MMP become even more noticeable (Figure B.16). The advantage is up to 1.8dB in relation to H.264/AVC and 1.4dB in relation to JPEG2000. For this image, H.264/AVC outperforms JPEG2000 for high compression ratios, but this tendency is inverted for medium-to-low compression ratios. MMP is consistently better than the transform-based encoders for the entire range, with the rate-distortion performance advantage increasing for lower compression ratios. This is so because the highly detailed regions present in this image impose a considerable degradation on the reconstruction's quality if a coarse quantization is applied.

The advantage of MMP becomes even more obvious for text images, as seen on Figure B.17, for image PP1205. The sharp edges of the characters result in a scattering of the energy to higher frequency coefficients, and transform-based encoders are not able to efficiently exploit this energy distribution. H.264/AVC is more efficient than JPEG2000 while dealing with this images, with an advantage of up to 2dB. However, both encoders are considerably outperformed by MMP. Gains are up to 5dB and 7dB, when compared to H.264/AVC and JPEG2000, respectively.

Figure B.18 presents the results for compound scanned image PP1209. As it can be seen, MMP also considerably outperforms both H.264/AVC and JPEG2000 for the entire range, with gains up to 2dB and 3dB, respectively.

The presented results show that the successive improvements of the original MMP algorithm [3] for still image coding allowed to reach a state-of-the-art rate-distortion performance for still image compression applications.

B.3.2 Observation of subjective quality

The objective performance evaluation is important since it provides an unequivocal measure of the encoder's compression efficiency, but it fails in demonstrating the visual quality of the reconstructions. A higher objective quality is not a guarantee that the perceptual quality is also better, as in some cases particular artifacts that do not seriously degrade the objective performance can be particularly annoying for the human visual system.

In Figure B.19, we present a detail from image Barbara, compressed respectively using MMP, H.264/AVC and JPEG2000. An high compression ratio was chosen, in order to enhance the particular artifacts typically introduced by each encoder. A target bitrate of 0.25bpp was chosen for the comparison, resulting in a compression ratio of 1:32.

From Figure B.19b, it can be seen that the major issues with the reconstruction ob-

(a) Original 8bpp (b) MMP 29.22dB

(c) H.264/AVC 28.56dB (d) JPEG2000 28.25dB

Figure B.19: Subjective comparison of detail from natural test image Barbara (512×512) coded at 0.25bpp.

tained from MMP are the blocking artifacts. These artifacts are common in block based encoders, and can be attenuated using post-filtering techniques, as we will discuss on Appendix F. The H.264/AVC algorithm is an example of block based algorithms that use deblocking filtering to reduce these artifacts. It can be seen that the reconstruction, shown on Figure B.19c, does not suffer from blocking artifacts. However, the aggressive filtering resulted on some blurring in the most detailed regions. As a consequence, the detail on the scarf, for example, was more affected than in MMP's reconstruction. Additionally, some ringing artifacts were also introduced (for example in the scarf, close to the shoulder). The ringing artifacts become even more evident on the image compressed using JPEG2000 (Figure B.19d). In this case, as JPEG2000 is not a block based encoder,

blocking artifacts are not present in the reconstruction, but both blurring and ringing artifacts are noticeable in the most detailed regions.

B.4 Conclusions

In this appendix, we described the multidimensional multiscale parser algorithm (MMP), the basis of the compression frameworks for still image and video compression discussed on this thesis.

A detailed description of the algorithm was presented on Section B.2, focused on two-dimensional signals, and on the improvements that contributed to the current state-of-the-art compression performance for still image coding. These improvements have been mainly oriented towards natural images compression, where earlier versions of MMP were not as efficient as transform-based encoders.

The improvements focused on the optimization of the adaptive dictionary and block segmentation, as well as on the introduction of a predictive scheme, which was further refined with the adoption of a more sophisticated prediction mode: the Least Squares Prediction (LSP).

In Sections B.3.1 and B.3.2, we performed an objective and subjective performance evaluation of the algorithm, comparing its experimental results with those from two state-of-the-art transform-based encoders, JPEG2000 and H.264/AVC. A superior objective performance for a wide range of input images types was demonstrated, as well as a good subjective reconstruction quality when compared with other encoders. The subjective evaluations were also oriented towards identifying the most relevant artifacts introduced by MMP, in order to identify possible improvements that can be performed on the encoder.

The results presented in this appendix demonstrate the high coding efficiency of MMP, as well as its high degree of adaptability, justifying its adoption for further researches on still image and video compression.

Appendix C

Compound document encoding using MMP

C.1 Introduction

The increasing relevance of digital media support for document transmission and storage justifies the need for efficient coding algorithms for this type of data. Traditional paper media is being replaced by digital versions, with the advantage of avoiding the large storage and preservation requirements associated with the paper versions, while making the documents easily available for a larger number of users.

An important part of this process is the scanning of paper documents. However, the generation of a large number of scanned document images arises the problem of efficiently coding them. A straightforward approach is to encode such images using traditional state-of-the-art image encoders, like SPIHT [52], JPEG [53], JPEG2000 [50] or H.264/AVC Intra [45, 54]. However, despite the efficiency of these algorithms for smooth, low-pass images, they are not capable of achieving a satisfactory performance for non-smooth image regions, like the ones corresponding to text or graphics, frequently present in scanned documents.

For smooth images, most of the transform coefficients representing the highest frequencies are of little importance, allowing their coarse quantization. This leads to a high compression ratio without compromising the perceptual quality of the reconstructed images. However, when the input image does not present a low-pass nature, the coarse quantization of these high-frequency coefficients results in highly disturbing visual artifacts, like ringing and blocking.

An alternative is the use of encoding methods specifically developed for text-like image coding, like JBIG [55]. Unfortunately, such algorithms tend to present serious limitations when used to encode smooth regions of compound images. One reason is the fact that text and graphics images usually require high spatial resolution to preserve the

document's readability. On the other hand, they do not require high color depth, since characters and other graphic elements usually assume only a few distinct colors over a solid background color. With natural images, the opposite tends to happen: due to their high correlation among neighboring pixels, they usually do not require high spatial resolution in order to maintain a good subjective quality, but often require high color depth.

Therefore, methods that are able to efficiently compress both pictorial and textual regions are of particular interest for compound document encoding, where smooth image regions coexist with text and graphics.

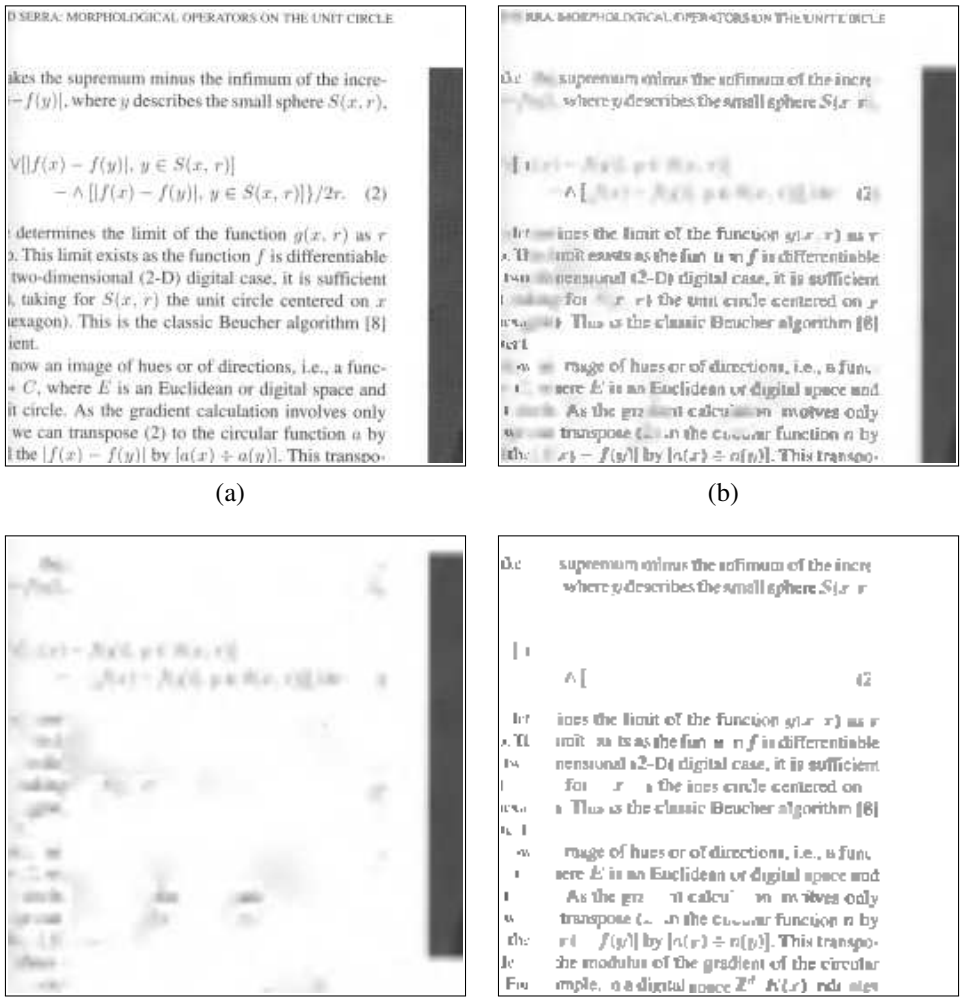
Several algorithms, like Digipaper [56], DjVu [57, 58], JPEG2000/Part6 [59], among others [60, 61], have been proposed for compound document compression. They adopt the MRC (Mixed Raster Content) model [62] in order to decompose the original image in several distinct layers [120].

A *background* layer represents the document's smooth component, including natural images regions and other smooth objects, as well as the paper's texture. A *foreground* layer contains the information regarding the text and graphics colors. One or more binary segmentation masks containing text and graphics shape information may also be used to blend the information of both layers. These distinct layers can usually be compressed individually in a much more efficient way than if we use a single encoder for the entire compound document, resulting in higher compression ratios and better subjective quality for the reconstructed image.

Despite the popularity of the MRC model for compound image compression, it presents some limitations. For example, it is based on the assumption that the segmentation process can accurately separate the text and graphics regions, which is not always true. For synthetic documents, where the character bounds are well defined, such a segmentation can be quite effective, but it loses effectiveness when the documents' complexity increases. Errors in pixel classification usually compromise the overall efficiency of the compression scheme.

The main objective of the work presented in this appendix is to develop an MMP based method to efficiently compress scanned compound documents. Such documents are usually originated by scanning book or magazine pages that contain both textual and pictorial contents. Unlike synthetic or computer generated documents, these images cannot be easily segmented into foreground and background objects; as a consequence, state-of-the-art compound document encoders tend to present poor results.

Figure C.1 illustrates the segmentation-related artifacts that may appear when DjVu [57] is used to compress scanned compound document SCAN0002 (see Figure I.6 of Appendix I). Figure C.1a presents a detail of the original image. The scanning process degrades the characters' crisp edges, causing their erroneous inclusion on the background layer (Figure C.1c). As this layer is coded using a wavelet based algorithm, the coarse quantization of its high frequency coefficients results in illegible characters in the text, as



(a) (b)

(c) (d)

Figure C.1: a) Detail from image SCAN0002 b) resultant reconstruction with DjVu at 0.31bpp: c) Background layer; d) Foreground layer.

can be seen on the reconstructed image presented on Figure C.1b.

This is an important drawback, as it can compromise the legibility of the entire document. Note that adjusting the segmentation threshold in order to successfully identify all the characters is not a good solution either. This is so because such a task would have to be performed independently for each document, favoring the introduction of artifacts in pictorial regions, as illustrated on Figure C.2.

Figure C.2a presents another detail corresponding to a pictorial region of image SCAN0002, with its generated reconstruction at 0.31bpp represented on Figure C.2b. It can be seen that high contrast artifacts are introduced on some regions of the natural image, that contain sharp edges, such as the bottle label. Figure C.2d presents the encoded foreground layer, where one can see that artifacts were introduced because these high frequency regions were erroneously classified as foreground and further binarized. For this reason, it is not straightforward to use standard compound document compression algorithms based on MRC decomposition for this type of applications.

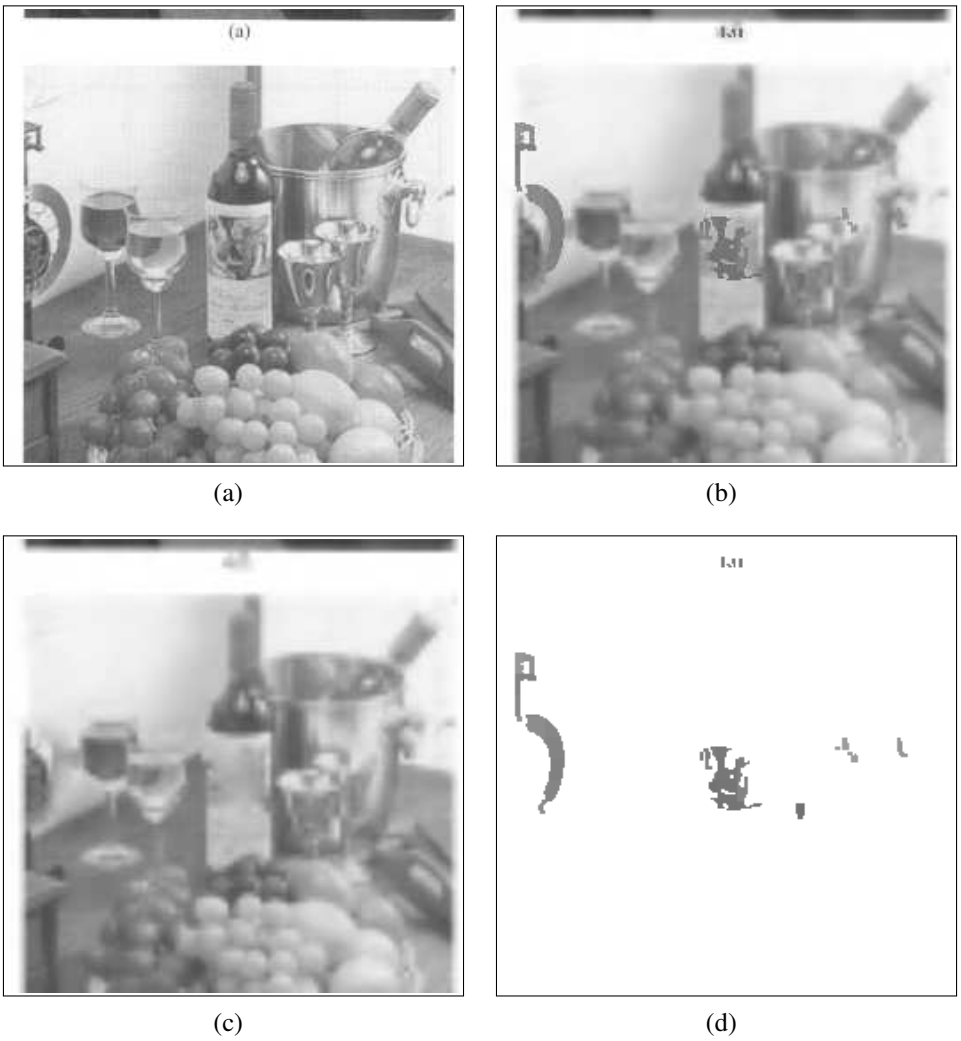


Figure C.2: a) Detail from image SCAN0002 b) resultant reconstruction with DjVu at 0.31bpp: c) Background layer; d) Foreground layer.

In this appendix, we introduce a novel compound document encoder based on the Multidimensional Multiscale Parser algorithm [3, 49]. The relevant results presented by MMP both for smooth and text image coding, as shown on Appendix B, suggested that it might have high potential to encode compound images. This motivated the development of a new algorithm for this particular application.

A hybrid architecture was adopted, with a block classification scheme used to separate pictorial macroblocks from text and graphics ones. Each of these two types of macroblocks are then encoded using a different version of the MMP algorithm, specifically tailored to their particular characteristics. The high degree of adaptiveness presented by MMP can be particularly useful for this application, contributing to make the resulting codec less sensitive to errors in block classification, that are an important source of inefficiencies in conventional MRC-based algorithms, or other block-based algorithms [65, 66].

Simulation results show that the proposed algorithm, while having state-of-the-art results for compound documents, still consistently outperforms transform-based encoders

for smooth images.

The remaining of this appendix is organized as follows. In Section C.2, we discuss the implementation of a hybrid compound document encoder based on text/graphics optimized algorithm and on state-of-the-art MMP approach for still image compression. The experimental results assessing the proposed methods are presented in Section C.3, while the conclusions regarding the developed algorithm are stated in Section C.4.

C.2 MMP for compound image coding

In this section we describe the proposed scanned compound document compression algorithm, which relies on the decomposition of the input document into smooth (pictorial) and non-smooth (textual) blocks, to be compressed separately using two different MMP-based encoders, named MMP-FP and MMP-text, respectively. Each of these encoders was specifically tailored to take advantages of some particular features observed on these image regions.

C.2.1 Architecture

As MMP is a block-based encoder, the segmentation process is also performed using a block-by-block basis, through the analysis of the gradient of each of the input blocks. Thus each block is classified as either pictorial or textual, and this information is signaled to the decoder using a binary mask.

Block-based segmentation has been proposed in the literature as an alternative to layer-based segmentation [63–68]. For example, in order to avoid the potential information leakage, layer-based encoders have to address the issue of coding partially masked foreground and background blocks, a problem not present in block-based approaches. Ideally, in layer-based segmentation, the masked data should not generate extra information to be transmitted; however, in practice, some sort of padding of the partially masked blocks should be performed in each layer. Some algorithms have been proposed in order to minimize this type of redundant information transmission, such as data filling [69, 70] and successive projection [71]. These algorithms are effective in alleviating this problem, but can only provide suboptimal solutions.

Another interesting property of block-based approaches is that, as the segmentation mask is signaled in a block-by-block basis, the overhead of transmitting it is much smaller than in the layer-based segmentation approach. For example, in our experiments we used blocks of dimensions 16×16 that yield a decrease of this overhead by a factor of 256, resulting in less than 0.004 bpp to transmit the mask.

The approach used by the MMP-compound algorithm is summarized in Figure C.3. First, the input image is analyzed, in order to classify each block as a smooth (pictorial) or

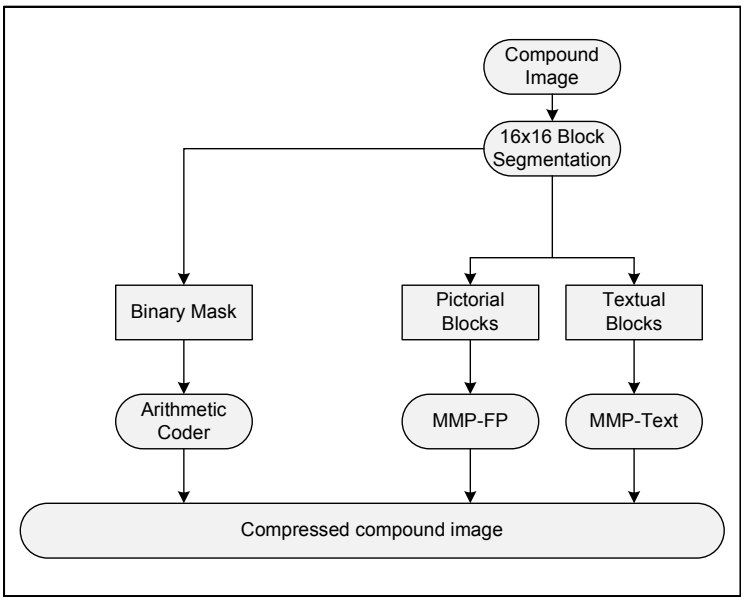


Figure C.3: MMP-compound compression scheme.

text and graphics (textual) block. The adopted segmentation method, originally proposed in [41], is basically performed in three steps, which are described in the next section.

C.2.2 Segmentation procedure

The segmentation procedure first starts by applying to the input compound document morphological grayscale top-hat and bottom-hat filter operators [72], in order to attenuate variations in the background of text regions, as well as to enhance the contrast of the foreground objects. A 7×7 pixels structuring element is used for this purpose. Two enhanced images are obtained: the one generated by the bottom-hat operator allows to identify dark foreground objects over a bright background, whereas the one obtained by the top-hat operator allows to identify bright objects over a dark background.

A block-based classification algorithm, based on [73], is then applied to the enhanced images. For the top-hat and bottom-hat images, the horizontal and vertical gradients of each 16×16 block are computed. Two thresholds are applied to the absolute value of these gradients in order to classify its pixels variations as low- medium- or high-valued. The lower threshold was set on 10, while the higher was set on 35, considering a grayscale document with 8 bits depth resolution (pixel values from 0 to 255).

Pictorial blocks tend to have low- to medium-valued gradient pixels in both directions, while the gradient pixels of textual blocks tend to be medium- to high-valued.

The pixels of each type are counted and the result is used as the input of the flowchart presented in Figure C.4, with Th1 set to 60% and Th2 set to 1% of the number of gradient pixels in a block.

Two classification masks are created with this procedure, one as the result of the pro-

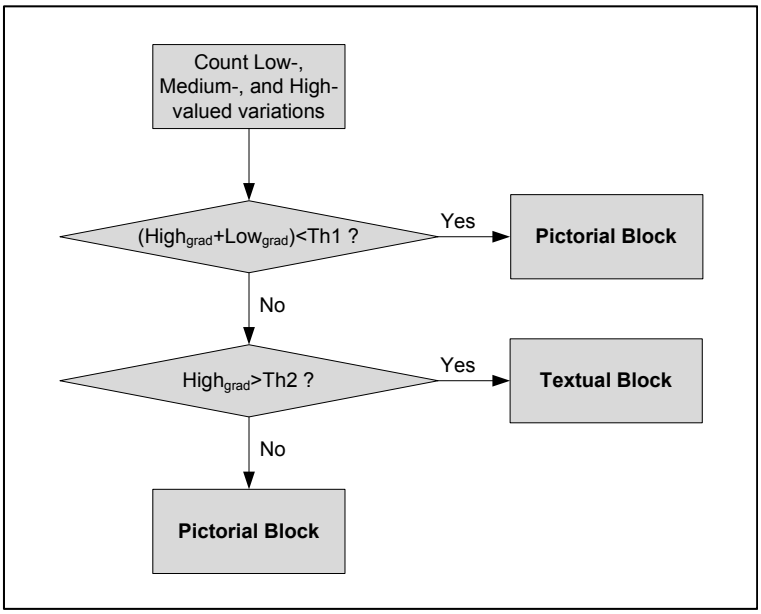


Figure C.4: Flowchart of gradient based algorithm.

cessing of the bottom-hat image and another of the top-hat image. A block is classified as a text block if it is a text block in at least one of the masks.

With the above segmentation process, some pictorial image blocks with high-pixel variations can still be misclassified as textual blocks. We alleviate this problem by refining the obtained segmentation, based on the detection of connected components in the image. This procedure, based on [74] is described in detail in [41].

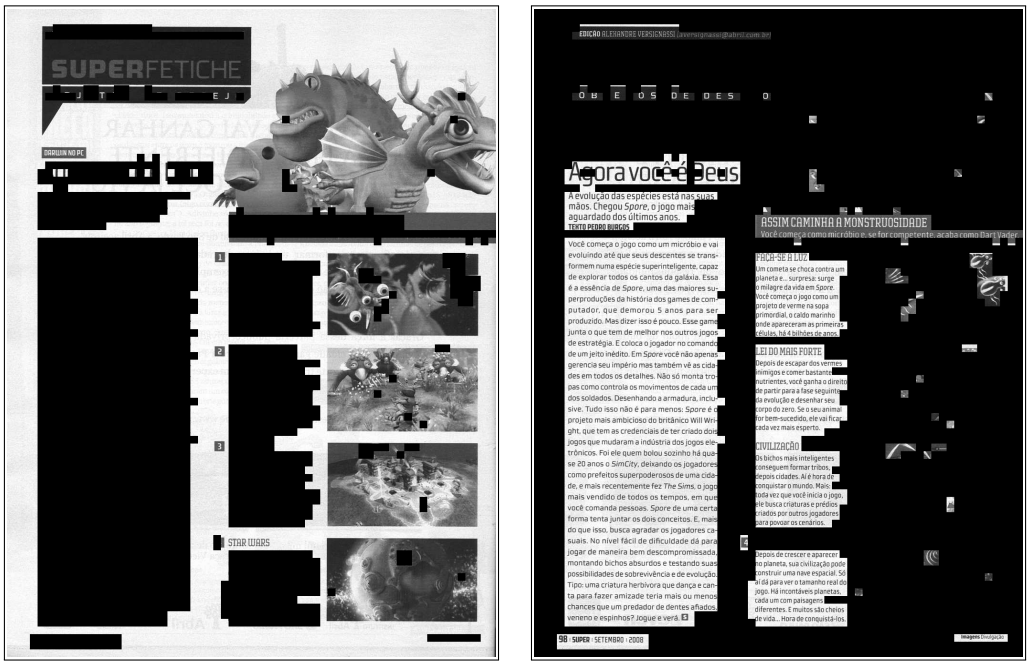
An important advantage of the presented segmentation algorithm is that, unlike most MRC based algorithms, it does not require any parameter adjustment when the input image varies. The proposed encoder, generated by the combination of MMP with the above segmentation method, presents a robust performance for a wide range of compound document types, with no need for human intervention. It is important to note that the adaptivity of the MMP-based encoders gives an important contribution for this robustness. This is so because it greatly attenuates the effect of small variations of the segmentation mask on the algorithm's rate-distortion performance.

Figure C.5 shows the decomposition of image Spore into its pictorial and textual components (the original image is presented in Figure I.9 from Appendix I).

C.2.3 Binary mask encoding

The segmentation procedure results on a binary classification mask, which relates each block of the input image, respectively, to the textual or pictorial component.

In Figure C.6b, we present the generated mask for image Spore, shown on Figure C.6a. In this particular case, the image has a resolution of 1024×1360 pixels, resulting in a mask with 64×85 pixels, when 16×16 pixels blocks are used. Only one bit is required



(a) (b)

Figure C.5: Image Spore a) natural component and b) text and graphics component.

to identify if each block belongs to the pictorial or to the textual group, so the mask can be transmitted using 5440 bits, without using any kind of compression.

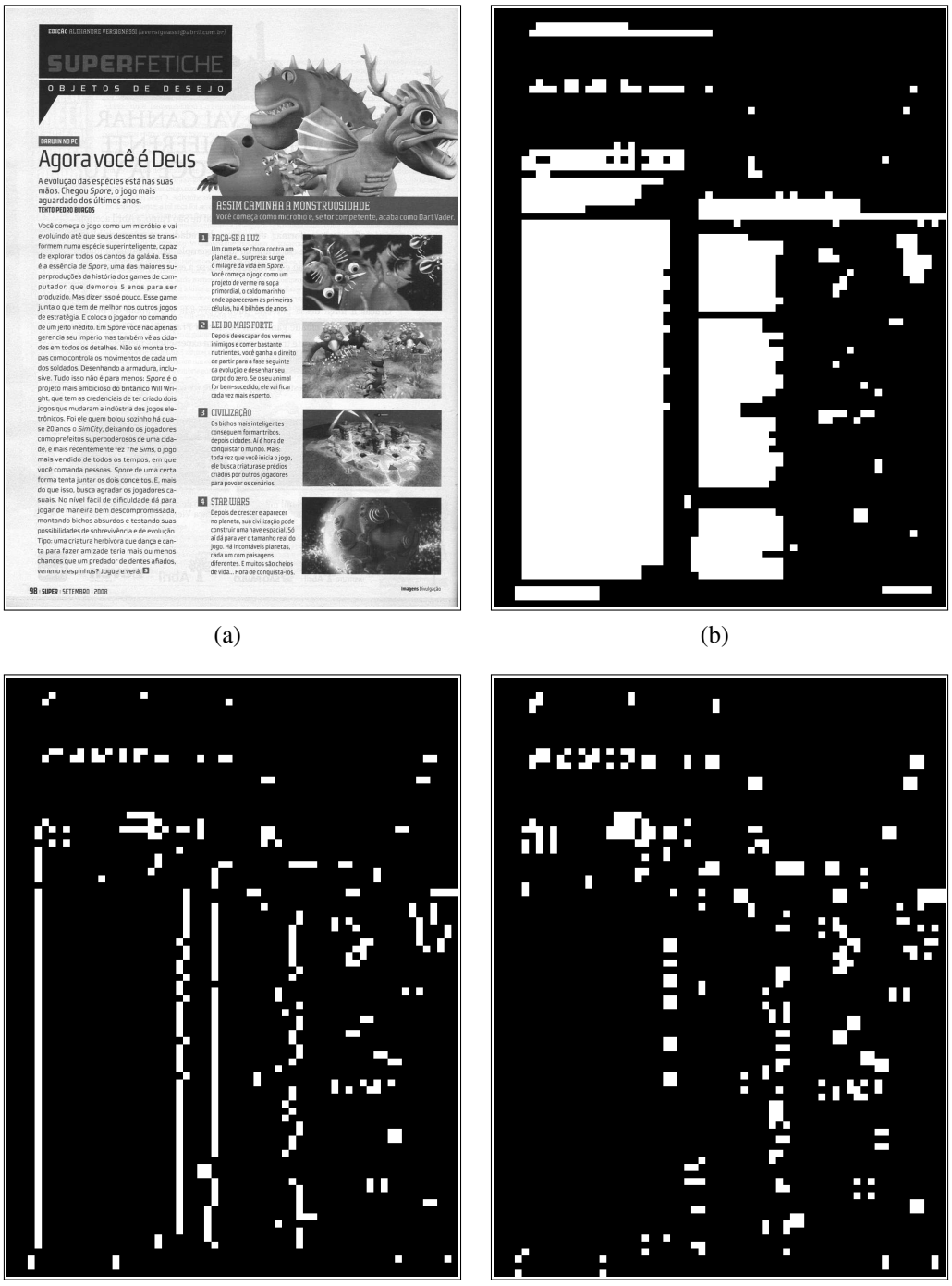
Instead of encoding the classification flags directly, we encode the changes in the flag value from the previous block in a raster-scan order. This procedure is efficient, since blocks with similar classification tend to occur in clusters.

However, it is reasonable to assume that the number of pictorial and textual blocks will be different, depending on the document that is being encoded. This indicates that the use of an adaptive arithmetic encoder [115] can be effective while reducing the final overhead.

Analyzing Figure C.6b, one can see that pictorial blocks are most likely to occur than textual blocks. In this particular case, 3532 of the 5440 blocks are smooth and 1908 correspond to text and graphics regions. This resulted on an average entropy of 0.935 bits for each binary symbol of the mask, reducing the final overhead.

Figure C.6c shows the differential mask obtained directly from Figure C.6b. In a raster scan order, the first block of each line is considered to be smooth by default. The value '0' is transmitted every time the next block has the same type as the current one, and the value '1' is transmitted every time a transition occurs. If the first block is a text block, the first pixel of the mask will have the value '1'.

For the case illustrated in Figure C.6c, only 353 symbols indicate changes relatively to the previous block flag, and 5087 symbols indicate that the the block is of the same type of its predecessor. With this approach, the average entropy of each symbol decreases to



(a) (b)

(c) (d)

Figure C.6: Image Spore a) original, b) generated mask, c) horizontal differential mask and d) horizontal and vertical differential mask.

0.347 bits per symbol, that corresponds to an overhead close to 0.001 bpp. Note that the decoder is able to reconstruct the original mask using the differential information from the differential mask represented in Figure C.6c. For this particular case, a compression ratio of approximately 3:1 on the mask representation is reached.

A similar approach could further be applied in the vertical direction, in order to exploit the remaining redundancy of the binary mask. The result is the mask presented in Fig-

ure C.6d, with 324 symbols indicating change on the block type and the remaining 5116 flags indicating that the current flag is the same type as the previous one. The average entropy decreases in this case to 0.215 bits per symbol. The coding efficiency improvement is more modest in this case, as the horizontal differential mask was able to exploit most of the correlation between the flags. In masks with several isolated blocks, or with several small clusters, this scheme is even likely to decrease the coding performance. This way, only a horizontal differential model is used, resulting in a good compromise for most cases. Furthermore, the overhead introduced by the proposed method is already almost negligible, and additional gains while coding the mask have almost no impact in the overall encoding performance of the proposed method. Hence, the adoption of more complicated compression schemes for the mask has irrelevant gains.

C.2.4 MMP for text images: MMP-Text

As the objective was the development of an MMP-based hybrid compound document encoder, it became relevant to separately optimize the encoder in accordance to the characteristics of each of these distinct regions. In order to optimize MMP for efficient coding of textual regions, some modifications are proposed for the MMP-FP algorithm. We refer to the resulting MMP-based codec as MMP-text.

We start by investigating the effectiveness of predictive coding for non-smooth image coding. The experiments carried out showed that the prediction is of little utility for text images, while requiring a significative increase in the computational complexity of the algorithm.

High frequency transitions compromise the accuracy of the prediction stages, resulting in residue blocks with an energy level close to that of the original block. This is illustrated in Figure C.7.

The use of a hierarchical prediction scheme presents in this particular cases many disadvantages. First, the algorithm needs to test exhaustively all the prediction modes. As none of them turns out to be particularly useful, this only contributes to increase the algorithm's computational complexity. Additionally, the overhead associated to the prediction mode transmission and the additional segmentation flags, which are required to identify the prediction segmentation pattern, is not compensated by a corresponding effective decrease in distortion of the reconstructed image. The fact that none of the prediction modes work well for text images, has also a negative impact in the arithmetic encoder's adaptation process. In natural images, where a high level of spatial correlation exists, the best prediction modes for each block tend to be correlated with the one from its neighbors. Thus, the arithmetic encoder is able to adapt to the statistical distribution of the prediction modes and segmentation patterns used for each region, reducing the amount of bits needed to encode this information. For text images, the choice of the prediction mode and

(a) (b) (c)

Figure C.7: Detail from image PP1205 a) original, b) prediction generated and c) residue to be coded.

segmentation pattern becomes more arbitrary, due to the lack of statistical correlation, restraining the ability of the arithmetic encoder to adapt to statistics of the input signal and thus reduce the entropy of the transmitted symbols.

This way, the overhead associated to the prediction mode transmission and the additional segmentation flags, added with the rate required to encode the residual block, tend to considerably exceed the number of bits spent to directly encode the image block when a non-predictive approach is used. Furthermore, it is important to notice that this difference has the tendency to increase with the use of the flexible segmentation, as the prediction segmentation possibilities increase in this case, with an obvious increase in the associated signalizing overhead.

This led us to investigate the elimination of the prediction stage for MMP-text.

Some of the dictionary optimization techniques introduced by MMP-II [49] were also re-evaluated, since the dictionary is now used to store original image blocks, instead of residue signals that have different dynamic range and statistical distributions. The optimal hypersphere radius associated with the redundancy control for the MMP-II algorithm was recalculated for the new algorithm. A procedure similar to that described in [49] was adopted, were a large set of text images was compressed in order to determine an optimized empirical model for the distortion radius. A heuristic observation of the experimental results resulted in the following function:

$$d(\lambda) = \begin{cases} 5, & \text{if} \quad \lambda \leq 15; \\ 20, & \text{if} \quad 15 < \lambda \leq 50; \\ 30, & \text{otherwise.} \end{cases} \tag{C.1}$$

The norm equalization procedure and the dictionary updating with the symmetric block [49] were also removed, since they were specifically oriented to work with residue blocks.

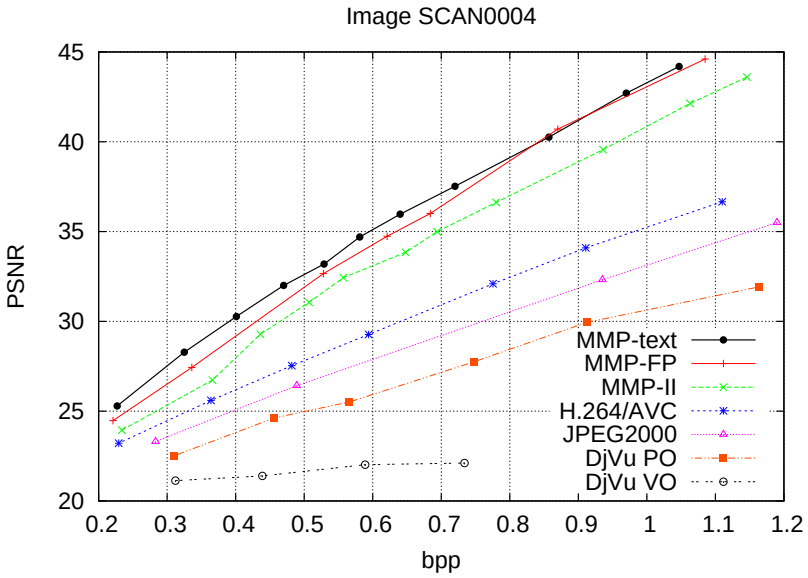


Figure C.8: Experimental results for text and graphics images Scan004 (512×512).

Experimental tests were then conducted in order to evaluate the performance of the new method, when compared with the one from MMP-FP and another state-of-the-art algorithms. A set of scanned grayscale text and graphics images was selected for that purpose. Since the proposed method does not compress text and graphics as binary images, we have compared MMP with the state-of-the-art continuous-tone image encoders, H.264/AVC [45] and JPEG2000 [50], due to their top PSNR performance. In addition, since we are dealing with compound documents, results for DjVu are also presented, for two distinct sets of encoding parameters. We adopted this approach because DjVu encodes text and graphics as binary objects, resulting, usually, in a good perceptual quality, but in a low PSNR for their reconstruction. For this reason, we include one plot (DjVu-VO) corresponding to the set of parameters that maximizes the visual quality of the reconstruction, and another one (DjVu-PO) corresponding to the set of parameters that maximizes the PSNR of the reconstruction, disabling the use of binarization of text and graphics.

The objective results are presented in Figures C.8 and C.9. Image Scan0004 (see Figure I.7 of Appendix I) was scanned from page 1363, of the *IEEE Transactions on Image Processing*, volume 10, number 9, September 2001, and image Cerrado (see Figure I.8 of Appendix I) was scanned from a book at 300 dpi. We selected images with different scanning resolutions in order to demonstrate the flexibility of the proposed method.

For text images, H.264/AVC tends to be more efficient than JPEG2000, showing consistent quality gains of about 1 dB for all tested compression ratios. For text images, the original MMP and MMP-II have similar performances, achieving gains of up to 4 dB over H.264/AVC and about 5 dB over JPEG2000 [3, 49]. The use of flexible partitioning (MMP-FP) also improved the performance for text image coding, by up to more than 1

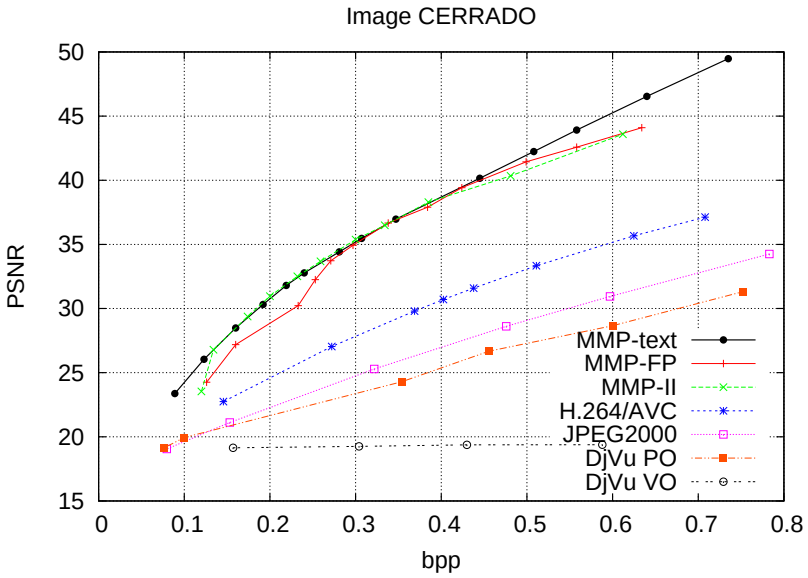


Figure C.9: Experimental results for text and graphics Cerrado (1056×1568).

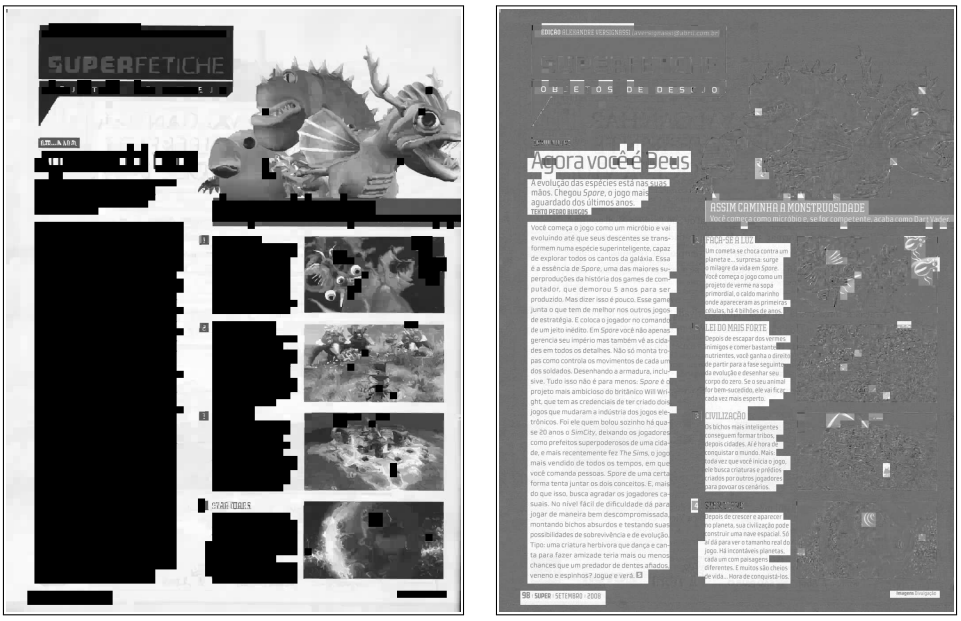
dB. In addition, the use of MMP-text improved the performance of the MMP-based encoders by 1 dB, establishing the overall advantage of these methods over JPEG2000 and the H.264/AVC *high profile* at about 7 dB and 6 dB, respectively. It is important to notice that the better performance over the MMP-FP algorithm was obtained with a lower computational complexity.

The textual regions are encoded with MMP-text immediately after encoding the binary mask. Since no predictive coding is used, each of these blocks can be encoded and decoded individually, with no need for reference neighboring blocks. A raster scan order is used to code the blocks sequentially, skipping all the blocks identified as pictorial regions. In the decoding process, the reconstructed segmentation mask indicates which blocks should be skipped to perform the reconstruction.

C.2.5 MMP for smooth images: MMP-FP

The main developments on MMP image compression methods were focused on their performance for natural images. Several approaches were proposed to optimize the MMP algorithm for smooth image compression, as described in Appendix B. Thus, MMP-FP was adopted for the pictorial blocks compression.

After encoding all non-smooth blocks with MMP-text, the algorithm encodes the smooth image blocks using MMP-FP. The text blocks previously encoded may therefore be used as references for the prediction step of MMP-FP. Although it may seem inappropriate to use the neighboring text blocks as prediction references for pictorial blocks, it makes sense because, as the segmentation is block based, it is common that the blocks on the frontier between smooth and text regions contain pixels of both types. In this case, two situations are possible:



(a) (b)

Figure C.10: Prediction generated while encoding image Spore at 0.38bpp

- The neighboring text block used for prediction already contains smooth image pixels in the border, that can accurately predict the next smooth block;
- The smooth block being coded still contains text pixels, that can be predicted by the neighbor text pixels from the text block in the frontier.

This approach contributes to increase the algorithm's coding efficiency, specially for complex segmentation masks, maximizing the prediction accuracy in the frontier regions. Furthermore, this reduces the algorithm's sensitivity to block's misclassifications.

Figure C.10a illustrates the prediction obtained while coding image Spore at 0.38bpp, and Figure C.10b shows its respective prediction error. In this case, the prediction is displayed with an offset of 128, in order to allow the visualization of negative values. Note that in textual regions, which are coded with MMP-text that does not use prediction, the signal encoded is the original signal instead of the prediction error. The low energy in the prediction error in smooth regions demonstrates that the prediction was accurate, even in the frontier regions using the previously encoded text blocks.

It is important to note that this segmentation-based approach generates two dictionaries, one for smooth blocks (MMP-FP) and one for non-smooth blocks (MMP-text). This has advantages regarding the efficient use of the dictionaries. As a dictionary is updated with previously encoded patterns, it is expected that the code-vectors created while encoding text regions will, in general, be of little use while encoding the prediction residues from smooth regions. Likewise, smooth patterns resulting from coding smooth areas are unlikely to be a good match for text blocks. Furthermore, the dynamic range of the code-vectors from smooth regions, which are encoded using a predictive scheme, is twice of

the dynamic range from codewords from the textual regions. Thus, by using different MMP-based encoders, each with its own dictionary, two dictionaries with two different groups of code-vectors are created. One with blocks originated by the concatenation of non-smooth blocks (that tend to have a considerable high-pass component), and blocks originated by concatenations of smooth blocks, generally of a low-pass nature. This way, the dictionary blocks created while compressing one layer do not contribute to increase the indexes' entropy of the dictionary from the other layer, as would be the case for the single encoder approach. This approach has the additional advantages on reducing the computational complexity, as less blocks need to be tested while performing the matches.

Since MMP is a block based encoder, it has the tendency to suffer from blocking artifacts on natural regions, specially at low bit rates. These artifacts are specially annoying in smooth regions, due to their high spatial correlation. A new post-filtering technique, presented on Appendix F, was adopted to reduce these blocking artifacts, introduced on the reconstructed images.

However, we have noticed that the filter application is usually not beneficial when applied to the compound image's textual components. Instead of estimating the filter's parameters values, the adopted approach exhaustively optimizes these parameters, in order to select the combination which maximizes the PSNR of the reconstruction.

If the same filter's parameters are used for the entire image, we had observed that aggressive filters introduced some degradation on sharp edges. This forced us to reduce its smoothing effect, either by modifying its shape or reducing its support length. This way, an insufficient deblocking effect is achieved for the pictorial regions. To obtain the required deblocking effect on pictorial regions, some degradation has to be introduced in the textual regions. Furthermore, the application of a deblocking filter on text and graphics regions also imposes an additional computational complexity, which is not compensated by a corresponding increase on the subjective and objective quality.

In the present case, the segmentation mask allows to simply disable the filter for textual regions, in order to maximize the deblocking effect on pictorial regions, without any type of degradation of text and graphics details. This resulted in a superior subjective and objective quality of the reconstructed documents.

C.2.6 Perceptual quality equalization

The proposed compression method was first evaluated using the same value for the lagrangian operator λ for both the textual and the pictorial image components. This corresponds to the optimal bit allocation between the two used encoders, maximizing the rate-distortion performance. However, examination of the images showed that the subjective quality of the textual component was considerably higher than that of the pictorial component, for a given value of λ.

This phenomenon is explained by the fact that, for the same rate, the squared error in a text region has the tendency to be higher than for smooth regions, due to the lower spatial correlation between the pixels. Then, when the overall rate-distortion performance is optimized, bits tend to be transferred from the smooth to the text regions. As a consequence, a large amount of blocking artifacts is introduced in pictorial regions.

Considering the expression of the Lagrangian cost, given by:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}), \tag{C.2}$$

a straightforward solution for the problem is to apply different values of λ in text and smooth regions, respectively, in order to equalize the perceptual quality for the two components.

Let us define a new parameter α, that relates the values of λ for pictorial and textual regions (λ_{nat} and λ_{text} respectively):

$$\alpha = \frac{\lambda_{nat}}{\lambda_{text}}. \tag{C.3}$$

By adjusting this parameter α, it becomes possible to allocate more or less bitrate for each component.

More specifically, imposing that $\alpha < 1$ means that more rate will be allocated for the pictorial component, then the PSNR will increase in the pictorial and decrease for the textual component, for the same global rate.

In Figure C.11, we present, respectively, the PSNR for the textual and the pictorial regions for image SCAN0002 *vs.* the global bitrate, for 4 different values of α, namely 1, 0.9, 0.8 and 0.7. The figure clearly shows the increase in PSNR for the smooth component and the decrease for text and graphics regions, when α decreases. The overall PSNR decreases with α, as expected, since the use of the same value for λ in both the text and graphics and smooth component (which corresponds to $\alpha = 1$), maximizes the objective quality of the reconstruction [44].

Defining an optimal value for α can however be a challenging task, as it may depend of the particular characteristics of each image. Thus, we defined the value of α using informal subjective evaluations, but in the future, some researches can be conducted in order to adjust this value to some particular applications.

The value $\alpha = 0.8$ was established as a trade off between the subjective and objective quality for the reconstruction. For that purpose, the lowest bitrate that results in a readable reconstruction was used. The value of $\alpha = 0.8$ was considered to deliver an acceptable quality for the smooth component, without compromising the readability of the document.

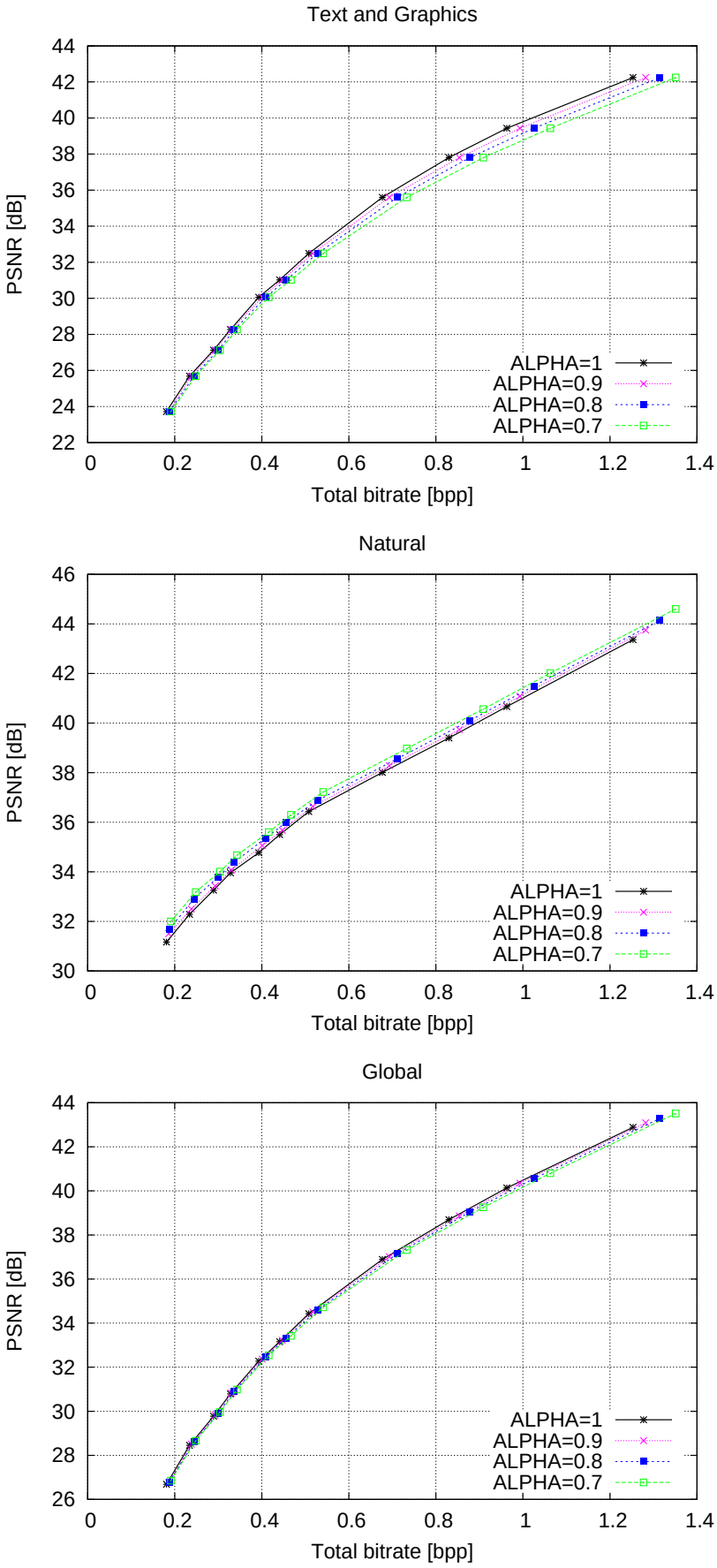
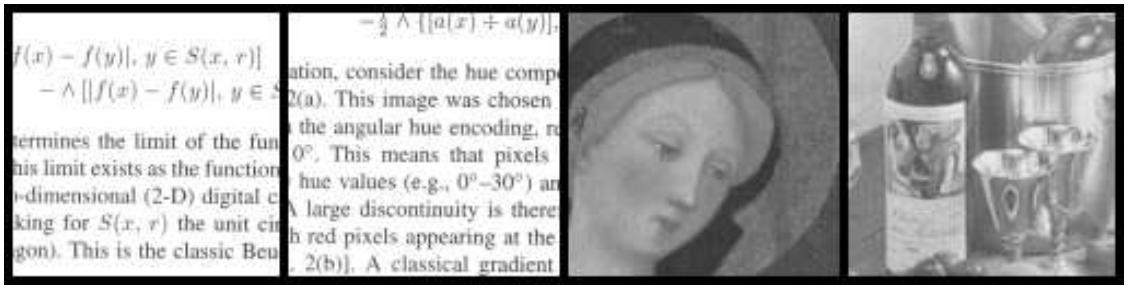
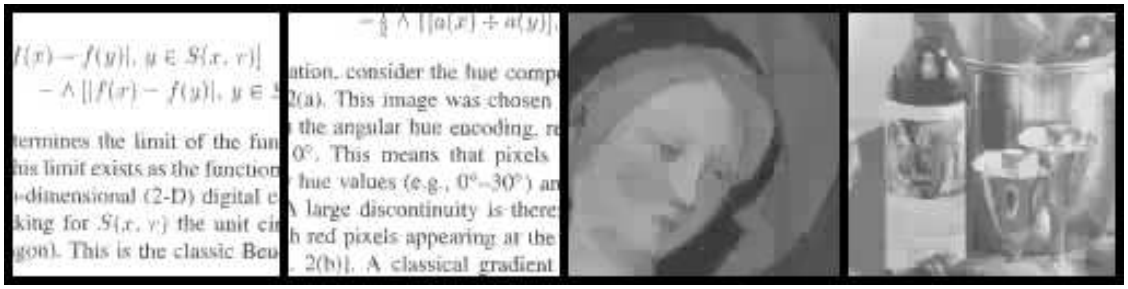


Figure C.11: PSNR variation for image Scan0002, for different values of α a) text and graphics component only; b) natural component only; c) entire image.

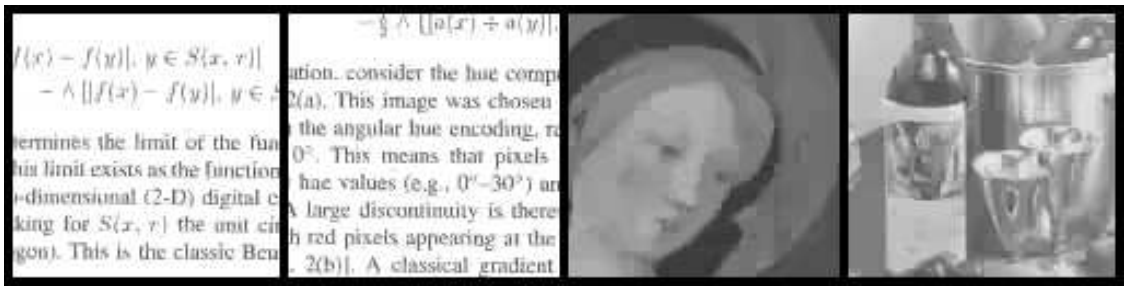
Figure C.12 shows some details of image Scan0002 coded at 0.30bpp using $\alpha = 1$ and $\alpha = 0.8$, respectively. It can be seen that the overall perceptual quality of the second reconstruction is higher than that of the previous one. The blocking effect on the smooth regions decreased, and the detail is considerably higher in these regions than in the first case. For the text regions, it can be seen that the perceptual quality was not considerably degraded, as only the edge sharpness is slightly affected, but the text readability is maintained.



(a) Original at 8bpp



(b) MMP-compound $\alpha = 1$ at 0.30bpp (30.25dB)



(c) MMP-compound $\alpha = 0.8$ at 0.30bpp (29.98dB)

Figure C.12: Details of compound image Scan0002 a) Original; b) $\alpha = 1$; c) $\alpha = 0.8$.

In spite of an average PSNR loss of around 0.2 dB, it can be seen that the subjective quality was improved. Furthermore, the rate-distortion performance of MMP-compound remained superior or equal to the one of the other algorithms, including the original MMP-based still image encoding algorithm, for compound scanned images.

This adjustment in our previous algorithm changed the target from the PSNR maximization to a conjugate optimization of both PSNR and subjective quality for the reconstruction. The OCR performance of the algorithm was not evaluated at this point, but it would be a good subject for future research.

C.3 Experimental results

In this section we present a performance comparison between the proposed algorithm and several state-of-the-art compound document encoders, for a set of scanned compound test images. Several scanned compound documents, presenting different characteristics were used to evaluate the proposed method. These images were originated by compound document scanning in grayscale at 8bpp, containing both textual and pictorial contents.

The performance of the proposed method was evaluated against two state-of-the-art transform-based encoders: JPEG2000 [50] and H.264/AVC High Profile Intra-frame still image encoder [45, 54]. The JPEG2000 has been chosen as a state-of-the-art reference for DWT-based image encoders. H.264/AVC has been chosen for several reasons: its excellent performance for image coding when using intra-coding tools [54], and its prediction modes, which have inspired the ones used by MMP-FP. Furthermore, the H.264/AVC compression standard was used to develop several document compression layouts, such as the ones presented in [75, 76, 121].

The proposed compression scheme results are also compared with those from Lizardtech's Document Express with DjVu - Enterprise Edition [77], one of the most successful examples of MRC-based algorithms. Tests using other MRC based commercial applications were also performed for comparison. However the performance of these MRC based applications revealed to be very similar to the one of Lizardtech's Document Express with DjVu, as their performance mostly rely on the result of the MRC segmentation step.

Note that for all the presented results, the in-loop deblocking filter of the H.264/AVC [78] encoder has been activated, and we enabled two features from DjVu which enhance the subjective quality of the coded document: subsample refinement and background floss.

C.3.1 Objective performance evaluation

A comparison with the rate-distortion performance of JPEG2000 [50] and H.264/AVC [45], as well as with the one of DjVu [45] (a state-of-the-art MRC-based algorithm for document compression), is presented in this section.

It is important to note that DjVu is usually not optimized to deliver the best rate-distortion performance, but to preserve the readability of the documents. The binarization used in the text regions usually preserves the subjective quality of the document, but tends to yield a very low PSNR for the reconstruction. For this reason, we adopted here the same approach used for Figure C.9 to Figure C.8, presenting the results for two sets of parameters. DjVu-PO corresponds to the parameters setup which maximize the objective rate-distortion performance, whereas DjVu-VO represent the results which optimize the subjective quality of the reconstructed documents (tuned for each image).

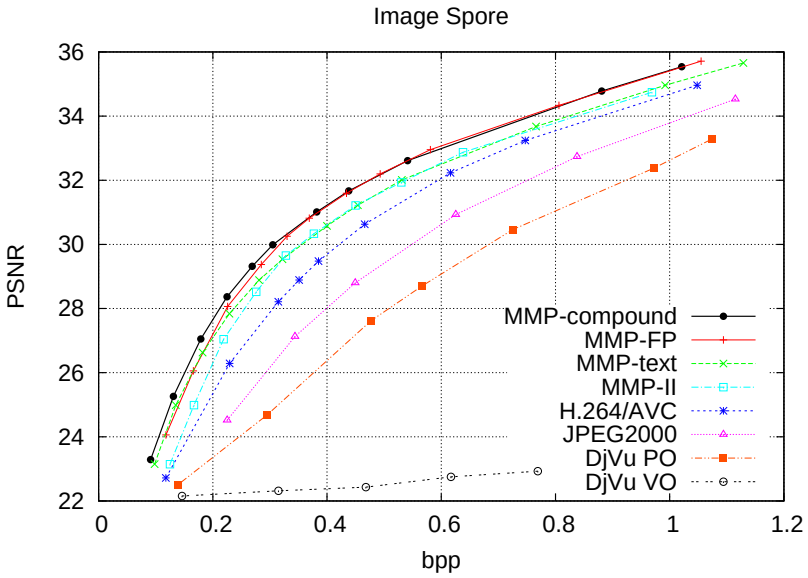


Figure C.13: Experimental results for compound image Spore (1024×1360).

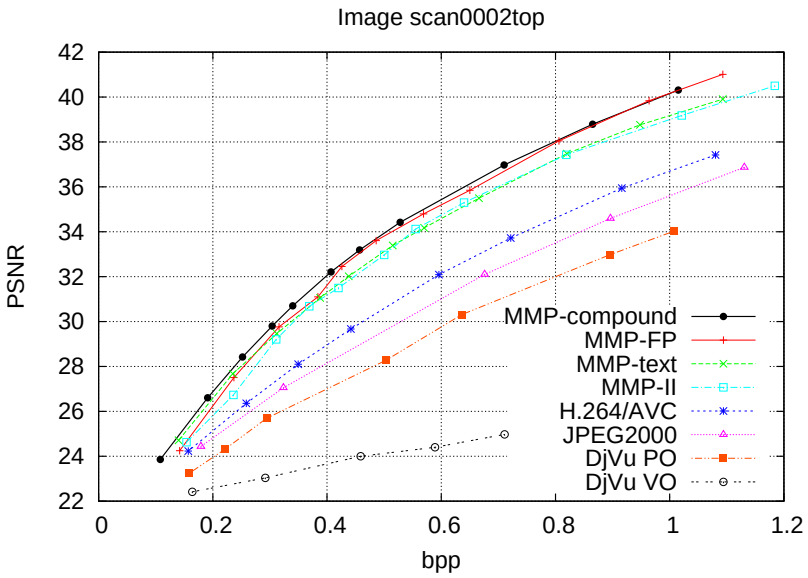


Figure C.14: Experimental results for compound image Scan0002 (512×512).

Figures C.13 and C.14 show the rate-distortion results for the tested methods, for the used test images. Table C.1 highlights the final PSNR for a set of compression ratios, for image Scan0002. From these results, one may observe the consistent gains achieved by the use of the hybrid MMP-based approach (MMP-compound) over each tested encoder.

In order to complement the rate-distortion performance evaluation of the proposed method, we also compared MMP-compound with H.264/AVC-based [45] algorithms, specifically optimized for scanned compound document encoding, such as H.264/ADC [121]. Despite the significant rate-distortion performance gains achieved by H.264/ADC over H.264/AVC, which in some cases are up to 4 dB, MMP-compound is still able to outperform H.264/ADC (by up to 2 dB, in some cases). Furthermore,

Table C.1: PSNR results from the image Scan0002 [dB]

Rate [bpp]	**0.20**	**0.40**	**0.60**	**0.80**	**1.00**
DjVu PO	23.95	27.00	29.76	32.00	33.96
DjVu VO	22.21	23.77	24.20	25.03	26.43
H.264/AVC	25.13	28.96	32.15	34.61	36.70
JPEG2000	24.82	28.15	31.02	33.51	35.61
MMP-II	25.81	31.17	34.75	37.20	39.00
MMP-FP	26.24	31.61	35.20	37.96	40.16
MMP-text	26.59	31.28	34.58	37.23	39.17
MMP-compound	**27.16**	**32.17**	**35.50**	**38.05**	**40.19**

we also compared the proposed method with an evolution of H.264/ADC, referred as HEDC [122], which is based on the upcoming HEVC [16] standard proposal. This method achieved considerable gains over H.264/ADC, but its performance is consistently bellow that of MMP-compound, by up to 1dB.

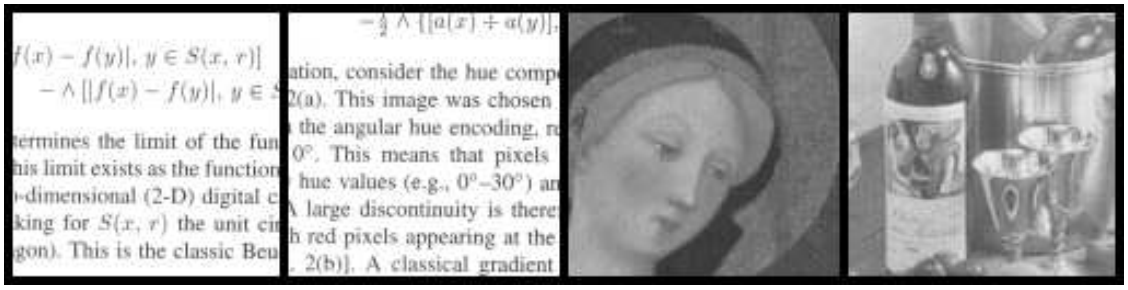
For example, for the case of image Spore encoded at 1 bpp, HEDC achieves a PSNR of 34.6 dB, while MMP-compound achieves a PSNR of 35.7 dB, which results in a performance advantage of 1.1 dB.

Several tests were also performed in order to evaluate the method's robustness against block misclassifications. These tests demonstrated a high performance, even for systematic classification errors. When a misclassification occurs, or when a block has mixed features (text and image), MMP is able to efficiently find a convenient match. This is achieved at the cost of an additional rate, spent to create these patterns through successive segmentations on the MMP coding step. Experimental tests show that the random switch of blocks from one layer to another only brings modest losses in the final performance of MMP-compound. In fact, the PSNR curve only suffers a noticeable degradation if a massive misclassification occurs. In the extreme case were all the blocks are misclassified (obviously an unrealistic scenario), the observed losses were not greater than 1.1 dB. This still places MMP-compound's results well above those of the state-of-the-art algorithms.

C.3.2 Observation of subjective quality

In order to assess the subjective image quality provided by the tested methods, Figure C.15 presents a detail from image SCAN0002 compressed using MMP-compound, JPEG2000, H.264/AVC and DjVu-VO at 0.3 bpp.

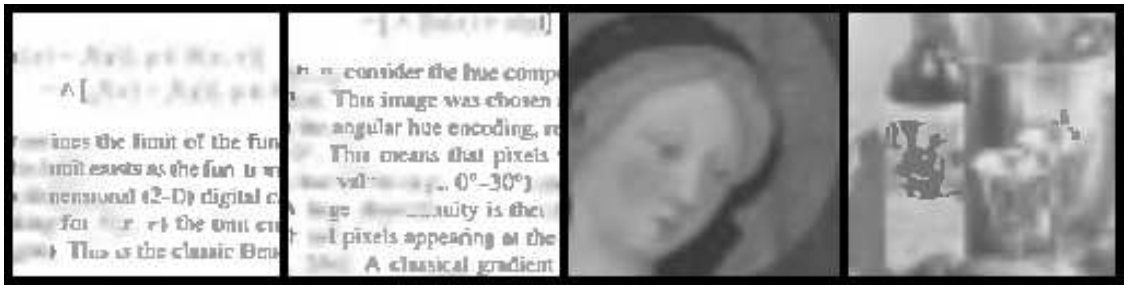
When we analyze the textual regions, the disturbing ringing and blurring artifacts become obvious for the JPEG2000 reconstructed images. With H.264/AVC, some artifacts also appear in these areas, but they are not so disturbing as the ones introduced by JPEG2000. For both algorithms, at such a high compression ratio, the legibility of the



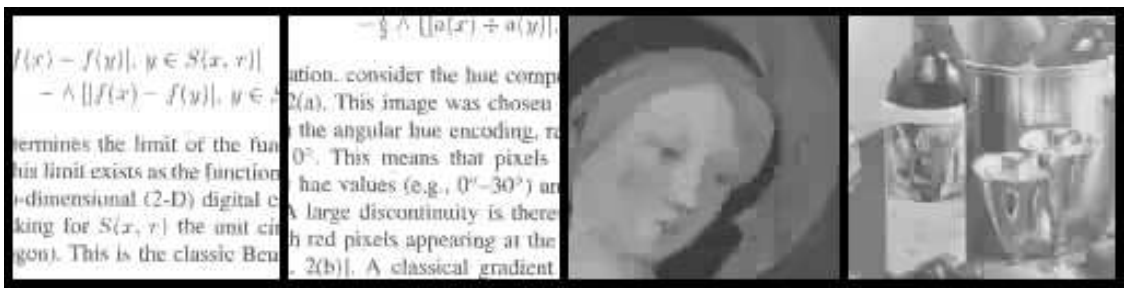
(a) Original at 8bpp

(b) JPEG2000 at 0.30bpp (24.44dB)

(c) H.264/AVC at 0.30bpp (27.11dB)



(d) DjVu at 0.31bpp (23.07dB)



(e) MMP-compound at 0.30bpp (29.98dB)

Figure C.15: Details of compound image Scan0002 a) Original; b) JPEG2000; c) H.264/AVC; d) DjVu; e) MMP-compound.

document is compromised. In the reconstruction obtained using DjVu, the sharp edges of the characters coded in the foreground layer contribute for a good perceptual quality, despite of their irregular shape. However, the wrong pixel classifications, which are common when one uses DjVu on scanned documents, result in some illegible text regions, such as the equation on the left top of Figure C.15d. For textual regions, the subjective quality advantage of MMP-compound over transform-based algorithms can be clearly observed. In addition, unlike DjVu, MMP-compound does not present legibility issues even for a high compression ratio. It is important to emphasize that the image encoded with DjVu presented in Figure C.15d was obtained using parameters that have been carefully adjusted for the best subjective quality. The images corresponding to the results in Table C.1, that have larger PSNR, present a worse readability than the one from Figure C.15d).

For the pictorial image regions, one may observe that JPEG2000 introduces some blurring and ringing effects. These artifacts are also noticeable in the reconstruction obtained using DjVu. In addition to these artifacts, DjVu reconstruction suffers from the effects of misclassified pixels (for example in the bottle in the bottom right detail of Figure C.15d). On the other hand, neither MMP-compound nor H.264/AVC present such artifacts. It is important to note that, although both H.264/AVC and MMP-compound are block based encoders, they do not suffer from pronounced blocking artifacts at this bitrate, because both of them use post-filtering techniques to alleviate blockiness.

The overall subjective advantage of MMP-compound, when compared to the other tested algorithms, is clear in this example. Unlike traditional methods, MMP preserves the readability of the text even at low rates, together with a subjective quality advantage in the smooth image regions.

C.4 Conclusions

In this appendix, a new compound document encoder based on multiscale recurrent pattern matching was proposed and described. The new algorithm uses a block classification approach, decomposing the image into smooth and non-smooth regions. Different MMP-based encoders (MMP-FP and MMP-text) were specifically optimized for each image type. MMP-FP is used as the state-of-the-art multiscale recurrent pattern matching algorithm to compress smooth images, outperforming state-of-the-art DWT and DCT-based encoders. The optimization of MMP for text image compression improved its rate-distortion performance for such images, yielding gains over DWT and DCT-based encoders of up to 7 dB.

The adaptive use of MMP-FP and MMP-text, leading to MMP-compound, provided a compound document encoder with both very good rate-distortion and subjective performances. One of the main factors contributing to this is that the universality of MMP-based methods results in high resilience to wrong block classifications. This is in contrast with

the results obtained with traditional document encoding algorithms, like DjVu, whose performance is very sensitive to such pixel classification errors.

The experimental results presented in this appendix demonstrate the high efficiency of MMP-based scanned compound document encoding.

Appendix D

Efficient video encoding using MMP

D.1 Introduction

The hybrid model has been confirmed over the past decades as the most successful architecture for video compression algorithms. Many successful video coding standards relied on this general architecture, including the H.264/AVC [45] video coding standard.

A motion-compensated prediction or Intra-frame prediction are used to respectively reduce the temporal and spatial redundancies of the signal, and the resulting residue is compressed using the traditional transform-quantization-entropy coding paradigm, that is able to efficiently exploit the remaining data's statistical correlation. The significant advantage in encoding performance of H.264/AVC [45] over its predecessors is not the result of any change in the coding paradigm, but results mainly from the exploitation of a richer set of tools for each of the encoders' modules, resulting in a more complex, but highly efficient method [51].

The high performance of the hybrid model has conditioned the use of alternative approaches to video compression. As for the case of still image compression, there have been rare proposals to adapt pattern matching-based algorithms to video coding applications. Some exceptions were presented in [36–39], but none of these methods has been able to achieve a performance near to that of current state-of-the-art hybrid methods.

As discussed on Appendix B, MMP was able to achieve high coding efficiency for a wide range of signals. Given these past experiences, one of the objectives of this thesis was to develop a new video compression method that was fully supported in the pattern matching paradigm, while achieving a coding performance competitive with that of H.264/AVC [45]. For that purpose, the transform-quantisation-entropy coding step used on H.264/AVC [45] was totally replaced by the Multidimensional Multiscale Parser (MMP) [3] algorithm. Additionally, some new improvements were introduced, targeting the improvement of the compression efficiency of video signals.

The use of MMP for motion compensated residue coding was already investigated in

the past, as described in [7, 14, 79]. This method used Multiscale Recurrent Patterns to compress the motion predicted residue, maintaining the original H.264/AVC approach to encode Intra frames. This architecture was mainly motivated by the fact that MMP was, at that time, considerably less efficient than H.264/AVC while compressing reference frames. The use of MMP to compress all the predicted residues resulted in an overall performance lower than that of H.264/AVC, as the gains achieved while compressing time estimated frames were insufficient to compensate the lower performance on Intra-frames, that are responsible for most of the required bitrate.

In this chapter, we describe the general architecture of the proposed video encoder, as well as some specific MMP optimizations, which allowed to better exploit some particular features of video signals. The performance of the new encoder is evaluated against the current state-of-the-art video compression standard: the H.264/AVC high profile video encoder [45].

In Section D.2, an overview of video encoding is presented, with emphasis on the features that have impact on the design of the proposed algorithm. The main features of the proposed algorithm are described in Section D.3. Experimental results are presented in Section D.4, and Section D.5 summarizes the conclusions of this work.

D.2 Video coding overview

A video sequence is a temporal succession of image frames, typically with high levels of spatial and temporal redundancies. The success in exploiting these redundancies is the key feature for a video encoder performance.

A common strategy, used by many state-of-the-art video standards, as H.264/AVC [45], consists in applying spatial or temporal prediction to each slice. The resulting residue is then compressed using a transform-quantization-entropy coding strategy.

Intra-predicted (I) slices are coded using only spatial prediction, based on previously coded regions in the same picture. These slices can then be used to generate temporal prediction to subsequent slices (inter-prediction), through motion estimation (ME). Moving objects appear in several frames with different spatial positions inside the scene. For this reason, an effective way to encode this information is to divide the image into blocks and transmit a motion vector (MV) for each block. Each MV represents the position of a similar block in a previously encoded slice. ME is the process of finding the best pair of reference/MV for each of the encoded blocks.

With this approach, the transmission of the luminance values for a displaced block can be replaced by the transmission of a two-dimensional vector, resulting in high compression ratios for those blocks. Additionally, a residue can also be encoded in order to reduce the distortion of the encoded block, compensating luminance variations or changes in the

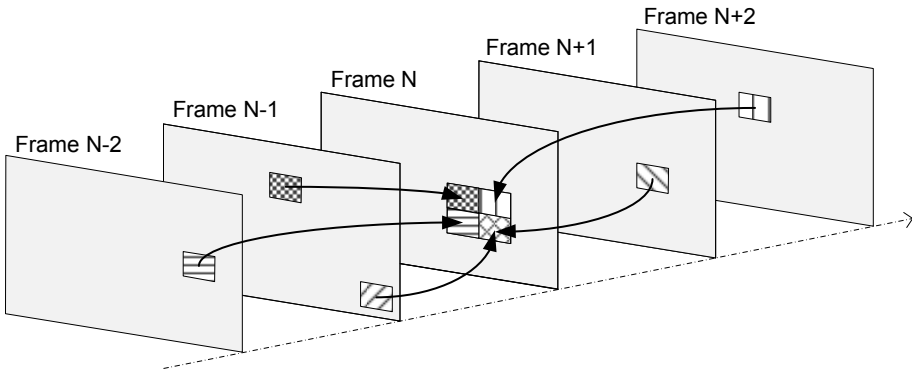


Figure D.1: Bi-predictive motion compensation using multiple reference frames.

shape of the objects.

ME can be performed using either only past slices as reference, or also future slices, in a bi-predictive scheme. In the first case, we refer to the predicted slice as a P slice, while in the second case as a B slice. The use of bi-predictive ME generally allows a better coding efficiency, but also requires added computational complexity, as more references need to be tested and stored.

Figure D.1 illustrates the case where a bi-predictive ME is used to encode slice N. As the order from which the slices are encoded differs from the order they are displayed, both past slices ($N-2$ and $N-1$) and future slices ($N+1$ and $N+2$) must be available when the slice N is encoded, in order to allow their use as references for ME.

Although all previously encoded slices could be used as references for motion estimation, generally only I and P slices are used for this purpose. For this reason, these slices are called key-slices, and encoding them with low distortion is essential for the video encoder's efficiency. The way non-key slices are encoded tends to have only a local impact on the rate-distortion performance, or in other words, the decisions made by the rate-distortion optimization process tend to only affect the rate and the distortion associated to that particular slice. On the other hand, the way key slices are coded has a global impact in the codec's performance, as it will determine the inter-prediction's quality for subsequent slices. Thus, a particular attention is required when encoding key slices, in order not to compromise the overall encoding performance of the video compression algorithm.

An interesting relation can be established between ME and pattern matching algorithms, as in LZ schemes, the references for ME correspond to a previously encoded portion of the message. We can thus compare the MVs in H.264/AVC with the LZ77 [17] pointers. The length of the message used in each approximation is implicitly encoded by the partition size used in the MC process. The LZ search buffer is defined by the reference frames that are used for each slice. Additionally, it presents an interesting additional feature in relation to the basic LZ scheme: the use of B slices allows the use of a combination of blocks from different reference frames. This means that each segment of the message may be encoded as a combination of two previously encoded segments of the message.

We can also regard the patterns stored in the reference frames as an adaptive dictionary. Because the dictionary is composed by previously encoded segments of the video sequence, this may be related either to an LZ78 [18] or a VQ algorithm [28]. Each MV acts as an index that identifies the chosen code-vector. The use of different partition sizes in the MC process can be regarded as the use of several dictionaries, that store blocks with different dimensions.

Furthermore, the dictionary adaptation process consists in the use of a variable set of code-vectors, that are chosen according to a temporal (related to the choice of reference frames) and a neighborhood (represented by the search window) criteria. This increases the dictionary's efficiency, because these codevectors are likely to be similar to the current block. As for the LZ77 analysis [17], the use of B slices can be interpreted as an extension of the dictionary-based coding paradigm, to the case were a weighted combination of two code-vectors is employed.

D.3 Video coding with multiscale recurrent patterns - MMP-Video

The H.264/AVC video coding standard introduced several highly efficient compression tools, combined in a versatile video coding platform. Since our main objective was the development of a fully pattern-matching-based video compression algorithm with state-of-the-art results, the adoption of some H.264/AVC most successful features, such as the optimization loop, the ME algorithms, the entropic compression schemes, just to name a few, where obvious choices. Thus, we based the proposed algorithm (MMP-video) on the H.264/AVC architecture, sharing the same structure of JM reference software [80].

Figure D.2 represents a simplified block diagram of the H.264/AVC encoder. In MMP-video, the blocks corresponding to the transform and quantization steps are substituted by the MMP algorithm, resulting in modified block diagram presented on Figure D.3. All the other features remain the same in the proposed video encoder, including the RD optimization scheme.

Although we do not employ transforms, two quantization parameter values are defined in order to control the target compression ratio of the encoder. The values of the QP parameter are set independently for the I/P slices (key slices) and for the B slices (non-key slices) [83], and have a direct correspondence with the value of the lagrangian multiplier λ [81], used to perform the RD optimization of the built-in MMP encoder.

The encoding of Intra and Inter macroblocks is explained in detail in the following sections.

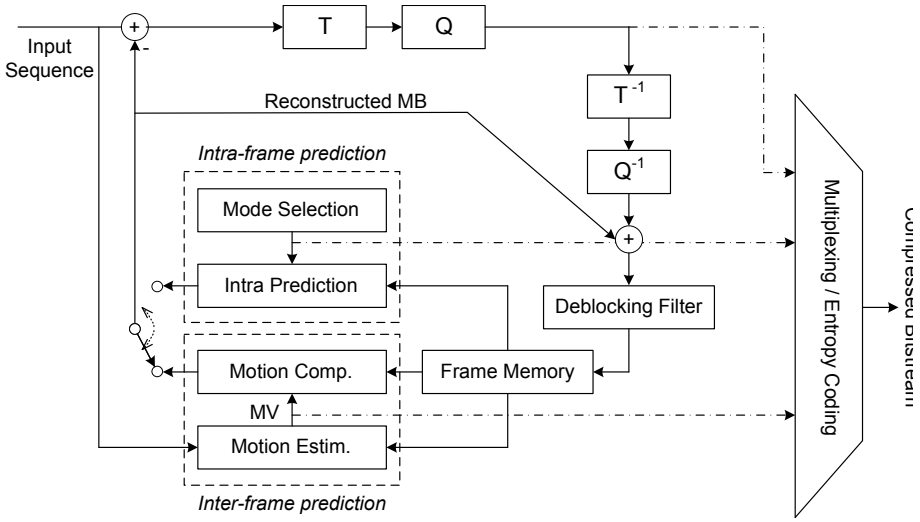


Figure D.2: Basic architecture of the H.264/AVC encoder.

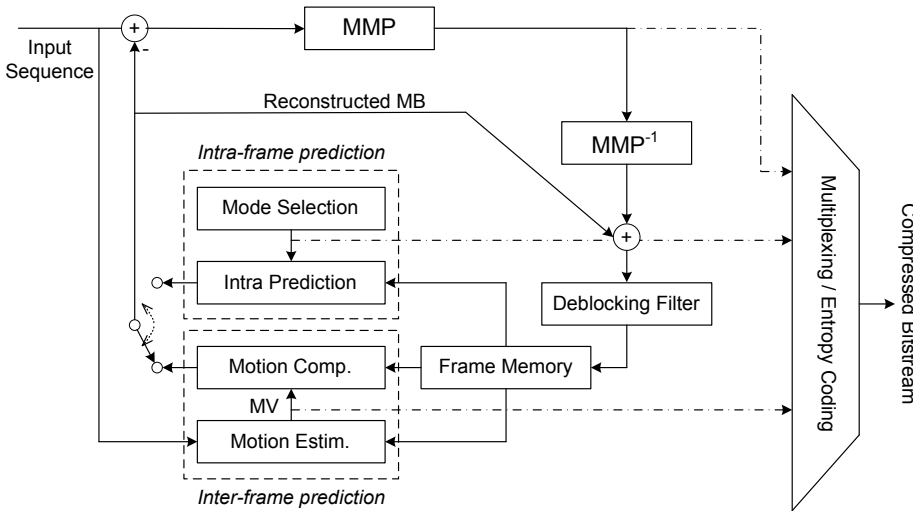


Figure D.3: Basic architecture of the MMP-Video encoder.

D.3.1 Intra macroblock coding

Despite being encoded as still images, without any reference to past or future slices, Intra MBs are determinant for ME based encoders' performance. Intra slices are used both as direct and indirect references for ME, since motion estimated slices can also be used as reference to other slices. Consequently, the distortion introduced in the compressed I slice will potentially propagate through the video sequence, limiting the ME's effectiveness, and compromising the encoder's overall performance.

MMP-based compression of intra MBs is a straightforward adaptation of MMP-FP, that is described in Appendix B. This way, MMP-video uses a hierarchical Intra prediction scheme similar to the one used in H.264/AVC [45], but with the introduction of several other features, like the use of new prediction schemes and prediction block-sizes.

As for MMP-FP, the DC prediction mode was replaced by the most frequent value

mode (MFV) and the LSP [46] was added as an extra prediction mode. Additionally, a prediction mode based on inter component correlation was added for chroma MB prediction [123]. This has been done because, despite the significant inter component decorrelation achieved with the use of the YUV color space, some residual correlation can still be exploited between the luma and the chroma components [124]. For each chroma component, a linear model is used in order to generate a prediction based on the previously encoded components. As Y, U e V are encoded sequentially, it is possible to use Y to linearly predict U, and both Y and U to predict V, resulting on the following linear models:

$$\hat{U}(x,y) = \alpha Y'(x,y) + \beta, \tag{D.1}$$

$$\hat{V}(x,y) = \gamma Y'(x,y) + \delta U'(x,y) + \epsilon, \tag{D.2}$$

where $\hat{U}(x,y)$ and $\hat{V}(x,y)$ represent the U and V generated predictions, respectively. $Y'(x,y)$ is the subsampled reconstructed Y block and $U'(x,y)$ the reconstructed U block.

The parameters that define the linear model are considered to be independent of the pixel coordinates (x,y) within a block, since it is reasonable to suppose that the blocks of an image constitute a stationary process. In this case, parameters are estimated for the whole block, using previously reconstructed neighboring samples of each available component.

Just like in the LSP prediction mode, the linear model parameters are estimated based on a least squares method, by minimizing the square of the prediction error:

$$\xi(x,y) = U(x,y) - \hat{U}(x,y). \tag{D.3}$$

In this case, α is one-dimensional and the equation can be written as:

$$\frac{\partial E[\xi^2]}{\partial \alpha} = 0 \quad \Rightarrow \quad \alpha = \frac{R_{Y',U'}}{R_{Y',Y'}}, \tag{D.4}$$

where $R_{A,B}$ means the cross-covariance between components A and B. Using a similar approach, β can be obtained by the equation:

$$\frac{\partial E[\xi^2]}{\partial \beta} = 0 \quad \Rightarrow \quad \beta = \bar{U}' - \alpha \bar{Y}', \tag{D.5}$$

with $\bar{U}$ and $\bar{Y}$ denoting the mean of the chrominance and luminance neighbor samples, respectively.

The linear model parameters used to predict the V components are obtained with an analogous procedure, resulting on the following equations:

$$\frac{\partial E[\xi^2]}{\partial \gamma} = 0 \quad \Rightarrow \quad \gamma = \frac{R_{U,Y'} - \delta R_{Y',V'}}{R_{Y',Y'}}; \tag{D.6}$$

$$\frac{\partial E[\xi^2]}{\partial \delta} = 0 \quad \Rightarrow \quad \delta = \frac{R_{U,V'} - \gamma R_{Y',V'}}{R_{V',V'}}; \tag{D.7}$$

$$\frac{\partial E[\xi^2]}{\partial \epsilon} = 0 \quad \Rightarrow \quad \epsilon = \bar{U} - \gamma \bar{Y}' - \delta \bar{V}'. \tag{D.8}$$

Finally, replacing equation (D.7) in (D.6), γ can be calculated as:

$$\gamma = \frac{R_{V',V'} R_{U,Y'} - R_{U,V'} R_{Y',V'}}{R_{Y',Y'} R_{V',V'} - R_{Y',V'} R_{Y',V'}}. \tag{D.9}$$

In Inter slice compression, when the motion estimation (ME) fails in some blocks due to occlusions or scene changes, this is solved by encoding MBs from Inter slices as Intra MBs. This option is tested in the R-D optimization loop, and selected when its cost is smaller than that of ME, as in H.264/AVC.

In order to reduce the computational complexity of the algorithm, the number of prediction modes used to encode Intra MBs on P and B slices was limited to only four modes (MFV, Vertical, Horizontal and LSP). This allowed us to maintain the algorithm's R-D performance, while considerably reducing the computational complexity.

D.3.2 Inter macroblock coding

MMP-video encodes Inter MB performing a motion-compensation, with the resulting two dimensional residue being encoded using MMP. As in H.264/AVC, MMP-Video uses adaptive block sizes (ABS) to perform motion estimation/compensation of the inter-predicted MBs. This approach represents a significant evolution relatively to previous video compression standards, where only fixed size MC was performed. H.264/AVC allows for seven different segmentation modes for the motion compensated blocks, organised into two hierarchical levels. In the first level, MBs can be partitioned into 16×16, 16×8, 8×16 or 8×8 luma blocks. In the second level, 8×8 luma blocks can be further partitioned into 8×4, 4×8 or 4×4 sub-blocks (see Figure D.4).

Thus, the use of ABS motion-compensation means that each MC-prediction error macroblock (with 16×16 luma samples) can be the result of the concatenation of several smaller segments, depending on the used partition sizes. For each resulting partition, an independent translational motion vector (MV) will be associated, which can correspond to a different reference frame. Consequently, each inter MB will be encoded using a number of MV ranging from 1 (if a 16×16 partition is used), to 16 (if the MB is decomposed only on 4×4 partitions).

In order to optimise the encoding process, the encoder tests exhaustively several encoding options. Because of the high complexity of MMP, the computation of the R-D cost function in MMP-Video is performed using the same metrics as in the H.264/AVC

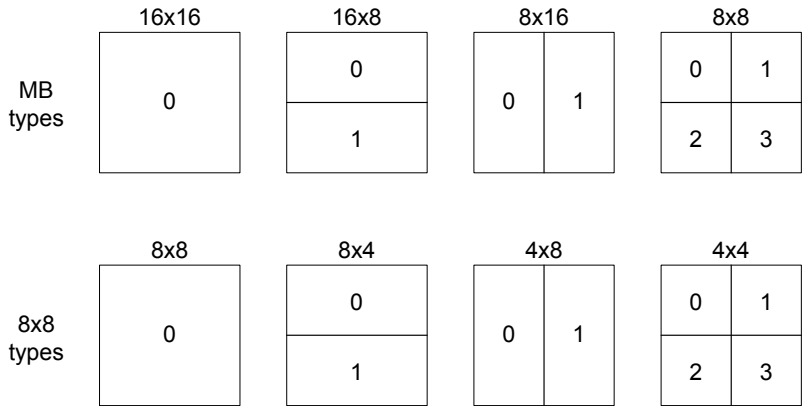


Figure D.4: Adaptive block sizes used for partitioning each MB for motion compensation.

encoder. This means that the distortion is estimated based on the transform coding of the residues, either by using the sum of absolute differences (SAD) or the sum of absolute transformed differences (SATD). Unlike the case of H.264/AVC's, the residue block with minimal SAD or SATD is not necessarily the block that would be more efficiently encoded by MMP-Video. This means that the MC parameter search is sub-optimal from the MMP coding point-of-view. Nevertheless, our simulations have demonstrated that the use of these error measures still allows MMP-Video to perform efficiently, with a significant reduction in computational complexity, when compared to a version that would perform ME with the MMP in the loop. The rate-distortion losses observed when this approach is used instead of the exhaustive optimization are less than 0.3 dB, in the worst case, for Inter slices. Furthermore, it is important to notice that the Inter slices only contribute to a limited percentage of the total bitrate resulting from the sequence compression, so this performance loss is attenuated in the overall algorithm's rate-distortion performance.

H.264/AVC encodes this MB residue using transform coding with a block size that depends on the MC partition size. MMP-Video disregards the MC partitioning choice and processes the entire 16×16 residue block. MMP is thus able to segment the 16×16 block in a way that optimizes the RD cost for the MC residue. Experimental tests showed a marginal performance gain for this option, compared to the case where the original MC partitioning was considered, specially for B slices [4].

Apart from the motion compensated residual data, which is compressed using MMP, all the information transmitted by MMP-Video is encoded using the techniques employed by H.264/AVC [45]. This includes all MC information, like the partition modes and the motion vectors for each block, as well as sequence, slice and MB headers. In these cases, the original H.264/AVC options were maintained, namely the use of VLC or CABAC [82], depending on the used encoding profile. Furthermore, the in-loop deblocking process [78] was also used, since experimental tests demonstrated the efficiency of this method for MMP-Video.

D.3.3 Dictionary design for MMP-Video

Several dictionary design techniques have been investigated for MMP-based still grayscale image compression [49]. However, in a colour video coding framework, the MMP dictionary design possibilities increase significantly, as it becomes possible to exploit some additional signal features. Similarly to H.264/AVC, MMP-Video divides the video sequence into I, P and B slices, which are encoded using different tools. Therefore, one may expect the residue signals to vary accordingly. Also, depending on the number of compressed slices, MMP-Video may have a longer period for dictionary adaptation, *i.e.* to "learn" new residue patterns, which is desirable for this type of dictionaries.

Additionally, the decorrelation that results from the use of the YUV colour space will generally lead to an average lower energy for the chroma residues, when compared with the residues resulting from the luma compression. Therefore, the information about the slice type or the YUV colour components can also be used in the MMP-Video dictionary design process, in order to better exploit the statistical distribution presented by the residues generated for each case.

The new design possibilities motivated an extensive research on new dictionary techniques and architectures, specifically optimised for colour video coding. The knowledge gathered from the work on MMP grayscale still image coding dictionary [49] was used, both in order to evaluate the impact of the former techniques and to develop new dictionary architectures, that are better suited for video compression.

In [49], the use of a single dictionary was proposed for grayscale still image coding. Furthermore, context conditioning techniques were applied, using separate contexts to encode indices corresponding to blocks that were originated at different scales. This allowed exploiting the different probabilities of using a given index, according to the original scale where the corresponding block was originally created. Two dictionary design methods were jointly investigated for the video coding dictionary: the use of independent dictionaries for different image components and/or slice types; and the use of context conditioning techniques based on the slice type and/or colour component.

In the former technique, each dictionary only "learns" the specific residue patterns of each type of source data. This result in highly specialized dictionaries, but each MB can only be approximated by blocks of the corresponding dictionary. As a result, each dictionary tends to have a smaller approximation power than a more general (and thus more complete) one, but lower average entropy for its indices. In the latter technique, all blocks are kept in a single dictionary, whose elements are organised into different partitions, which use independent probability contexts for the arithmetic encoding of the indexes. The following criteria were used to define the probability contexts:

- The slice type: each partition contains the vectors that were created for the I, P or B slices;

- The colour component: each partition contains the vectors that were created for the Y, U or V colour components.

A set of experimental tests was performed to evaluate the relative performance of several dictionary configurations. Each tested dictionary configuration varied according to the number of independent dictionaries (for the I, P, B slices and Y, U, V colour components) and the context partitioning criteria.

For that purpose, we have selected a set of eight video sequences, representing a wide range of motion types and levels of spatial detail: Bus, Container, Flower, Foreman, Mobile&Calendar, News, Tempete and Waterfall. These sequences were compressed using the tested dictionary architectures, in order to access which one was able to achieve, on average, the higher performance. The results from these experimental tests have shown that two dictionary configurations achieved the best overall results:

- The use of three *independent* dictionaries, respectively for I, P and B slices, that learn the residue patterns that correspond to every MB of all three colour components of each slice type.

- The use of a single dictionary to approximate all residue blocks, independently of their corresponding component and slice type. This dictionary uses context conditioning, by considering segments that are created according to the original scale of each new block. The use of a single dictionary means that all dictionary blocks are always available, regardless of the slice type and colour component that is being encoded.

A small rate-distortion performance advantage was observed, on average, when separate dictionaries for each slice type were used. This can be explained since, for the cases where ME is able to generate accurate predictions, the motion compensated residue blocks tend to have a lower energy than that of Intra-predicted residues. This reduces the probability of using dictionary blocks created for Intra slice types in Inter slices and vice-versa. Consequently, a marginal improvement in the dictionary's approximation power is, in general, not enough to compensate the increase in the average entropy of the indices, from a rate-distortion point-of-view. This was also the case when we combined the P and B MB's in a single dictionary. In this case, the more accurate prediction achieved by using multiple reference slices also justifies the infrequent codeword sharing between P and B slices. The use of a single dictionary has also an additional disadvantage, associated with the larger computational complexity, as it is necessary to test a larger set of codewords for each block.

The experimental results also showed that the use of a common YUV dictionary is advantageous, when compared with separate dictionaries for each colour component. This happens because the chroma dictionary adaptation process is conditioned by the lower energy (on average) associated with the predicted colour blocks. This limits the number of

MMP segmentations and dictionary updates, resulting in a sparser dictionary. By combining the luma and chroma blocks in a single dictionary, MMP is able to use a richer, more efficient dictionary to encode the chroma components. As a downside, a small efficiency loss may be observed for luma encoding, but the overall results remain advantageous. Note that the use of downscaled chroma MB's does not impose any constraint, since the multiscale pattern matching adjusts the block dimensions before performing the match. In this case, the larger dictionary scales are simply not used while encoding the chroma components.

The dictionary redundancy control scheme and the scale restriction technique, originally proposed in [49] for MMP-based still grayscale image compression, were also optimised for the new dictionary configuration and source characteristics. The maximum dictionary capacity was fixed on 100.000 elements for each dictionary scale, with the older and less used codevector being discarded when the dictionary is full and a new pattern is created. With this approach, the encoder is able to efficiently adapt to the statistical distribution of the residue patterns. The maximum dictionary capacity was defined through experimental tests, as a compromise between coding efficiency and computational complexity. Larger dictionaries tend to be advantageous from a rate-distortion performance point-of-view, but at the cost of a greater computational complexity.

It is important to notice that the dictionary growth process depends not only on the compression ratio but also on the amount of details of the video sequence being encoded, as described in [49]. When the compression ratio is low, the distortion becomes more relevant in the optimization criterium, resulting in average, in more segmentations. As new patterns are originated by concatenation of segmented blocks, more codevectors are inserted in the dictionary. Similarly, highly detailed sequences tend to require more segmentations for the same target distortion, resulting in more dictionary updates. This dictionary adaptation scheme is able to generate a large number of code-vectors, when they are required, but is also able to create a sparse dictionary, when the rate is the major concern in the optimization process.

The dictionary redundancy control scheme and the scale restriction technique, developed for MMP-based image compression [49], were also optimized to be used in MMP-Video. The use of redundancy control introduces consistent quality gains for all sequences. As for the case of still image encoding, the best value for the used distortion threshold, d, depends on the target rate, and is related with the value of λ. An experimental optimisation of the $d(\lambda)$ rule was performed, according to a procedure similar to the one described in [49] for grayscale still image coding. Experimental results shown that the rule proposed in [49] is also appropriate for colour video signals. As in the case of still image compression, the restriction of the scale transforms used in dictionary updating [49] also achieves relevant computational complexity gains, without a noticeable impact on the rate-distortion performance of MMP-Video.

D.3.4 The use of a CBP-like flag

A coded block pattern (CBP) parameter is used on the H.264/AVC standard to signal the existence of non-null encoded residue block transform coefficients for each MB. This allows to save a significative amount of overhead bits, avoiding the transmission of information associated with some null residual coefficients.

This efficient approach to encode null residual blocks can also be exploited by the MMP encoder, that usually requires the transmission of a total of six symbols for these null residual blocks: one flag (the non-segmentation flag) and one index, for each of the three color components.

Despite of the arithmetic encoder's ability to adapt and reduce the entropy of these symbols, the adaptation process can take some time to converge, resulting in some inefficiency relatively to H.264/AVC, specially in sequences with low motion, where the prediction is very effective.

In order to overcome such inefficiency, a binary flag is encoded using an adaptive arithmetic encoder and transmitted for every tree leaf, immediately before the MMP's dictionary index, that encodes the residue patterns. The null residual pattern is represented by the flag zero, which requires no further information. Other patterns are encoded using the flag one followed by the corresponding index. Such approach increases the rate needed to encode non-zero patterns, but decreases significantly the rate required to encode null residue patterns, which are expected to occur very often if the video codec is able to generate a good prediction for the block.

Note that this approach differs from the proposed on [4], where a CBP flag was used to signal the absence of residual data on a MB-by-MB basis. The extension of the CBP flag for each tree leaf allows to achieve a better representation for the cases where only a portion of the MB residue presents a low energy, with the other portion still having a considerable amount of activity. Furthermore, the inclusion of the decision regarding the use of the CBP flag in the RD optimization loop, allows the algorithm to achieve the optimal trade-off between the rate saving and the amount of distortion introduced in the residue block reconstruction.

The effects of the adoption of the CBP-like flag were investigated for each macroblock type, in order to evaluate the algorithm's performance on each case. Experimental tests have demonstrated that this approach is advantageous when applied to Inter MBs only, degrading the overall rate-distortion performance of the algorithm when used also for Intra macroblocks.

The explanation of these results are related to the global impact of a local decision. When a CBP-like flag is available, it becomes much more attractive, from a lagrangian cost point of view, to transmit null residue patterns, due to their very low rate. This results in much more blocks coded as zero residue blocks, contributing to a higher global

distortion and lower bitrate. From a strictly local RD-optimization point of view, this approach tends to have a positive impact on the encoder performance, since the lowest lagrangian cost is always selected. However, it may have a negative impact in the long term performance, for two distinct reasons:

- Codewords that could be useful in the future are not created because of the rate-conservative approach. The low lagrangian cost of the null pattern limits the codebook growth, specially for patterns of low energy. In other words, blocks that could contribute to a long term decrease in the global coding cost are discarded, based on a local decision;

- Despite being determined based on a local optimization, the additional distortion can have a significant impact in the temporal prediction of other blocks. In other words, spending some extra bits to encode the current MB with lower distortion, can, in some cases, be compensated by a better prediction reference for subsequent blocks, increasing the overall rate-distortion performance of the encoder.

The use of the CBP-like flag also acts like a dictionary growth control tool, by restricting the insertion of codewords with a norm close to zero on the dictionary. Despite contributing to slightly reduce the algorithm's computational complexity, this revealed to be not beneficial on a rate-distortion point-of-view, in some cases. Thus, the CBP-like flag was adopted on MMP-video, only to encode the inter-predicted residues.

D.4 Experimental results

In this section, we present a comparison between the experimental results from the proposed method *vs.* the JM17.1 H.264/AVC reference software.

In order to evaluate the comparative performance of the encoders, the test set was composed by several video sequences, of different types. However, a consistent relation has been observed, independently of the input video sequence. Therefore, only the results of four representative CIF sequences, and two 720p sequences (1280×720 pixels) are presented. The CIF sequences are Bus and Foreman, with moderate movement, and Mobile&Calendar and Tempete, that contain strong motion, while the 720p sequences are Old Town Cross, presenting moderate movement, and Mobcal, which contains strong motion.

For the experimental tests, a set of commonly used parameters was adopted, namely a GOP size of 15 frames with an *IBBPBBP* pattern, at a standard frame-rate of 30 fps and 50 fps, for the CIF and for the 720p sequences, respectively. This configuration guarantees that at least two reference Intra frames are transmitted every second, resulting in a low synchronization time for the video sequence.

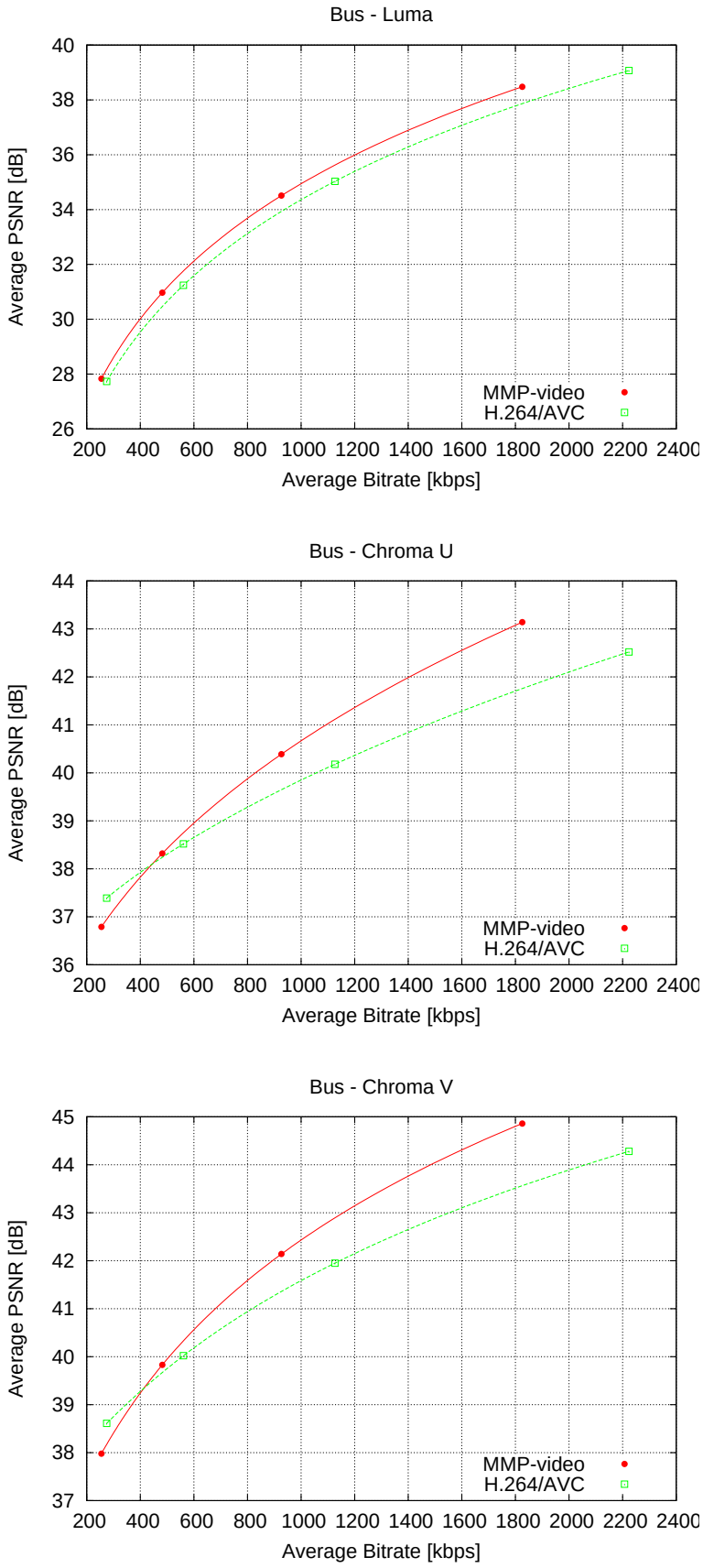


Figure D.5: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Bus sequence (CIF).

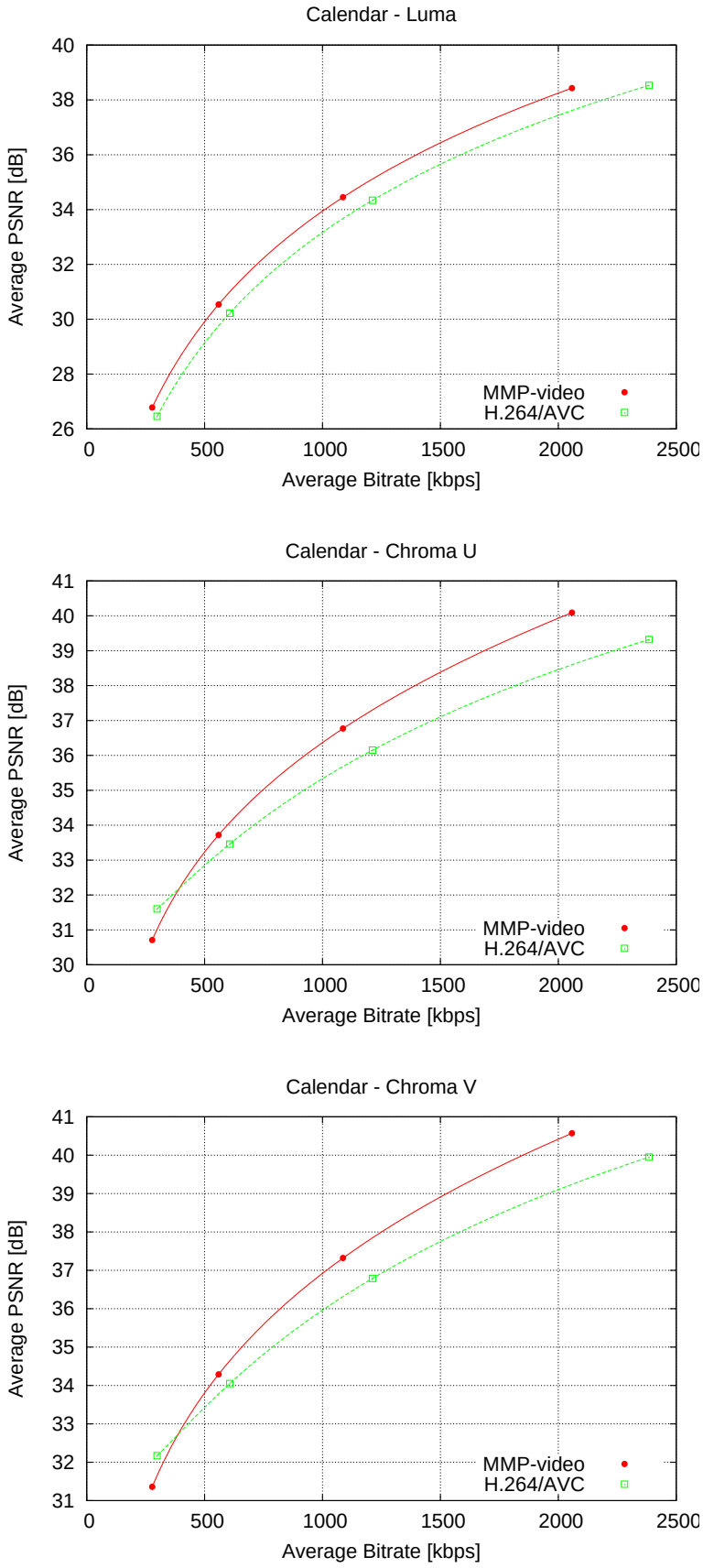


Figure D.6: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Mobile & Calendar sequence (CIF).

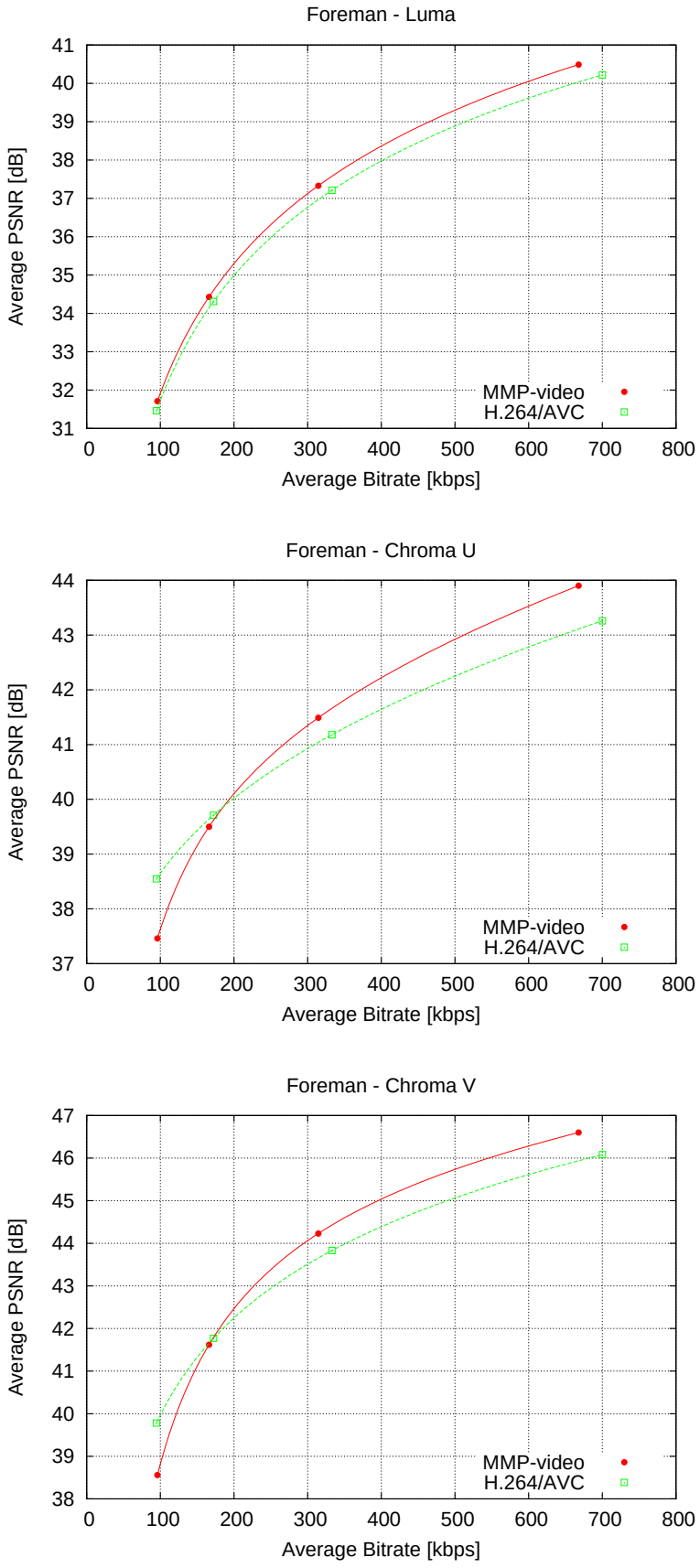


Figure D.7: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Foreman sequence (CIF).

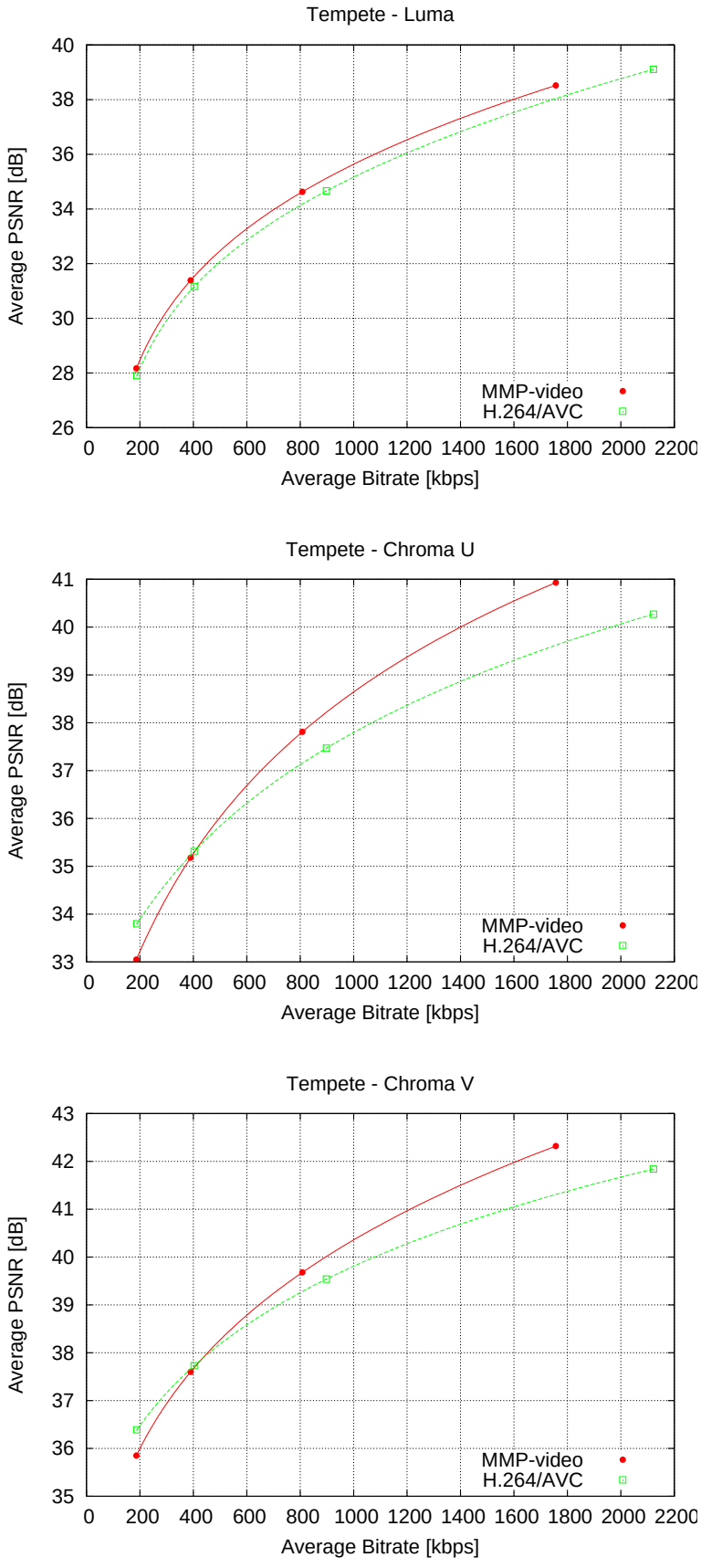


Figure D.8: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Tempete sequence (CIF).

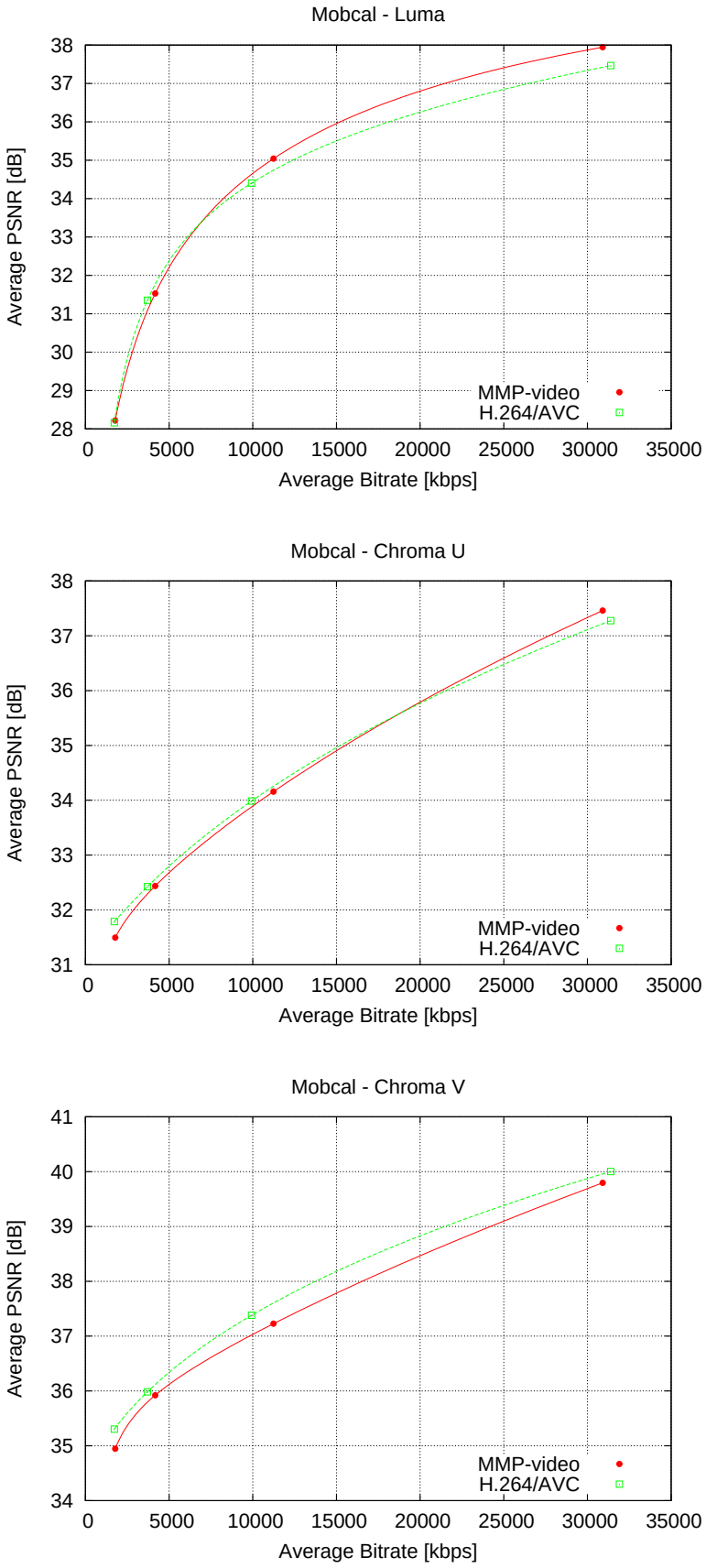


Figure D.9: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Mobcal sequence (720p).

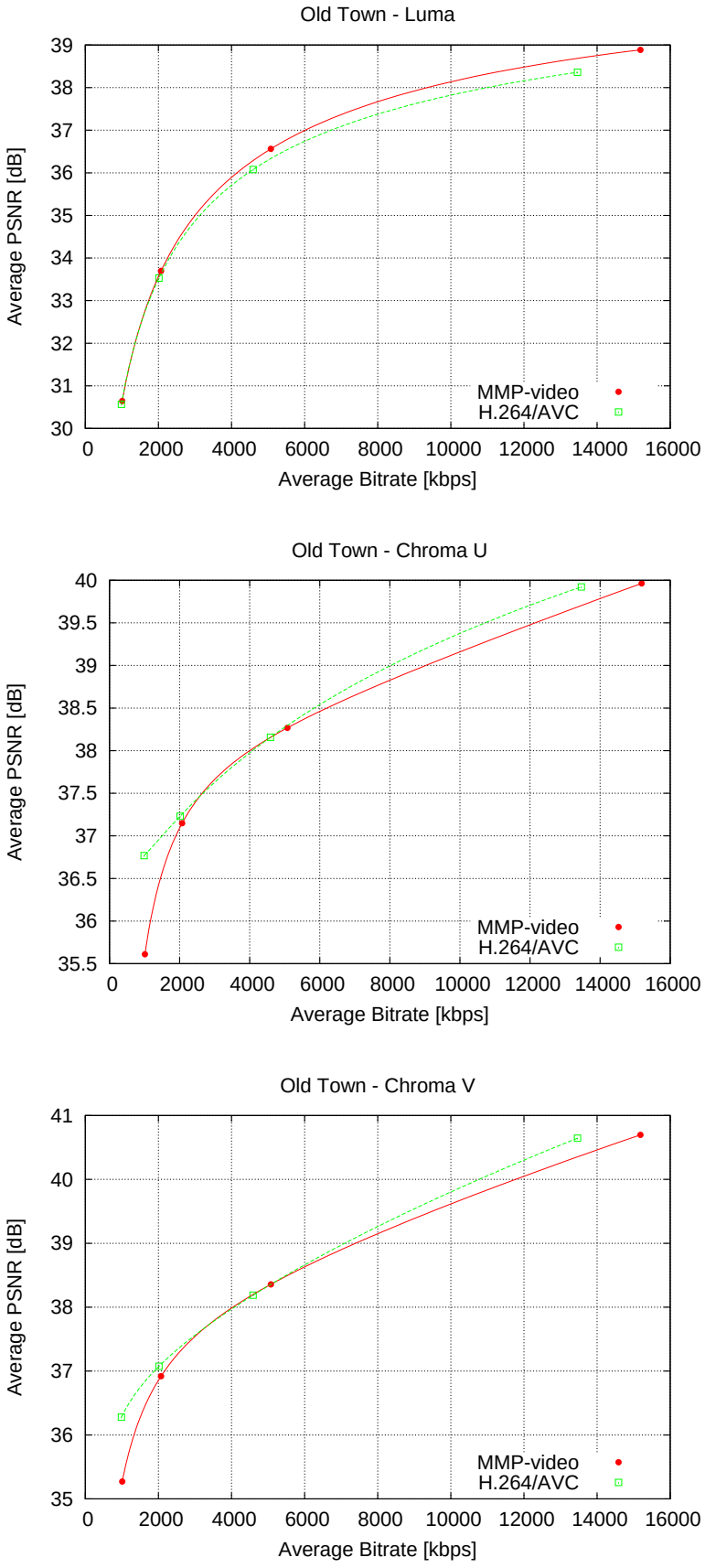


Figure D.10: Comparative results for the MMP-Video encoder and the H.264/AVC high profile video encoder, for the Old Town Cross sequence (720p).

The high profile, RD optimization and the use of Intra MB in inter-predicted frames were enabled, while no error resilience tools and no weighted prediction for B frames were used. The context-based adaptive arithmetic coder (CABAC) option was set for both encoders. For ME, we used a Fast Full search with ± 16 search range and 5 reference frames. Variable bit rate mode was adopted and the encoders were tested for several quality levels of the reconstructed video sequence, by setting the QP parameter for the I/P and B slices separately [83]. Four distinct combinations of QP values were used: 23-25, 28-30, 33-35 and 38-40.

Figures D.5 to D.8 present the average PSNR of all frames *vs.* bitrate, for the first 120 frames of each video sequence, in order to evaluate the global performance of the proposed method when compared with JM17.1 reference software, for each color component of the video sequences. Figures D.9 and D.10 present the average PSNR of all frames *vs.* bitrate, for the first GOP of the video sequences. Considering that the dictionary and the arithmetic encoder statistics are reset for each GOP, these results are representative of the behavior of the algorithms for longer sequences.

Third degree polynomial functions were used to interpolate the four R-D points (one for each QP) obtained with each encoder for each sequence. This approach provides a clear visual interpretation of the obtained results, as well as a low interpolation error, when compared to the exhaustive test of all available QPs [84].

In order to better demonstrate the comparative results of our method, we also computed the Bjøntegaard delta (BD) PSNR [84] for each colour component. This measure reflects the average PSNR gain of the proposed method relatively to JM17.1, along all the tested bitrate range, and can be seen in Table D.1. This table also summarizes the results presented in Figures D.5 to D.10.

The BD-PSNR is a metric that has been widely used to compare results between distinct encoders, specially when the results are close to each other. Because it is computed as the average gain in the interval of overlapping bitrates for the results, it provides a reliable indication of which encoder performs better, in average. This measure is specially useful when the plots present several intersections and it is difficult to clearly identify, by mere visual inspection of the plots, which encoder has the best performance.

As can be seen from BD-PSNR, the proposed method is able to globally outperform state-of-the-art H.264/AVC video encoder for all the tested sequences, at all compression ratios, and for all the color components for the tested CIF sequences. Analyzing the rate *vs.* distortion behavior of each encoder for each separate color component, it can be seen that MMP-Video surpasses H.264/AVC, except for the chroma components, at high compression ratios (highest QP tested). However, the BD-PSNRs tell us that, on average, the proposed method brings better results than H.264/AVC. It can be seen that the performance advantage of the proposed algorithm generally increases for sequences with a high degree of non-smooth elements and high amount of motion. The gains achieved

Table D.1: Comparison of the global R-D performances between MMP-video and the H.264/AVC JM 17.1. The BD-PSNR corresponds to the performance gains of MMP-video over H.264/AVC.

		H.264/AVC				MMP-Video				BD-PSNR		
	QP	**BR**	**Y**	**U**	**V**	**BR**	**Y**	**U**	**V**	**Y**	**U**	**V**
	[I/P-B]	[kbps]	[dB]	[dB]	[dB]	[kbps]	[dB]	[dB]	[dB]	[dB]	[dB]	[dB]
Bus	23-25	2223.56	39.07	42.52	44.28	1825.34	38.48	43.14	44.86	**0.54**	**0.47**	**0.50**
	28-30	1126.33	35.03	40.18	41.95	926.81	34.51	40.39	42.14			
	33-35	560.95	31.24	38.52	40.02	482.17	30.97	38.32	39.83			
	38-40	274.56	27.73	37.39	38.61	254.88	27.83	36.79	37.98			
Calendar	23-25	2384.86	38.53	39.32	39.95	2057.89	38.43	40.09	40.57	**0.77**	**0.72**	**0.67**
	28-30	1212.11	34.34	36.15	36.79	1087.08	34.45	36.77	37.32			
	33-35	606.44	30.22	33.46	34.05	559.36	30.54	33.72	34.29			
	38-40	298.52	26.46	31.60	32.17	277.89	26.78	30.71	31.36			
Foreman	23-25	700.09	40.22	43.26	46.08	667.82	40.49	43.90	46.60	**0.33**	**0.14**	**0.20**
	28-30	332.99	37.21	41.18	43.83	314.49	37.33	41.49	44.23			
	33-35	172.23	34.31	39.71	41.77	166.13	34.43	39.50	41.62			
	38-40	94.71	31.46	38.55	39.78	96.08	31.71	37.46	38.56			
Tempete	23-25	2121.89	39.11	40.27	41.84	1756.62	38.52	40.93	42.32	**0.41**	**0.32**	**0.20**
	28-30	897.94	34.66	37.47	39.54	808.16	34.63	37.81	39.68			
	33-35	403.09	31.16	35.31	37.73	390.02	31.39	35.18	37.60			
	38-40	188.55	27.91	33.80	36.39	186.79	28.17	33.05	35.85			
Mobcal	23-25	31392.50	37.47	37.28	40.00	30906.05	37.94	37.46	39.79	**0.10**	**-0.09**	**-0.29**
	28-30	9927.13	34.40	33.99	37.38	11235.7	35.04	34.16	37.23			
	33-35	3703.90	31.35	32.42	35.98	4169.38	31.53	32.44	35.92			
	38-40	1724.65	28.16	31.79	35.30	1779.97	28.22	31.50	34.95			
Old Town	23-25	13464.03	38.36	39.92	40.64	15186.57	38.89	39.96	40.69	**0.16**	**-0.21**	**-0.21**
	28-30	4591.38	36.08	38.16	38.19	5075.72	36.56	38.27	38.36			
	33-35	2013.83	33.52	37.23	37.07	2072.52	33.70	37.15	36.92			
	38-40	986.63	30.57	36.77	36.28	1009.50	30.64	35.61	35.27			

over JM17.1 are more noticeable for sequences like Mobile&Calendar than for sequences with less activity, like Foreman, because of the high degree of adaptivity presented by MMP. The transform-based approach relies on the assumption that most of the transform coefficients representing the highest frequencies are of little importance, and can be submitted to a coarse quantization or even neglected. However, since this is not a valid model for high activity sequences, this results in a decrease of the algorithm's performance. On the other hand, MMP does not make any assumption about the spectral content of the input sequences and has the ability to adapt its dictionary to their particular features, as it grows along the encoding process. These lends MMP a high performance for non-smooth signals, when compared with transform-based algorithms.

For the case of the 720p sequences, it can be seen that the proposed codec is able to outperform H.264/AVC while encoding the luma component, but this tendency is inverted for the chroma components. However, it is important to notice that the chroma components are responsible for a small percentage of the total bitrate of the compressed video sequence when a 4:2:0 color subsampling is used. Thus, it would be possible to

Table D.2: Comparison of the R-D performances by slice type between MMP-video and the H.264/AVC JM 17.1 for the Bus sequence. The BD-PSNR corresponds to the performance gains of MMP-video over H.264/AVC.

		H.264/AVC				**MMP-Video**				**BD-PSNR**		
Frm type	**Qp** [I/P-B]	**Avg bits/frm** [bits]	**Y** [dB]	**U** [dB]	**V** [dB]	**Avg bits/frm** [bits]	**Y** [dB]	**U** [dB]	**V** [dB]	**Y** [dB]	**U** [dB]	**V** [dB]
I	23-25	209086	40.64	43.27	44.90	209554	40.73	43.65	45.31	**0.19**	**0.16**	**0.16**
	28-30	137741	36.42	40.42	42.23	130490	36.04	40.56	42.36			
	33-35	84127	32.45	38.56	40.05	76143	32.03	38.35	39.78			
	38-40	45218	28.64	37.36	38.52	42160	28.53	36.74	37.77			
P	23-25	115350	40.08	42.70	44.40	113696	40.64	43.59	45.22	**0.73**	**0.30**	**0.35**
	28-30	63380	35.99	40.17	41.93	59561	36.12	40.48	42.26			
	33-35	35523	32.04	38.44	39.95	30721	32.17	38.28	39.83			
	38-40	15760	28.34	37.32	38.56	15533	28.69	36.74	37.97			
B	23-25	44125	38.51	42.38	44.18	24786	37.40	42.91	44.68	**1.48**	**1.10**	**1.27**
	28-30	17186	34.50	40.16	41.93	9421	33.71	40.34	42.07			
	33-35	6622	30.80	38.55	40.04	4159	30.38	38.33	39.83			
	38-40	2898	27.40	37.42	38.64	2268	27.42	36.79	37.98			
Global	23-25	74116	39.07	42.52	44.28	60813	38.48	43.14	44.86	**0.54**	**0.47**	**0.50**
	28-30	37542	35.03	40.18	41.95	30863	34.51	40.39	42.14			
	33-35	18696	31.24	38.52	40.02	16041	30.97	38.32	39.83			
	38-40	9149	27.73	37.39	38.61	8465	27.83	36.79	37.98			

trade off some of the PSNR advantage of the luma component to obtain more rate to encode the chroma components with less distortion, achieving a rate-distortion performance advantage for all components.

In Table D.2, results for sequence Bus are presented separately for I, P e B slices, in order to allow a more detailed evaluation of MMP-Video's results. The results for Intra frames are consistent with those presented for still image encoding [46, 123]. MMP-video is able to outperform H.264/AVC for all sequences, except for the chroma components at high compression ratios. The performance of both encoders is closer for I slices, with MMP-video being, on the average, marginally superior to H.264/AVC. Despite the small gain observed for the I references, MMP-video has the ability to generate considerably better P slices than H.264/AVC (note the BD-PSNR for P frames). This results from a better performance in the compression of MP residues, that is even clearer for B slices.

The previous results clearly demonstrate the advantage of the proposed video encoder over the JM17.1 H.264/AVC reference software. Nevertheless, they do not reveal if the performance advantage comes from the substitution of the integer transform by the MMP algorithm or from the additional prediction modes used.

Figure D.11 shows the results for sequence Foreman, compressed with two versions of MMP-video: with and without the LSP modes. These results are also compared with those from the JM17.1 H.264/AVC reference software. This way, it is possible to evaluate the improvements that resulted from the substitution of transforms by the MMP algorithm and from the extra prediction modes used.

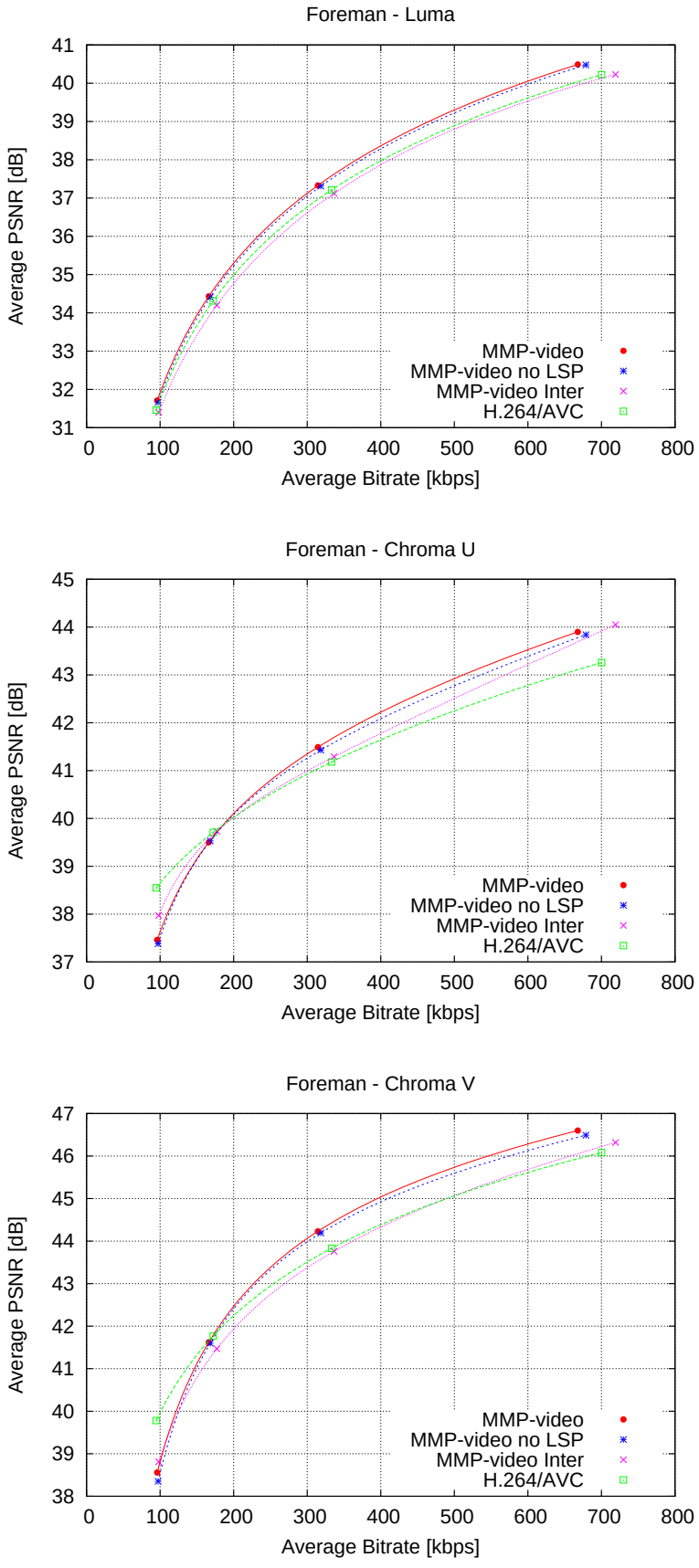


Figure D.11: Comparative results for the MMP-Video encoder with and without the LSP prediction modes, and the H.264/AVC high profile video encoder, for the Foreman sequence (CIF).

It can be seen that the MMP-based video encoder is able to outperform JM17.1 even without the use of the additional prediction modes. The BD-PSNR values are in this case 0.25 dB, 0.08 dB and 0.09 dB, for the luma and the chroma components, respectively. When using the two additional LSP modes, the BD-PSNR increase 0.08 dB for the Luma and 0.08 dB and 0.11 dB for the chromas, to respectively 0.33 dB, 0.14 dB and 0.20 dB, as seen on Table D.1.

It is important to notice than the most significant gain achieved by the additional prediction modes occurred for the V component, where the linear prediction is able to deliver better results, as it is calculated based on two other components. For the U component only Y is used to compute the linear prediction.

Similar results where verified for other sequences, demonstrating not only that the pattern matching paradigm is able to outperform the JM17.1 encoder, but also that the additional prediction modes are successful while increasing the performance of the final proposed video codec.

Figure D.11 also includes the results for the encoder presented in [7, 14, 79], that used MMP to compress only the ME residue. We refer to this encoder as MMP-video Inter. As it can be seen, the performance of this encoder was close to that from the JM H.264/AVC reference software. The experimental results presented on [7] shown a similar performance between JM and MMP-video Inter, with the last presenting a performance advantage when compressing B slices ME residue. For the case of Intra slices, the performance of both methods is equivalent, since MMP-video Inter uses the same encoding tools to compress these slices.

In the results presented in [7, 14, 79], the performance gains are emphasized by the large GOP used in the test setup. These previous works adopted a GOP size of 100, that is of little practical utility, as synchronization times of more than 3 seconds are not acceptable for most applications. This setup was adopted in order to enhance the impact of the ME residue compression in the overall performance of the algorithm, as the ME performance tends to degrade due to the large temporal distance between reference frames. If a small GOP was used in this case where Intra slices are encoded using the same tools from H.264/AVC, the performance advantage on the ME residue compression would present a very little impact on the overall codec rate-distortion performance, and the more efficient compression of the ME residue would be hard to evaluate.

Furthermore, it can be seen that the gains achieved when the two additional LSP modes are used and the performance advantage over the MMP-video Inter encoder [7, 14, 79] remain consistent for the other tested sequences. Thus, we will not present extensive results for all sequences. These results demonstrate that the performance increase of the new fully pattern-matching-based algorithm is relevant, relatively to the previous video MMP-based encoding algorithm.

It is important to point out that, as a pattern matching method, MMP-video's R-D

performance advantage comes at the expense of a higher computational complexity, when compared to H.264/AVC, as discussed on Appendix B. This can be an obstacle for some applications. However, for applications where the input video sequence is only encoded once, and decoded many times, the impact of the higher computational complexity tends to be smaller, and may be justified by the gains in the R-D performance.

Despite the work described on this appendix being focused on the algorithm's R-D performance, future research in computational complexity reduction will bring an important contribute to the practical application of the proposed video codec. Some computational reduction techniques are proposed on Appendix E, and the increasing computational power available in multi-core processors and GPU's can have an important role in affirming pattern matching methods as viable alternatives to the transform-based paradigm, as pattern matching algorithms generally involve repetitive integer precision operations, with a high potential for parallelization.

D.5 Conclusions

In this appendix, we presented MMP-video, a video compression algorithm based on multiscale recurrent patterns. The proposed encoder adopted the use of the Multidimensional Multiscale Parser (MMP) algorithm to encode both the Intra prediction and the Motion Estimation residues, replacing the traditional transforms and quantization used in state-of-the-art image and video encoders. This way, the use of transforms and quantization is totally abolished in the proposed codec, resulting in an algorithm entirely based on the pattern matching paradigm. The proposed method presents results competitive with those from the state-of-the-art transforms-quantization-entropy encoders, like H.264/AVC.

Several functional optimisations for the MMP algorithm were investigated, specially oriented to the video signal characteristics. Experimental tests have shown that, in spite of its larger computational complexity, MMP-video is able to outperform the H.264/AVC JM17.1 reference software in terms of the rate-distortion performance, specially for medium to high bitrates, being particularly efficient while encoding bi-predicted slices.

Appendix E

Computational complexity reduction techniques

E.1 Introduction

As seen on the previous appendices, a high degree of adaptability allows MMP to outperform state-of-the-art compression methods for a wide range of applications. However, as other pattern matching methods, MMP presents a high computational complexity, previously discussed on Appendix B, which presents an important drawback for most practical applications.

In applications where input data needs to be encoded only once and decoded many times, a high encoding computational complexity can be justified by a superior rate-distortion performance. Nevertheless, MMP's decoder also presents a considerable computational complexity, limiting its application on receivers with low resources. To overcome this issue, several computational complexity reduction techniques which can be applied to any MMP based algorithm were studied.

In [85], one of these methods has been proposed for the MMP encoder. However, this method only has impact in the encoder, and can cause rate-performance losses of up to 1 dB, due to its non-exhaustive optimization nature.

In this appendix, we discuss the critical time consuming processes in the MMP algorithm and propose two new computational complexity reduction techniques. The optimizations done on previous works are briefly described in Section E.2, while the two new proposed techniques are described in Section E.3. Experimental results comparing the two methods with a previous fast implementation and the benchmark version of MMP are presented in Section E.4. Overall conclusions of this research are summarized on Section E.5.

E.2 Previous computational complexity reduction methods

The high computational complexity presented by MMP-based algorithms has motivated the search for efficient implementations. As a result, some modifications, which reduced its computational complexity, were proposed. Some of these techniques were aimed at complexity reduction, but in some cases, this was a collateral effect of some rate-distortion optimisation technique.

Some of the proposed methods do not have any impact on the rate-distortion performance of the method, while others may reduce the rate-distortion performance of the algorithm.

In the next sections, we describe some of the proposals with major impact on the computational complexity of MMP-based encoders, classified in accordance to the existence or not of any rate-distortion performance losses relatively to the original algorithm.

E.2.1 Methods with no impact in the rate-distortion performance

Two optimizations were previously proposed to accelerate the time required by the dictionary searches. As these optimizations did not have any impact in the rate-distortion performance of the algorithm and were successful in reducing its computational complexity, they were adopted as implementation related modifications, and became part of the MMP algorithm.

The first, was the use of a memory table containing pre-calculated results for exhaustively performed operations. The calculation of squared values is an example of such operations, as it is intensively performed to calculate the sum of squared differences (SSD), used to determine the distortion of each block. Another example is the calculation of logarithms, performed many times to estimate the rate required by each index of the dictionary, while calculating their lagrangian cost. By using this approach, arithmetic operations are replaced by memory accesses, which in our experimental tests have shown to be less time consuming operations.

The second optimization uses the difference between the average of the original block and that from the codeword being tested, to avoid the SSE calculation for blocks which present a high distortion. The difference between the averages can be compared with the best match found so far, and if the average differences are very high, the blocks are known to be a worst match than the existent one [4]. Every time a better match is found, this comparison allows to discard more blocks with different averages. An additional table is used to store the average of each codevector of the dictionary, so this value is computed only once for each block.

E.2.2 Methods with impact in the rate-distortion performance

Other proposed modifications, with impact in the MMP's computational complexity, also affect the rate-distortion performance of the algorithm.

As seen in [49], the proposed redundancy control tool not only increased MMP's rate-distortion performance, by avoiding the insertion of useless codevectors, but also had a positive impact in reducing the computational complexity of the algorithm. By restricting the insertion of new codevectors on the dictionary, searches became less time consuming, as in average less codevectors need to be tested for each matching procedure. A similar effect was achieved by the scale limitations. As the new generated codevectors are inserted in fewer scales, the dictionary experiments a slower growth, reducing the time required for each search.

A parameter that also has a large impact on the computational complexity of the MMP-based encoders is the initial block size. Equation B.16 clearly shows the dependency of the computational complexity with the initial block size, and experimental results regarding this topic were presented in [49]. However, the conclusion of the experimental tests presented in [49] suffered some variations when the flexible segmentation scheme [5] was introduced. The rate-distortion performance is more affected in a codec which uses the flexible partition scheme, but the gain in computational complexity is also higher. This happens because each initial block can be further divided according to more options, and the use of smaller initial blocks eliminates more scales from the dictionary, resulting in a considerably lower number of searches, while optimizing each block.

Similarly, the limitation of the maximum capacity of the dictionary also has a significant impact on both the computational complexity and the rate-distortion performance of MMP-based algorithms. Larger dictionaries demonstrated to be more efficient from a rate-distortion point of view. Nevertheless, they increase the computational complexity of the algorithm. This topic was also discussed in [4].

In [85], another approach was proposed, oriented towards the predictive MMP-based algorithms. In the original MMP algorithm, each block of the input image generates a single segmentation tree, that needs to be optimized. In a predictive scheme, a different segmentation tree has to be optimized for each prediction mode, increasing significantly the computational complexity of the algorithm. This problem is further aggravated by the recursiveness of the predictive scheme.

The method proposed in [85] uses the energy of the generated residues to estimate the best prediction mode. The mode with the lowest energy residue block is chosen, and only its respective residue block is optimized. It is important to notice that in a MMP point-of-view, a lower energy is not a guarantee that the block will be encoded with a lower Lagrangian cost. Nevertheless, experimental results show that this is a good approximation [85].

Considering the best prediction mode as the one which originates the lowest energy residue block, allowed to considerably reduce the computational complexity of the encoder, at the expense of a reduction on the rate-distortion performance. Additionally, the decoder's complexity was not reduced. As a collateral effect of the error energy-based optimization, the dictionary tends to grow even more, and thus the time required to decode an image often increases.

E.3 New computational complexity reduction methods

In this section, we propose two new computational reduction techniques for MMP based encoders. These two techniques are related with the two most computationally exhaustive steps of the algorithm, namely the dictionary searches and the segmentation tree optimization.

E.3.1 Dictionary partitioning by Euclidean norm

The most time consuming task on pattern matching algorithms is the search for the best match amongst the dictionary codewords. For each input block, the sum of squared errors (SSE) has to be computed for all codewords, in an exhaustive and time consuming process. The use of large dictionaries and large blocks considerably contributes to increase this problem, resulting in millions of mathematical operations to be performed. Furthermore, the algorithm's recursiveness makes each input block to be compared with codewords from different dictionary scales, increasing the total number of matching operations which need to performed. For algorithms that use adaptive dictionaries, the updating stage also may require exhaustive searches for existing codevectors that are similar to the new pattern, in order to avoid redundancy. For these cases, a similar operation is required on the decoder, to maintain a synchronized copy of the codebook.

A careful organization of codewords may give an important contribution in accelerating the searching processes. For example, if codewords are sorted by ascending Euclidean norms, it is possible to start searching for the best match of a block X^l, in codewords with a norm close to $\|X^l\|$. The search can then proceed to find the global optimum. The lowest distortion, D, found at each moment, may then be used to restrict the searching region. In this case, all the codewords with norms outside $[\|X^l\| - \sqrt{D}; \|X^l\| + \sqrt{D}]$, are known to have a distortion larger than D, and consequently do not need to be tested.

However, sorting the dictionary every time a new codeword is inserted is a cumbersome task, which easily overrides the gains achieved by the more efficient searching approach. This issue could be overcome, for example, by using norm-based indexation for the dictionary elements. Codewords would remain disposed arbitrarily in the dictionary, and an additional field would indicate each codeword's norm. Thus, those with the closest

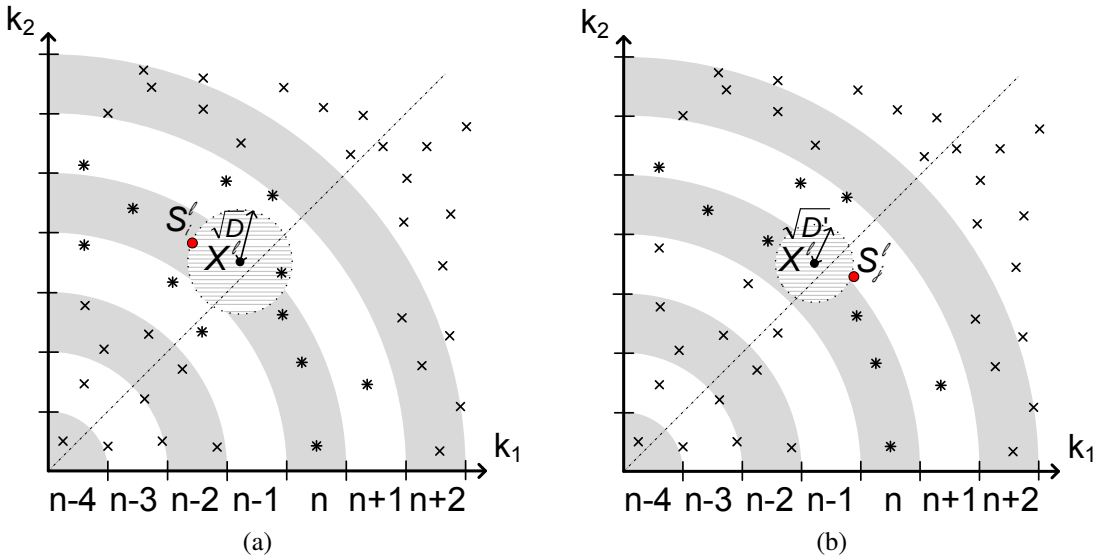


Figure E.1: Searching region for a two-dimensional input block X^l, using a distortion restriction.

norms could be tested first, and then the algorithm would progressively skip the remaining ones. However, this approach would impose a large amount of memory jumps, which are also known to be very time consuming operations.

We propose a method that combines the two previously referred techniques, in order to overcome the problems arisen by each one. The dynamic range of possible norm values is divided into N slots, with codewords being disposed sequentially inside the slot they correspond to. With this approach, codewords inside each slot can be processed sequentially, minimizing the number of memory jumps, while the existence of distinct slots preserves the ability to discard codewords with distant norms, that do not need to be tested.

Consider a generic codeword X^l, with Euclidean norm $\|X^l\|$. If an exact match exists for X^l, it will belong to the norm slot n, whose boundaries contain $\|X^l\|$. The slot n will be taken as the starting point for the search, and the codeword, S_i^l, that currently better represents X^l, with a distortion D, can be used to further restrict the search. In a strict distortion optimization, the best match must belong to a norm slot contained in the interval $[\|X^l\| - \sqrt{D}; \|X^l\| + \sqrt{D}]$. After processing the slot n, the algorithm then proceeds sequentially to the slots $n + k$ and $n - k$, for increasing values of k. Every time a best match is found, the value of D decreases and the searching region is potentially reduced, reducing the maximum value for k. The process will converge once all the slots contained in the interval $[\|X^l\| - \sqrt{D}; \|X^l\| + \sqrt{D}]$ have already been tested. This way, it is expected that most of the norm slots can be discarded, without the need for testing all the codewords they comprehend.

Figure E.1 represents the search region for a two-dimensional block X^l. Note that this analysis is extensive to higher dimensions, but we adopted a two-dimensional example because of the clarity of representation. In this case, two-dimensional norm slots

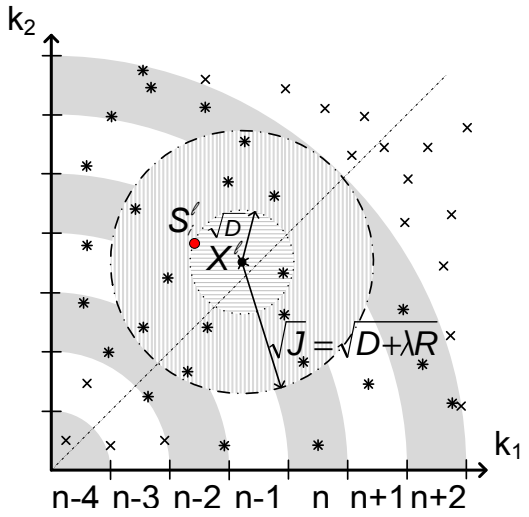


Figure E.2: Searching region for a two-dimensional input block X^l, using a Lagrangian cost restriction.

corresponds to concentric regions in the two-dimensional plane.

As $\|X^l\|$ belongs to slot n, this slot will be the starting point for the search, with S_i^l being the best match found in this slot (Figure E.1a). The distortion between X^l and S_i^l (which is equal to the euclidean distance between X^l and S_i^l), allows to determine the maximum searching region, as all vectors that belong to norm slots which are not intersected by the circle with radius $\sqrt{D}$, are known to present greater distortion relatively to X^l than the current best match (S_i^l). This restricts the searching region to only the slots $n+1$, n and $n-1$. In other words, only the codewords represented as * need to be tested, and all the codewords represented as x can be discarded without the risk of loosing the same solution found found using the exhaustive searching approach.

If at any point, a block with a lower distortion S_j^l is found, the new distortion D' will narrow the searching region, and thus will allow to also discard the vectors from other slots (in the case of the example of Figure E.1b, the slot $n-1$).

Note that the described approach is only applicable for strict distortion optimization. When a rate-distortion optimization is used, as in the case of MMP, the search region will not only depend from the distortion D, but also from the representation rate R. Hence, the searching region will depend from the lagrangian cost J, instead of the distortion D. This is so because the optimization procedure may select codewords with higher distortion, if the rate required for its representation is sufficiently low to compensate the difference between the distortion values. In this case, the amplitude of the search region will be larger, because of the λR term, with a direct dependence on λ.

A higher value of λ imposes a larger region to be tested, in order to guarantee that the solution provided by the exhaustive optimization is always reachable, since the rate component becomes more relevant on the optimization. This situation is illustrated on Figure E.2. In this case, S_i^l restricts the search to slots $n-2$ to $n+2$, because of the

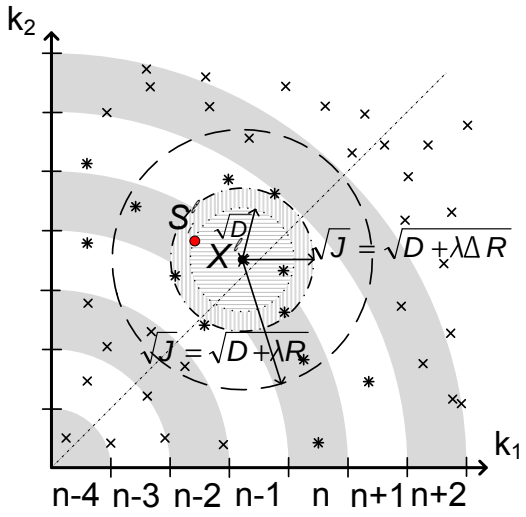


Figure E.3: Searching region for a two-dimensional input block X^l, using a differential lagrangian cost restriction.

additional λR term.

However, codewords located in the frontier of the searching region could only be the better solution if it was possible to represent them using a null rate, which is known to be impossible. For this reason, we can reduce the searching region radius to $\sqrt{J} = \sqrt{D + \lambda(\Delta R)}$, with ΔR corresponding to the difference between the rate required to encode the current best match, and the minimum rate required to encode any codewords from scale l of the dictionary. This may result in a more restrictive but still optimized search (for slot $n-1$ to $n+1$ in the example of Figure E.3), in a rate-distortion point-of-view.

An additional field containing the average of each codeword was also included in the dictionary. This allows the exclusion of codewords, inside a norm slot, that, despite having similar norms, are located on distant regions of the space. Consider S_i^l and $-S_i^l$, with this last presenting the same norm but symmetric coordinates, relatively to S_i^l. Assuming a two-dimensional space, if S_i^l is in the first quadrant, $-S_i^l$ will be located on the third quadrant. If only the norm classification was considered, both codewords have the same norm, and would belong to the same norm slot. As a result, if S_i^l is the lowest lagrangian cost solution in a rate-distortion point-of-view, $-S_i^l$ would also be tested, despite of the high distortion associated. In order to avoid that codevectors in this situation are tested, the average of X^l is compared with that of each vector before proceeding to the distortion calculation, and if the averages' difference is significant, it would be a sufficient condition to discard the block.

In the case of the two-dimensional example, presented on Figures E.1 to E.3, this would eliminate all vectors from the third quadrant, as well as most of the vectors from the second and fourth quadrants, from the slots which need to be processed.

The number of norm slots in the dictionary is a factor which significantly impacts the

performance of the proposed method. A high number of slots can be more effective in reducing the search range, but generally imposes a large amount of computationally costly memory jumps. The amount of memory jumps can easily override the gains achieved by the more efficient search, so the trade-off between the number of codewords tested in each slot and the number of memory jumps, will define an optimized value for N.

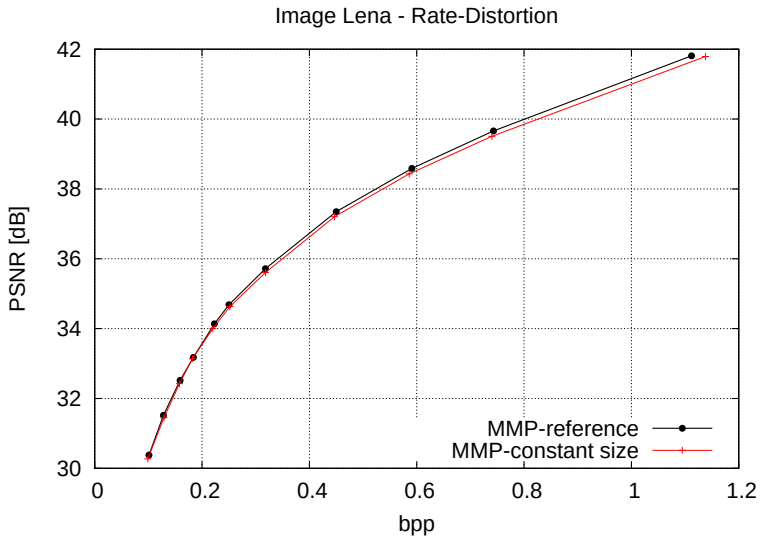
As the MMP algorithm uses a multiscale approach, the value of N was optimized for each dictionary level l. The optimization for each scale is justified by the fact that when the dimensionality increases, the vectors tend to be more sparsely distributed in space. Experimental tests were used to determine the suited value of N for each scale, which has been shown to be properly represented by:

$$N(l) = \left\lceil \frac{\sqrt{Range^2 * Height(l) * Width(l)}}{4} \right\rceil, \tag{E.1}$$

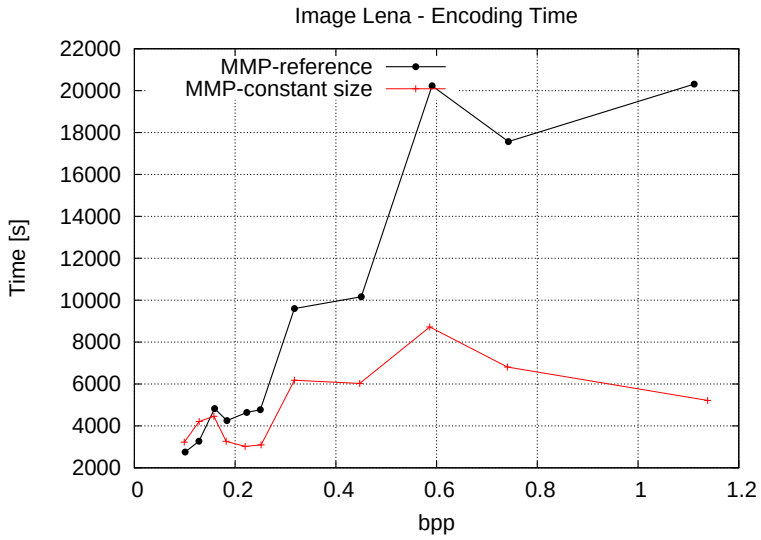
where $Range$ represents the dynamic range of the input signal (255 for 8 bit depth images) and $Width(l)$ and $Height(l)$ the dimensions of the blocks from scale l.

The dictionary from each scale was first divided into $N(l)$ slots with equal capacity, which added up to the maximum dictionary capacity (MDC). However, since codewords need to be discarded whenever the slot's maximum capacity is reached, a particular issue occurred with this approach. Residue blocks have a norm distribution highly peaked near zero, and lower norms slots became full much earlier than those corresponding to higher norms. Consequently, the algorithm was forced to discard codewords which would be available in a non-segmented dictionary, and would be useful in the future. In other words, this will limit the dictionary growth in the most populated regions, with a negative impact in the overall rate-distortion performance of the encoder.

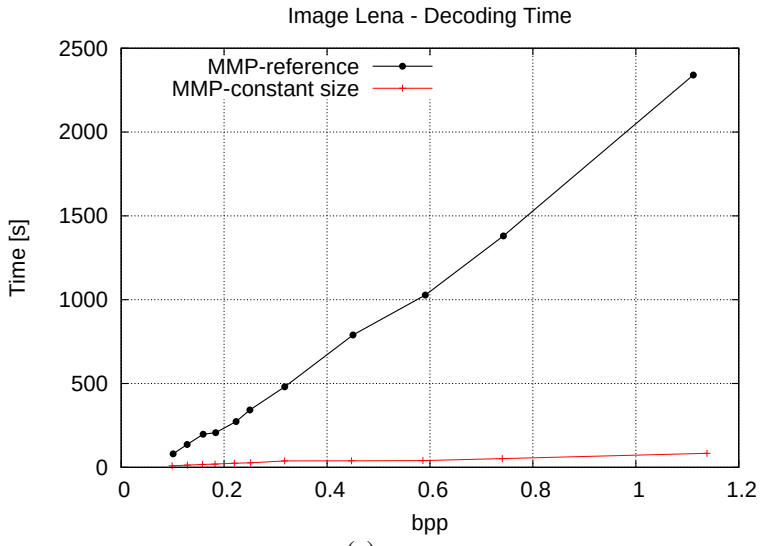
Figures E.4 and E.5 show the results achieved by the MMP encoder using the dictionary partitioned into constant amplitude norm slots, while compressing images Lena and Barbara at several bitrates. The reference implementation of MMP is used for comparison purposes. A rate-distortion performance loss up to 0.2dB can be shown on Figures E.4a and E.5a. The performance loss increases at lower compression ratios, for two reasons. First, better matches, required at high bitrates, become more difficult in smaller dictionaries. Second, a larger number of new patterns are created due to the lower distortion, and most of these patterns, which are made available on the original algorithm need to be discarded with this approach. At higher compression ratios, the differences on rate-distortion performance are almost negligible, because the dictionary growths is more moderated in this case, and the slots are unlikely to overflow.



(a)

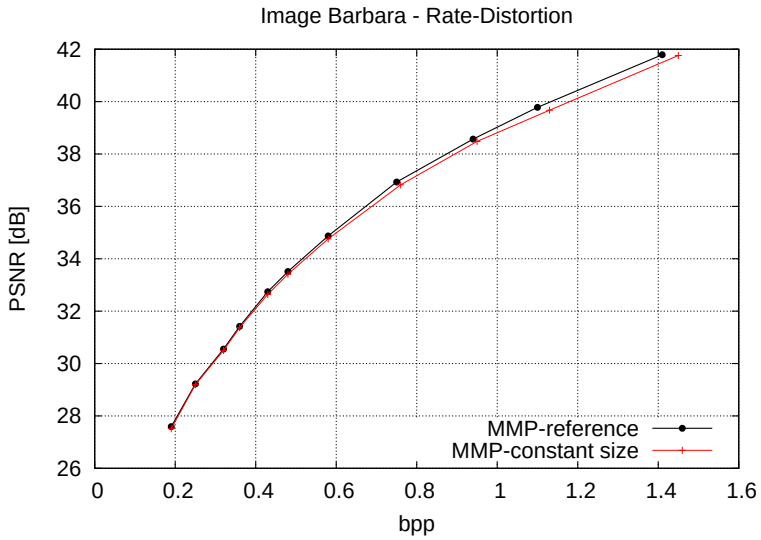


(b)

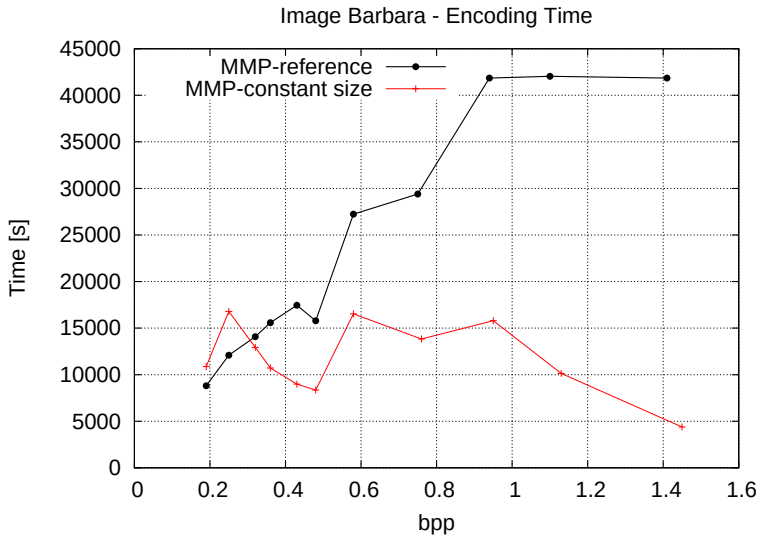


(c)

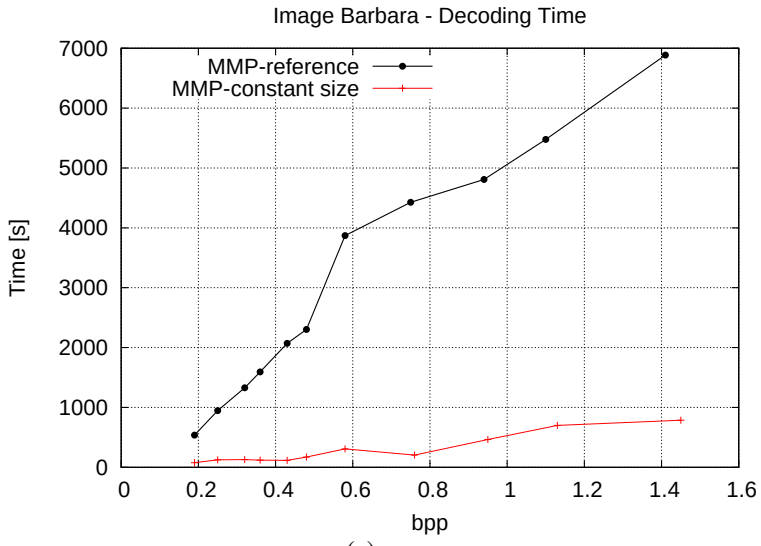
Figure E.4: Performance results for image Lena using constant sized norm slots.



(a)



(b)



(c)

Figure E.5: Performance results for image Barbara, using constant sized norm slots.

As can be seen in Figures E.4b and E.5b, the computational complexity of the encoder was reduced up to 74% at high bitrates, but was slightly increased for high compression ratios. This is so because high compression ratios correspond to high values of λ, which are less effective while limiting the radius of the searching region, as illustrated in Figure E.3. Hence a large number of slots need to be tested, that imposes a large number of memory jumps, increasing the time needed to encode a given image.

In the case of the decoding time, the use of constant amplitude slots proved to be beneficial for all compression ratios. Time savings of up to 96% were reached using this approach. In the decoder, the major computational complexity corresponds to searches for similar blocks when new codevectors are created. If the parameter that sets the dictionary's redundancy control is defined as 0, only the norm slot that comprehends the vector's norm value needs to be tested, instead of the entire dictionary. Otherwise, it is straightforward to determine how many norm slots need to be tested and identify them, minimizing memory jumps. This way, the increase on the number of norm slots is usually beneficial from the decoder's computational complexity point-of-view.

A possible solution to minimize the rate-distortion performance losses on the multi-slot approach is to increase the capacity of the slots corresponding to the most populated regions, which are known to get full during the encoding process. This approach trades-off some of the computational gains for less significant performance losses, as the most used slots also will contain more vectors to be tested. However, on the other hand, the existence of more vectors increase the probability of closer matches, which allows to better restrict the searching region in the encoder's rate-distortion optimization.

Based on this observation, two approaches were investigated: the reduction of the dynamic amplitude of slots corresponding to the most populated regions, and the increase of the capacity of these slots. The second approach revealed advantageous after experimental tests.

Several experiments were conducted to determine the optimized cardinality for each slot. A large set of images was encoded with the original MMP algorithm, in order to determine the statistical norm distribution of the generated code-vectors, when no growing restrictions are applied to the dictionary, except for a maximum overall capacity of each level. If the capacity of each slot is close to the number of generated vectors from each hypothetical slot on the non-restricted algorithm, no vectors need to be discarded, and the rate-distortion losses will be null.

The results from the tests revealed two interesting particularities from the codewords norm distribution:

- The use of intra prediction makes the distribution shape highly independent of the input image's type, for a given compression rate;
- The distribution's shape depends on the target distortion, and thus on the value

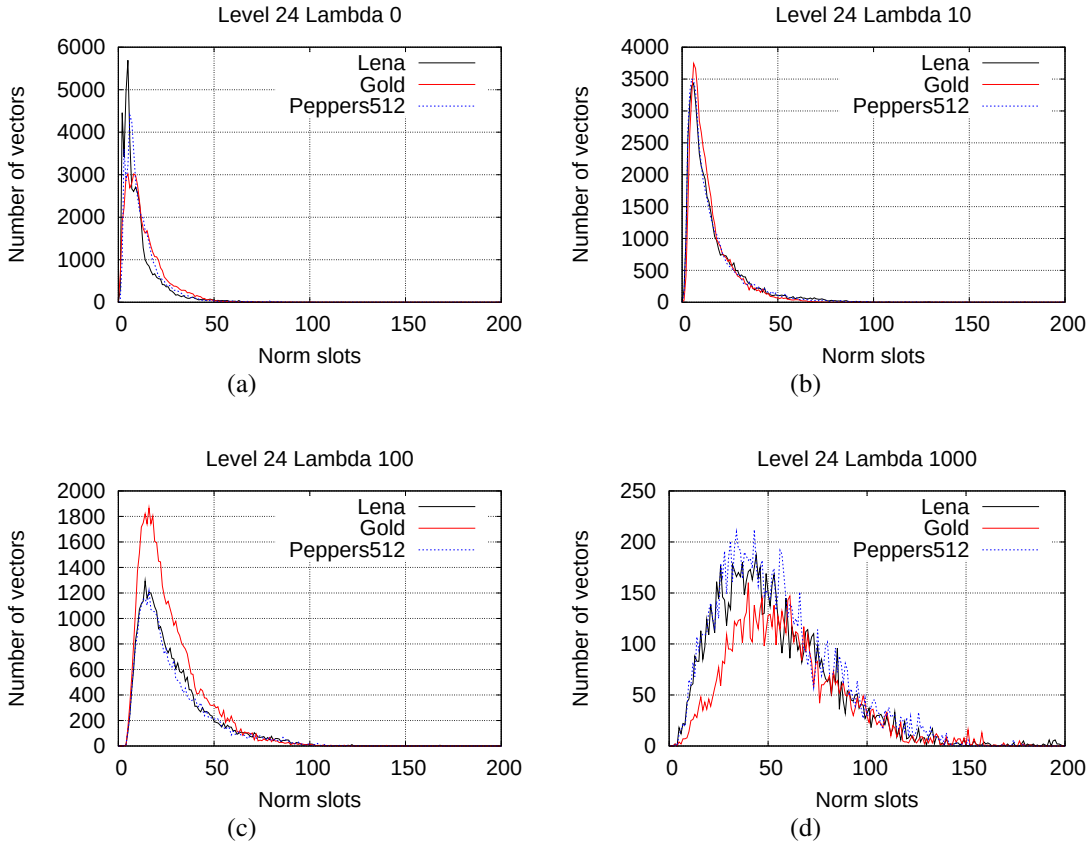


Figure E.6: Norm distribution inside slots for λ a) 0 (lossless), b) 10, c) 100 and d) 1000.

of the lagrangian operator λ. Low distortions mean better predictions, and hence, residue blocks tend to have lower norms, concentrating the distribution around zero.

Figure E.6 shows the vectors distribution for level 24 (16×16 pixel blocks), for 4 different values of λ, namely 0 (lossless), 10, 100 and 1000. In this case, the maximum dictionary dictionary size was set as 50000, but only the distribution across the first 200 norms is presented, to enhance the region of interest. The distribution of codevectors in the remaining region is very low, and would not be visible at the scale adopted to visualize the peak region, especially for low values of λ.

A Rayleigh distribution, varying with the value of λ, was chosen to modulate the capacity of each slot. As observed in the experimental tests, the lower distortion, obtained for small values of λ, produces code-vectors with lower norms. When λ increases, the vectors' norms become more sparse, and the capacity of norm slots needs to be more evenly distributed across the dynamic range, as code-vectors with high norms are most likely to appear.

A minimum capacity was thus set for each slot, that increased with the value of λ. This guarantees that higher norm slots have sufficient capacity to accommodate all vectors, even if the prediction becomes inefficient and their norm distribution becomes sparse. The remaining capacity is distributed across the lower norm slots, using a Rayleigh dis-

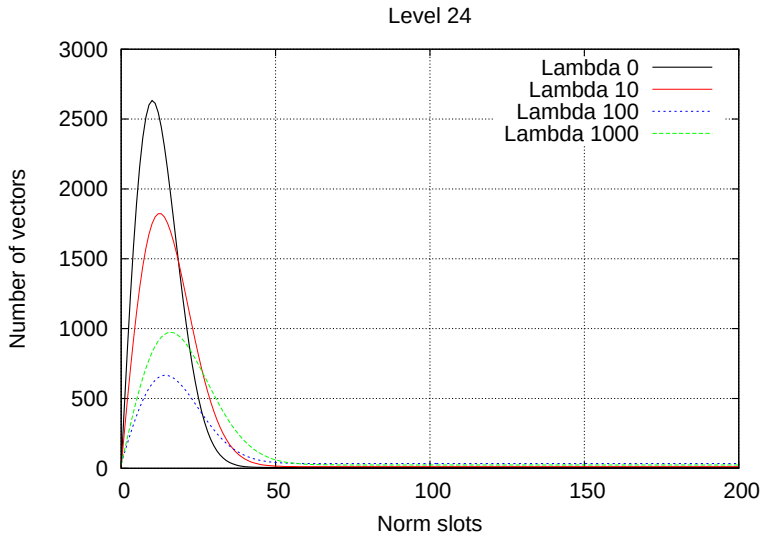


Figure E.7: Norm distribution modulated by Equation E.2 for level 24, using 4 different values of λ.

tribution. Thus, the distribution becomes more peaked near zero when λ decreases.

Using these premisses, we developed an expression that determined the capacity of each slot, independently from the input image's characteristics:

$$C(n) = a\left(\frac{2n}{b}e^{\frac{-n^2}{b}}\right) + c. \tag{E.2}$$

The parameter c guarantees the minimum capacity for each slot, and increases with λ. A logarithmic dependence proved to be suitable to represent the relationship between c and λ:

$$c = \left\lceil MDC\frac{log(\lambda+1)+1}{8}\right\rceil. \tag{E.3}$$

The value a in Equation E.2 corresponds to the remaining elements over the MDC, that are distributed between the lower norm slots, resulting in the expression:

$$a = MDC - c.N(l). \tag{E.4}$$

The value b defines the concentration of the distribution around zero, and is well modeled as:

$$b = \frac{0.2log_{10}(\lambda+1)+2}{2}.N(l). \tag{E.5}$$

The shape of the distribution is presented on Figure E.7, for level 24 (16×16 pixel blocks). The dependence from λ is obvious, with the distribution becoming more peaked around zero for lower values of λ. Note that this function was defined using a conservative criterium, presenting a less peaked shape than the original distribution, obtained when

compressing most of smooth images. This way, experimental tests demonstrated that this distribution is sufficiently peaked to allow a convenient growth of the dictionary, without compromising the performance of MMP in cases where the code-vectors' norm distribution varies from the adopted model, as is the case of text images.

Figures E.8 and E.9 show the results achieved by the encoder using variable amplitude norm slots, when compared with constant amplitude slots and with the reference implementation of MMP. As can be seen, the performance losses are practically avoided. Note that, the computational complexity of the encoder is lower than the one presented by the constant norm slots, except for low values of λ. This unexpected fact has a simple explanation.

In one hand, the use of adaptive slot sizes increases the number of code-vectors in most used slots, which increases the computational complexity. On the other hand, the existence of a richer set of patterns improves the matches, that are very efficient in restricting the search. This way, the search converges more quickly, resulting on lower encoding times. This tendency is however inverted for low values of λ, because in these cases, the amplitude of the searching region is less affected by the λR term. The searches performed to detect redundancy of new generated patterns become also more efficient, and are more significant at lower compression ratios where more new code-vectors are generated.

It is important to note that this method reduced both the complexity of the encoder and of the decoder, unlike the method proposed in [85]. Furthermore, this method can be adapted to be used for other VQ based algorithms, not being specifically oriented towards MMP.

E.3.2 Gradient analysis for tree expansion

Another time consuming task on MMP based encoders is the optimization of the block segmentation pattern. In order to determine the optimized segmentation tree for each input block, MMP performs a hierarchical matching procedure for each scale, down to 1×1 blocks, calculating the lagrangian cost associated to each tree leaf. Since the lagrangian cost depends from both the distortion and the rate required for the representation, the probability of finding the representation with the lowest lagrangian cost for a very homogenous texture using blocks from lower scales is very low. Additional segmentations would require the transmission of several flags and indices, increasing the rate required for the representation. As the distortion of the representation of a homogenous block is unlikely to be sufficiently reduced through successive segmentation to compensate this rate increase, one may expect that this option will in most cases present a higher lagrangian cost.

Thus, in order to avoid testing segmentation patterns that are unlikely to present low lagrangian cost, we propose a new segmentation stopping criterion, based on the block's

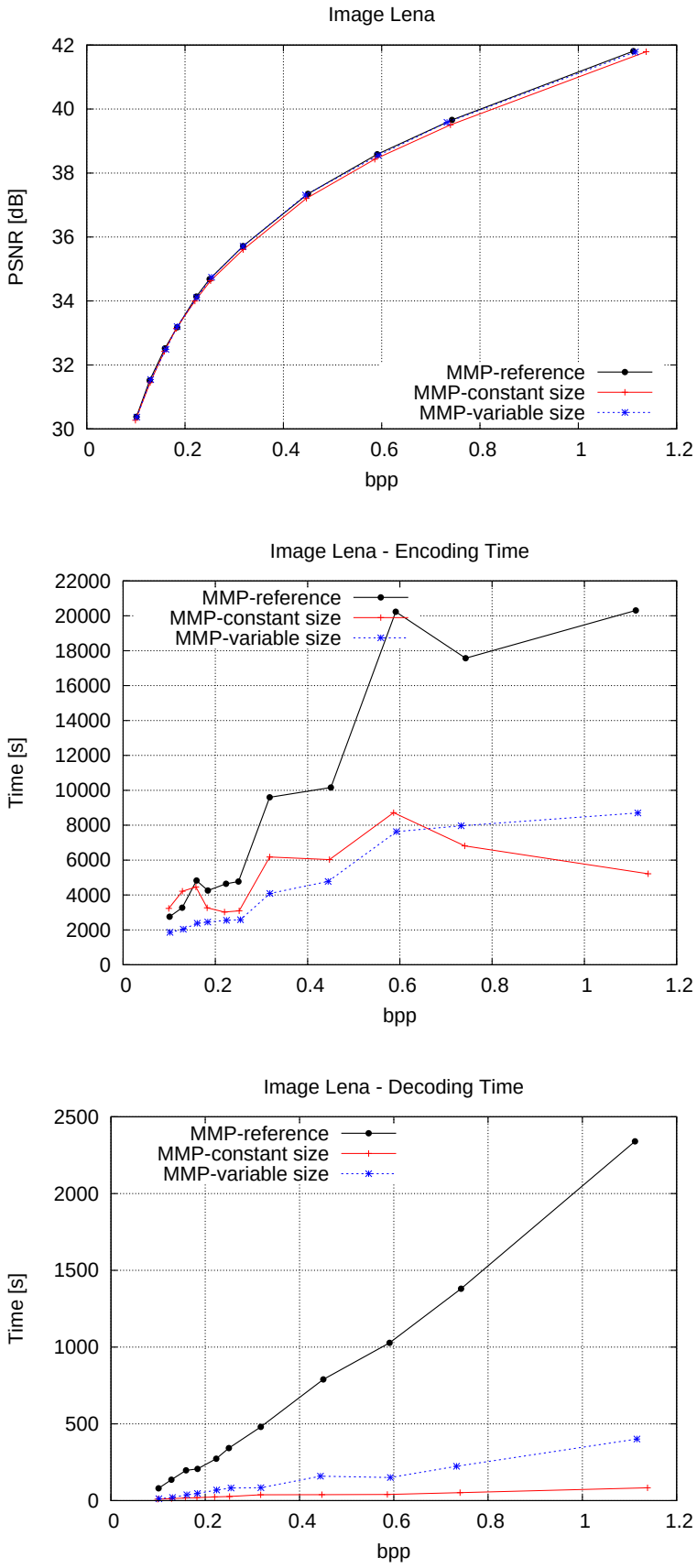
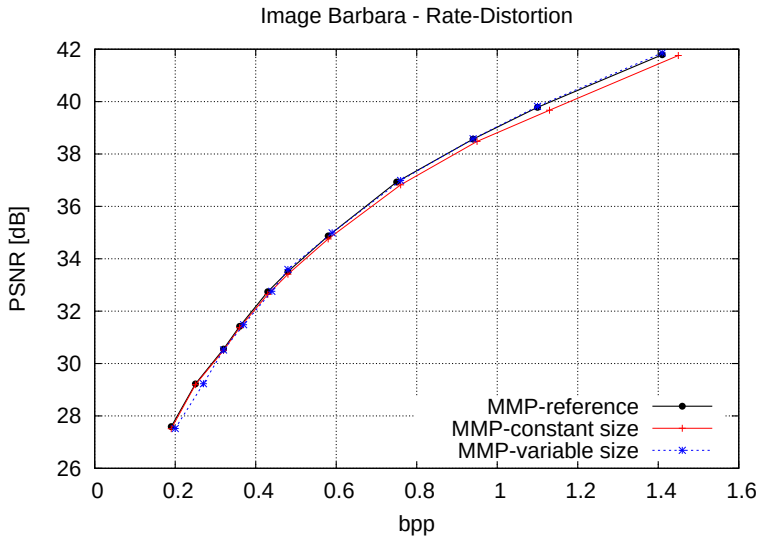
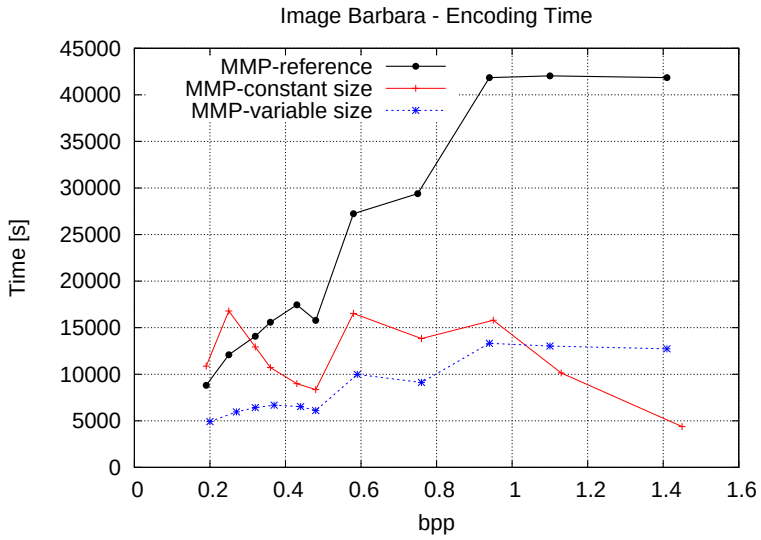


Figure E.8: Performance results for image Lena, using variable sized norm slots.





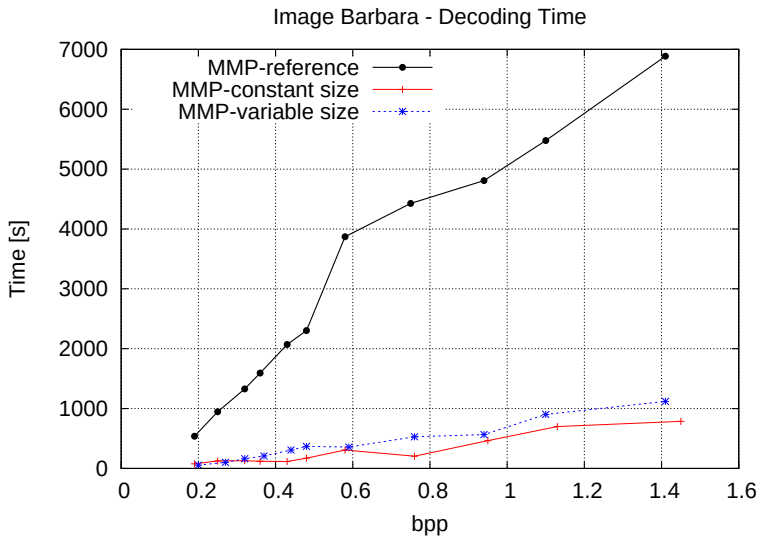


Figure E.9: Performance results for image Barbara, using variable sized norm slots.

total variation analysis. The total variation from each block is computed both in the vertical and horizontal direction:

$$\mathcal{G}^v = \sum_{i=1}^{N-1} |\mathcal{X}_{(i,j)} - \mathcal{X}_{(i-1,j)}|, \tag{E.6}$$

$$\mathcal{G}^h = \sum_{j=1}^{M-1} |\mathcal{X}_{(i,j)} - \mathcal{X}_{(i,j-1)}|, \tag{E.7}$$

and the block segmentation is interrupted in the corresponding direction, if the total variation on this direction is lower than a pre-established threshold τ. Intuitively, the dependency between τ and λ becomes obvious: large λs mean that the rate has a higher weight than the distortion, so even if large values of τ are defined, the algorithm will still be able to test a segmentation tree very similar to the one resulting from the exhaustive optimization. On the other hand, low λs mean that a low distortion is required, and the probability of segmenting any block to decrease the distortion is higher, even if it only provides a modest reduction on the distortion, so τ needs to be decreased.

A compromise between the rate-distortion performance and the computational complexity reduction can be determined depending on the need for computational complexity reduction. If the value of τ is set in a distortion conservative way, more segmentations will be allowed, and the computational complexity reduction decreases. If τ is set with a high value, segmentations will be restricted even in blocks with many details, resulting in computational savings but with possible losses of the rate distortion performance of the algorithm. In the limit, if a very large value is assigned to τ, the original blocks will never be segmented, and MMP converges to a traditional VQ algorithm.

Experimental tests were performed to establish a suitable relationship between τ and λ. The expression:

$$\tau_l = (0.001\lambda + 1.5) * size(l), \tag{E.8}$$

was found appropriate to describe this dependence, where $size(l)$ represents de block's number of pixels in the tested direction. A larger gradient variation is allowed on large blocks, as the possible distortion is distributed between more pixels, while only small variations are allowed on small blocks, to preserve details in regions of high activity.

Figure E.10 shows the mapping obtained using the proposed expression to compress image Lena using $\lambda = 5$. The darker pixels correspond to regions that can be segmented to lower levels, while lighter pixels correspond to regions where the algorithm limits the segmentation. We can see in this figure that the pre-processed map is able to efficiently identify uniform regions, where segmentations are unlikely to be used.

Figures E.11 and E.12 present the experimental results obtained using the expression from Equation E.8. Computational complexity reductions of up to 20% were achieved, while no performance losses were noticeable. Unlike the method presented in Sec-

(a) (b)

Figure E.10: a) Original image LENA 512×512 and b) obtained maximum segmentation map.

tion E.3.1, this method only has a considerable impact in the encoder computational complexity. The computational complexity of the decoder also decreases slightly, since the restriction of segmentations has the collateral effect of reducing the number of new codewords generated by MMP (note that the dictionary is primarily updated using concatenation of code-vectors, which result from the segmentations). However, the decrease on the decoder's computational complexity is not significant if the threshold τ is properly defined, as ideally, it would not affect the configuration of the segmentation tree obtained for a given block.

It is also important to notice that the use of this method results in an algorithm fully compliant with the previous versions of the MMP algorithm, since it only impacts the rate-distortion optimization decisions. Thus, there is no need to perform any modifications in the decoder.

E.4 Experimental results

In this section, we present the results achieved by the several described complexity reduction techniques, as well as their combination, when compared to a benchmark version of the MMP algorithm. The proposed methods are also compared with the method proposed in [85], which we will be further referred to as MMP Intra-fast.

Several images with different characteristics have been used on the experimental tests. The results for 4 representative images are presented in this section: smooth natural image Lena, natural image with high detail Barbara, scanned text image PP1205 and scanned compound document PP1209.

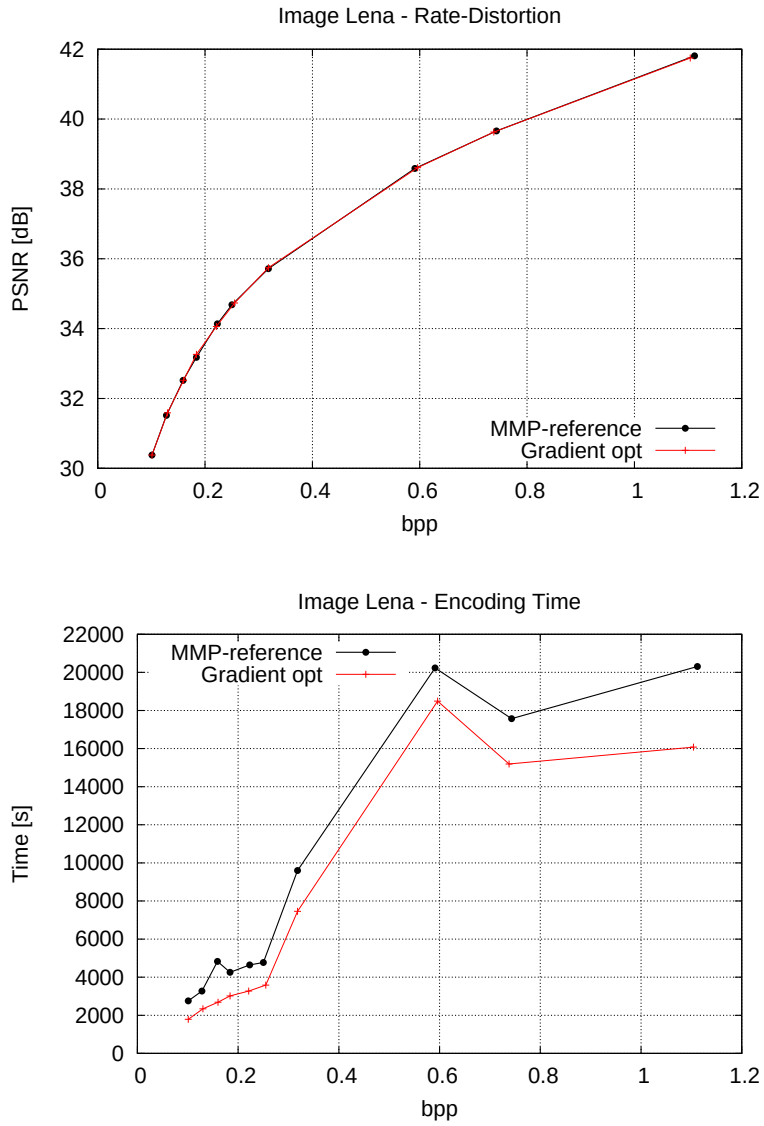


Figure E.11: Performance results of the gradient tree expansion for image Lena.

For the experimental tests, we defined the maximum dictionary capacity as 50000 codevectors for each dictionary scale, and used a 16×16 pixels initial block size. Memory structures were used on all encoders to calculate logarithms and squared values. The average of the codevectors was also used as a discarding parameter in the matching procedure. The redundancy control used the empirical rule defined on [49] and new code-vectors were only inserted on scales whose dimensions are half or twice each of the dimensions of the new originated block.

Table E.1 summarizes the time savings percentages achieved by the proposed methods. The resulting times are compared with the benchmark version of the MMP algorithm. The encoder that only uses adaptive capacity norm slots is referred as Enc. I, while Enc. II comprehends both the proposed techniques (adaptive capacity norm slots and gradient analysis). As the method proposed on Section E.3.2 only has considerable impact in the

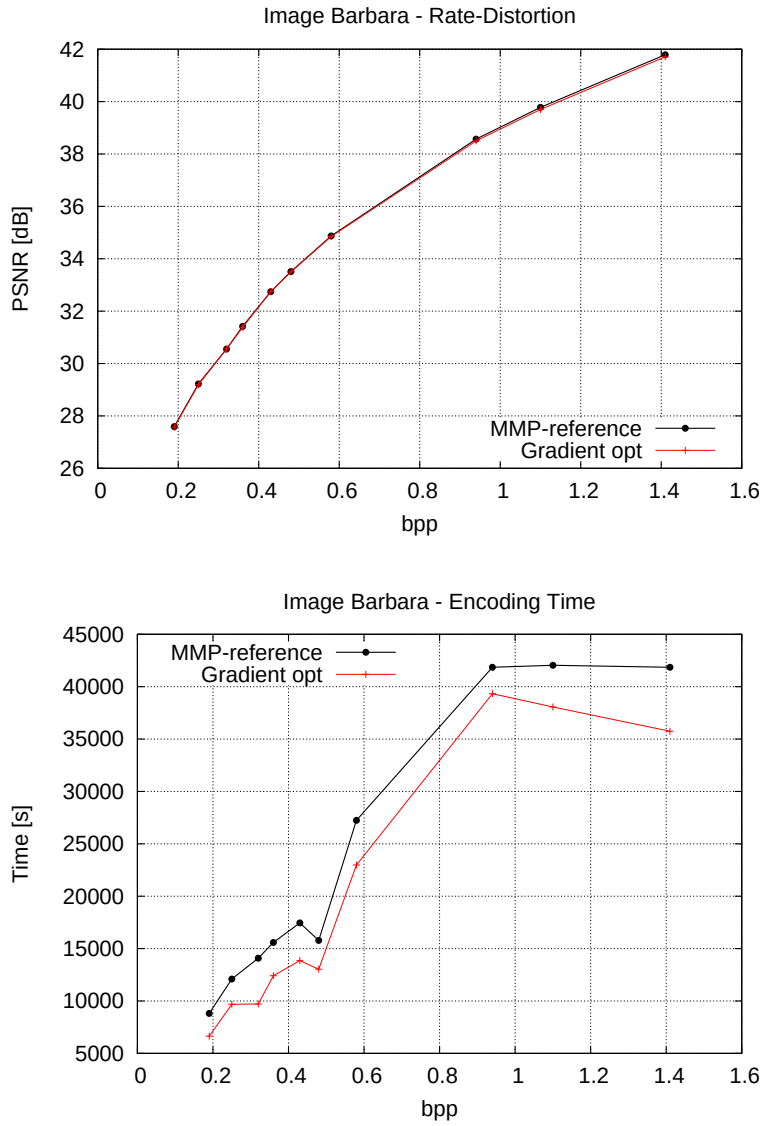


Figure E.12: Performance results of the gradient tree expansion for image Barbara.

optimization step and does not imposes any changes in the decoder, only the results for the final decoder are presented.

As can be seen in Table E.1, an average gain of 69% on the encoding time and 87% on the decoding time were achieved, while conjugating the two proposed techniques.

Figures E.13 to E.16 show the rate-distortion performance of the reduced complexity encoder when compared with the original version and with two state-of-the-art transform based encoders: H.264/AVC and JPEG2000. Despite the significant computational complexity reduction, the proposed method only presents residual R-D performance losses for smooth images. For text and compound images, the statistical distribution of residue norms tends to vary, since the efficiency of the prediction stage decreases. For this reason, the losses on the rate-distortion performance slightly increase, to up to 0.2 dB, in the worst case. However, these cases also achieve a greater reduction in computational com-

Table E.1: Percentage of time saved by the proposed methods over the reference codec.

	Rate	0.25bpp	0.50bpp	0.75bpp	1.00bpp	Average
Enc. I	Lena	46%	53%	55%	57%	53%
	Barbara	51%	61%	69%	69%	63%
	PP1205	63%	72%	79%	84%	75%
	PP1209	50%	65%	66%	65%	62%
	Average	53%	63%	67%	69%	63%
Enc. II	Lena	56%	59%	63%	65%	61%
	Barbara	59%	65%	71%	72%	67%
	PP1205	73%	78%	82%	87%	80%
	PP1209	60%	68%	68%	70%	67%
	Average	62%	68%	71%	74%	69%
Decoder	Lena	73%	80%	84%	84%	80%
	Barbara	87%	81%	85%	87%	85%
	PP1205	94%	94%	92%	94%	94%
	PP1209	91%	87%	89%	90%	89%
	Average	86%	86%	88%	89%	87%

plexity. Furthermore, the proposed encoder still considerably outperforms state-of-the-art transform based compression algorithms.

The comparison with the MMP Intra-fast method proposed in [85], is presented on Table E.2. Both the results for the encoder and decoder of each method are displayed, relatively to the benchmark version of the MMP algorithm.

The proposed method is able to achieve an encoder's computational complexity reduction close to that from the MMP Intra-fast, with a considerably lower degradation on the rate-distortion performance. The rate-distortion performance losses of Intra-fast are up to 1 dB, while the losses for the proposed methods do not exceed 0.2 dB.

The major advantage of the proposed method is the reduction on the decoder's complexity, a tendency that is not verified on the Intra-fast method. Furthermore, the computational complexity of the Intra-fast algorithm increased relatively to the benchmark version, a very undesirable effect as the decoder's computational complexity is the major issue for its practical applications on encode-once-decode-many scenarios.

This fact has a simple explanation: more codewords are created due to the sub-optimal choice of the block's prediction mode, which increases the average residual energy. As a consequence, more segmentations are performed while encoding the residual data, resulting in an increase on the final dictionary's size. Thus, both the encoder and decoder will need to perform more searches for similar blocks, while inserting new codevectors on the dictionary and this increases the time required for these searches. This time is diluted in the encoders' complexity gains, but is very relevant on the decoder's side, as searches for existing codewords in the dictionary updating stage correspond to most of the decoder's

Table E.2: Percentage of time saved by the proposed methods and by the Intra-fast method, over the reference codec.

		Rate	0.25bpp	0.50bpp	0.75bpp	1.00bpp	Average
Proposed	**Encoder**	Lena	56%	59%	63%	65%	61%
		Barbara	59%	65%	71%	72%	67%
		PP1205	73%	78%	82%	87%	80%
		PP1209	60%	68%	68%	70%	67%
		Average	62%	68%	71%	74%	69%
	Decoder	Lena	73%	80%	84%	84%	80%
		Barbara	87%	81%	85%	87%	85%
		PP1205	94%	94%	92%	94%	94%
		PP1209	91%	87%	89%	90%	89%
		Average	86%	86%	88%	89%	87%
Method from [85]	**Encoder**	Lena	69%	73%	80%	75%	74%
		Barbara	71%	62%	68%	67%	67%
		PP1205	68%	74%	76%	71%	72%
		PP1209	72%	72%	76%	77%	74%
		Average	70%	70%	75%	73%	72%
	Decoder	Lena	5%	-3%	-13%	-3%	-4%
		Barbara	-11%	-9%	-15%	-6%	-10%
		PP1205	-4%	-7%	3%	8%	0%
		PP1209	-14%	-4%	-2%	-3%	-4%
		Average	-6%	-6%	-7%	-1%	-5%

computational complexity.

The rate-distortion performance results of the Intra-fast method are also included on Figures E.13 to E.16, in order to allow a direct comparison with other methods.

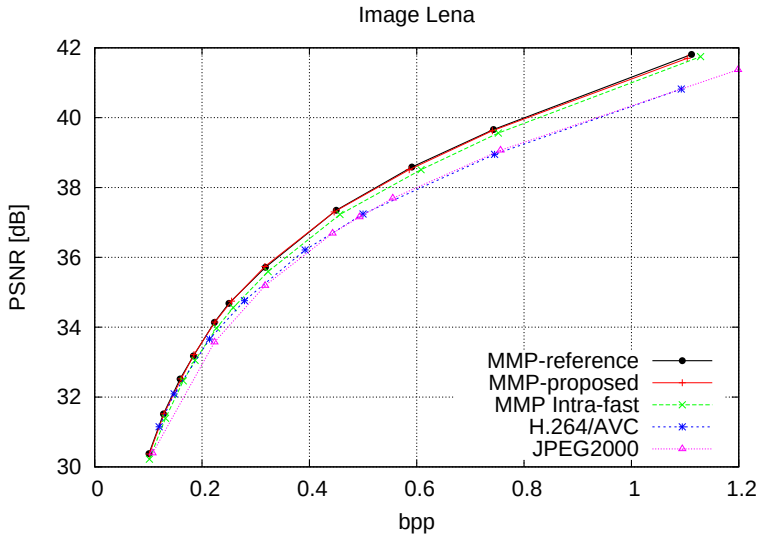


Figure E.13: Experimental results for image LENA 512×512.

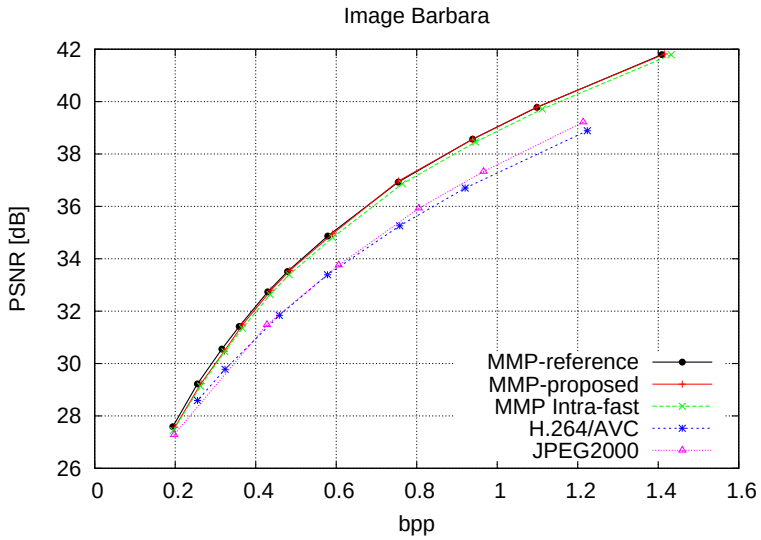


Figure E.14: Experimental results for image BARBARA 512×512.

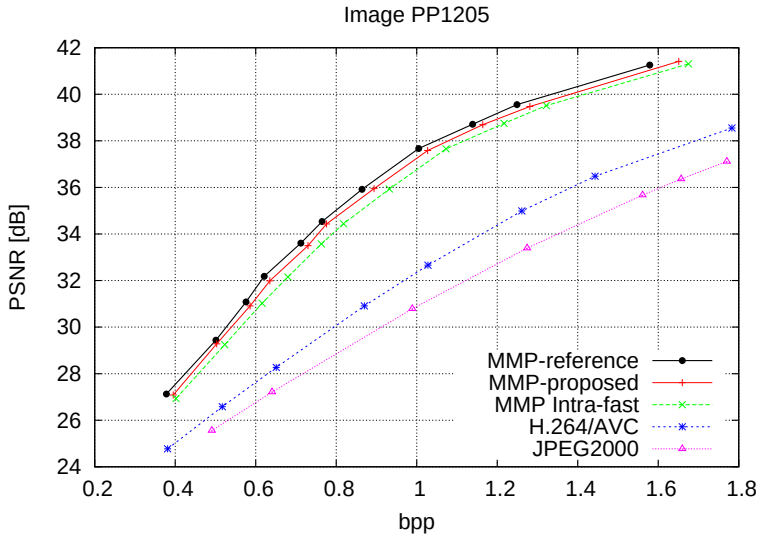


Figure E.15: Experimental results for image PP1205 512×512.

To conclude the study, we analyzed the combined impact of the proposed methods with the MMP Intra-fast scheme. As both methods attempt to overcome the computational complexity issue by a different angle, it is possible to exploit their combined effect to obtain a faster algorithm. However, rate-distortion losses are expected, mostly due to the sub-optimal prediction choice performed by the Intra-fast method.

The results from such algorithm are summarized on Table E.3, and the rate-distortion performance is shown on Figures E.17 to E.20, when compared to the benchmark algorithm, JPEG2000 and the H.264/AVC Intra coder.

As can be seen in Table E.3, an average reduction of 90% and 87% was achieved respectively for the encoder's and decoder's computational complexity, while combining both methods. This means that the encoder is able to compress and decompress an image

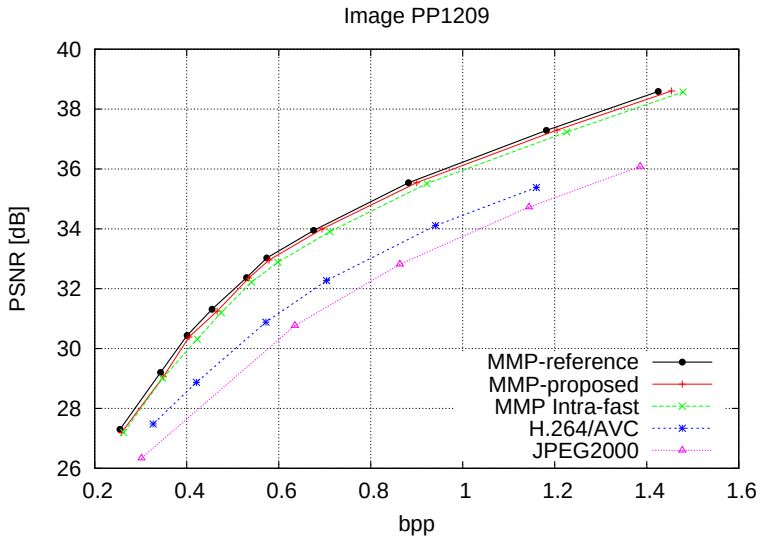


Figure E.16: Experimental results for image PP1209 512×512.

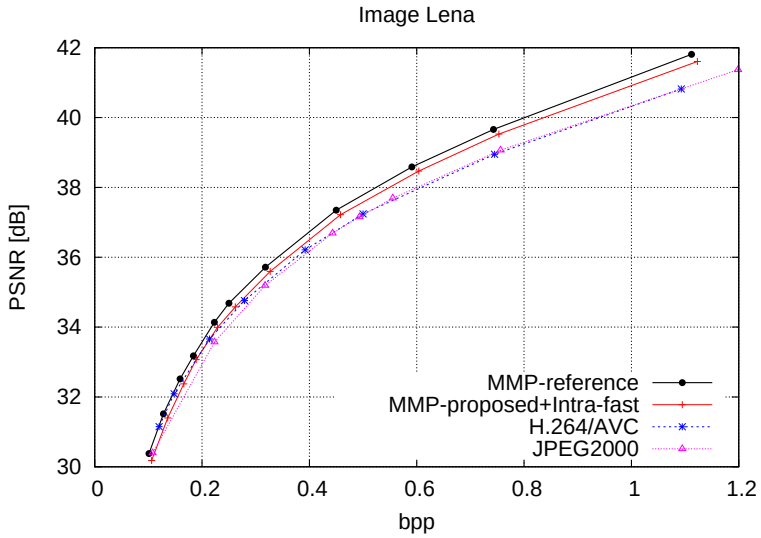


Figure E.17: Experimental results for image LENA 512×512.

in a tenth of the original time. As a reference, encoding the image LENA at 0.15 bpp in a I7 at 3GHz processor, originally required 3270 seconds and is now encoded in 700 seconds with the new framework. For the case of the decoder, the time required to decode the same image reduced from 136 seconds to 24 seconds.

The MMP's computational complexity still remains considerably higher than that of transform-based encoders, but the proposed methods allowed to considerably reduce the encoding and decoding times.

From Figures E.17 to E.20, we can see that despite a rate-distortion performance loss of up to 1 dB in the worst case, the final algorithm still considerably outperforms state-of-the-art transform-based encoders for all tested images.

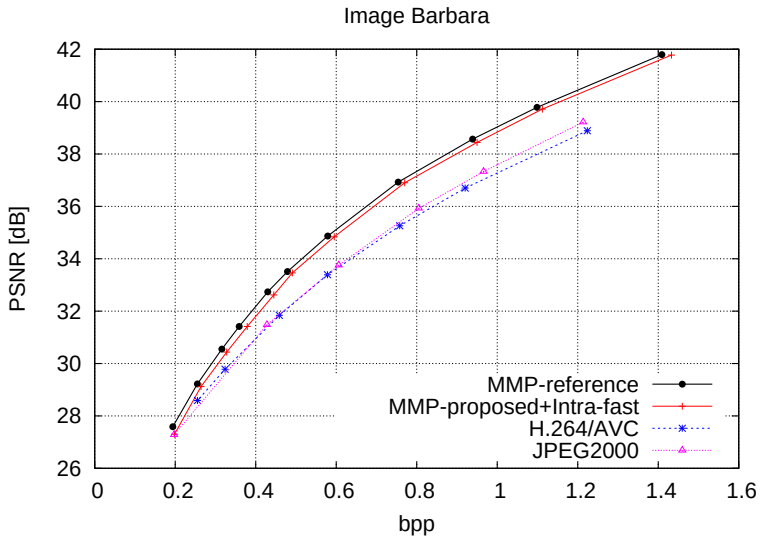


Figure E.18: Experimental results for image BARBARA 512×512.

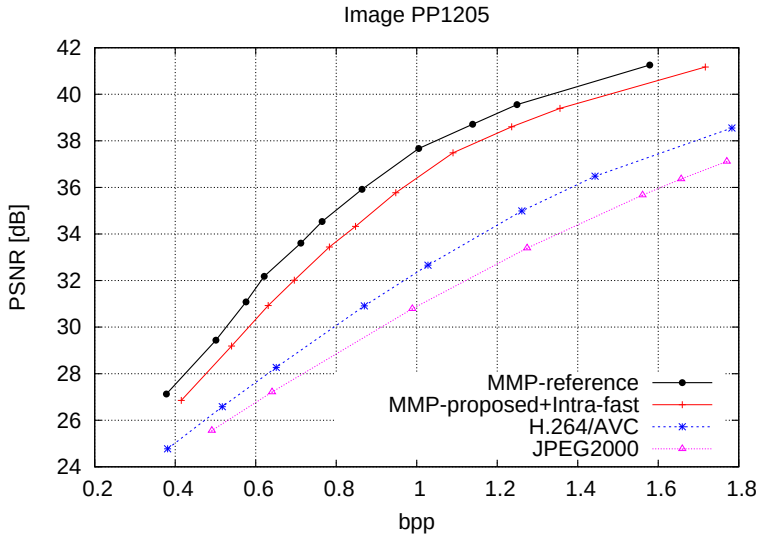


Figure E.19: Experimental results for image PP1205 512×512.

E.5 Conclusions

In this appendix, we have presented two computational complexity reduction techniques specially developed for the MMP algorithm, but that can be adapted to other pattern matching methods. These techniques considerably reduce the MMP's computational complexity, with only marginal rate-distortion performance losses.

The combination of the proposed methods with previously proposed computational complexity reduction techniques, further increased the average time savings to about 90% both on the encoder and decoder.

MMP's rate-distortion performance advantage, for a wide range of applications, makes the convergence between its encoding time and those from transform-based algorithms an important factor in affirming the pattern matching paradigm as a viable alterna-

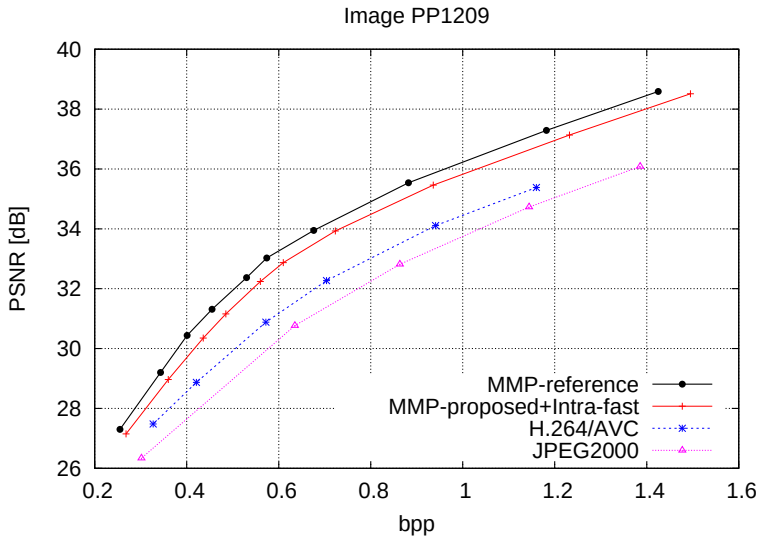


Figure E.20: Experimental results for image PP1209 512×512.

Table E.3: Percentage of time saved by the combined methods over the reference codec.

	Rate	0.25bpp	0.50bpp	0.75bpp	1.00bpp	Average
Final	Lena	82%	87%	90%	90%	87%
	Barbara	87%	88%	90%	91%	89%
	PP1205	92%	94%	94%	95%	94%
	PP1209	87%	89%	91%	91%	90%
	Average	87%	90%	89%	92%	90%
Decoder	Lena	71%	80%	83%	82%	79%
	Barbara	90%	84%	86%	87%	87%
	PP1205	93%	93%	91%	93%	93%
	PP1209	89%	86%	89%	89%	88%
	Average	86%	86%	87%	88%	87%

tive to the actual paradigm. Despite being still considerably more computationally complex, an important step was achieved in that direction, especially if we consider "encode once and decoded many times" application scenarios, where the encoder's high computational complexity can be easily justified by a state-of-the art rate-distortion performance.

Several improvements can still be achieved in the future, namely by exploiting the existence of repetitive tasks, like multiscale dictionary searches, that may be parallelized, and the intensive use of integer operations. Nevertheless, these advances are implementation related, while the methods proposed in this appendix are algorithm design techniques. Another interesting area of research are multi-core based processing, either through the use of GPUs or general purpose multi-core processors. These systems have enjoyed a recent increase in popularity, and may be used for future optimized implementations.

Appendix F

A generic post deblocking filter for block based algorithms

F.1 Introduction

Such as several image and video compression standards, from JPEG [53] to H.264/AVC [45], MMP can be classified as a block-based encoder. The approach used by such encoders is to partition the input image into non-overlapping blocks which are sequentially processed using transform coding, quadtree decomposition, vector quantization or other compression techniques.

Despite the high compression efficiency achieved by some of these algorithms, the visual quality of the compressed images is often affected by blocking artifacts, resulting from the discontinuities induced in the block boundaries, specially at high compression ratios.

Several approaches have been proposed in the literature in order to attenuate these artifacts, such as adaptive spatial filtering [86, 87, 125], wavelet-based filtering [88], transform-domain methods [89, 90] or interactive methods [91], just to name a few. In [94], a blocking artifact reduction method specifically oriented towards MMP compressed images was presented.

Some of these deblocking techniques have been developed to work as in loop-filters, such as [78], the deblocking filter adopted by the standard H.264/AVC [45]. However, the use of loop filters requires that every compliant decoder must replicate the filtering procedure, in order to stay synchronized with the encoder. This can be an inconvenient, as a decoder would loose the flexibility to switch off the deblocking filter in order to trade-off visual quality for computational complexity, if needed. Post-deblocking methods have been proposed to overcome this drawback [92, 93]. In this case, the filtering procedure is only performed after the decoding process is finished, thus not interfering with the encoder/decoder synchronization. Traditionally, the post-processing strategies tend to be

less efficient than the in-loop filters, as they are not able to exploit all the information available in both the encoding and decoding process that helps to locate blocking artifacts and avoid filtering unwanted regions.

For the upcoming HEVC coding standard [16], a new filter architecture [126] was proposed, combining an in-loop deblocking filter and a post-processing Wiener filter. The in-loop filter reduces the blocking artifacts, while the Wiener filter is a well-known linear filter which can guarantee the objective quality optimized restoration of images degraded during the compression process by gaussian noise, blurring or distortion. A unified architecture for both filters is proposed in [126], but it results again in an in-loop filter that, despite its high efficiency, is still not able to present the advantages of post-deblocking methods.

In order to overcome some inefficiencies presented by the method proposed in [94], some research work was conducted targeting the increase of the perceptual quality of MMP compressed images. The result of such investigation is a new versatile post deblocking filter, which is not only able to achieve significant performance gains over the method described in [94], when applied to MMP compressed images, but also to achieve a performance comparable to the ones of state-of-the-art in-loop methods, when used in still images and video sequences compressed with several other block-based compression algorithms, such as H.264/AVC [45], JPEG [53] and the upcoming standard HEVC [16].

The new method is described in this appendix, and evaluated for images and video sequences encoded with JPEG, H.264/AVC, MMP and HEVC. The appendix is organized as follows: in Section F.2 we present some related work that motivated the development of the proposed method; Section F.3 describes the new algorithm used for mapping and classifying the blocking artifacts, as well as the adaptive filter used to reduce those artifacts. Experimental results are shown in Section F.4, and Section F.5 concludes this appendix.

F.2 Related work

The development of the proposed post-processing deblocking algorithm was motivated by the use of the Multidimensional Multiscale Parser algorithm (MMP) [3]. The use of variable sized patterns in MMP restricts the use of most existing deblocking methods, as they were developed to be employed with fixed size block-based algorithms, such as JPEG [53] and H.264/AVC [45]. In such cases, the location of the blocking artifacts is highly correlated with the border areas of the transformed blocks, and consequently depends mostly on the block dimensions. This extra information is exploited by some deblocking methods, such as the ones in [78], [93] or [127]. It is usually employed to help classifiers to locate regions that need to be deblocked, avoiding the risk of filtering unwanted regions. Unlike these algorithms, the multiscale matching used in MMP may introduce artifacts at any location along the reconstructed image.

A similar situation occurs on motion compensated frames from encoded video sequences. Although the location of blocking artifacts on the Intra-coded slices is predictable, motion compensation may replicate these artifacts to any location on Inter-coded slices if no deblocking method is applied before performing the motion compensation. As a result, post-processing deblocking methods for Inter-coded frames should be able of efficiently locate blocking artifacts away from block boundaries. The method proposed in [92] addresses this issue by using the motion vector information in order to identify the regions of the motion compensated slices which used possibly blocked locations of the reference slices. Therefore, it cannot be considered a pure post-deblocking technique, as in addition to the decoded video, it needs information provided by the decoded bitstream. As a consequence, this technique is specific for H.264/AVC, and will not work for other algorithms that use a different encoding scheme.

In [94], a bilateral adaptive filter was proposed for the MMP algorithm, which achieved satisfactory results when used in a non-predictive coding scheme. However, that method showed considerable limitations when used with predictive MMP-based algorithms, which present state-of-the-art results for natural image coding [49], as referred in Appendix B. Additionally, this method is also algorithm specific, since it needs information provided by the MMP bitstream.

Based on the above, we see that both block-based video encoders and MMP with a predictive scheme would benefit from a versatile post-deblocking method. In the following sections, we describe such method.

F.3 The deblocking filter

In this section, we describe the proposed deblocking method. It is based on the use of a space-variant finite impulse response (FIR) filter, with an adaptive number of coefficients. Prior to the filtering stage, the input image is analyzed, in order to define the strength of the filter to apply to each image region. This results in a filtering map, which indicates the length of the filter's support that will be applied to each pixel or block in the image. High activity regions will have a shorter support length associated, while smooth areas will use a longer filter support, in order to provide a higher blocking artifact reduction.

F.3.1 Adaptive deblocking filtering for MMP

As shown in Appendix B, the RD control algorithm used in MMP only considers the distortion and the rate of the encoded data, without taking into account the block borders' continuity, which is the main source of blocking artifacts. As blocks at different scales are concatenated, these artifacts are likely to appear in any location of the reconstructed image, unlike the case of transform based methods, where blocking artifacts only arise in

predetermined locations, along a grid defined by the size of the block transform.

Let us define an image reconstructed with MMP, $\hat{\mathbf{X}}$, as:

$$\hat{\mathbf{X}}(x,y) = \sum_{k=0}^{K-1} \hat{\mathbf{X}}_k^{l_k}(x - x_k, y - y_k), \tag{F.1}$$

i.e., the concatenation of K non-overlapping blocks of scale l_k, $\hat{\mathbf{X}}_k^{l_k}$, each one located on position (x_k, y_k) of the reconstructed image. One can notice that each block $\hat{\mathbf{X}}_k^{l_k}$ also results from previous concatenations of J other elementary blocks, through the dictionary update process. Defining these elementary blocks as $\mathcal{D}_{0_j}^{l_j}$, where l_j represents the original scale and (u_j, v_j) represent the position of the elementary block inside $\hat{\mathbf{X}}_k^{l_k}$, we obtain:

$$\hat{\mathbf{X}}_k^{l_k}(x,y) = \sum_{j=0}^{J-1} \mathcal{D}_{0_j}^{l_j}(x - u_j, y - v_j). \tag{F.2}$$

In this equation, one may identify the border regions of each pair of adjacent basic blocks, which correspond to the most probable location for discontinuities in the decoded image, that may introduce blocking artifacts.

In [94], a deblocking method was proposed for the MMP algorithm, that stores the information regarding the original scale of each elementary block that composes each codeword, in order to locate all the existing boundaries. These boundaries correspond to the regions where the deblocking artifacts are most likely to appear, and this information is used to generate a map that defines the length of the filter's support for each region. This is done by imposing that blocks, $\mathcal{D}_{0_j}^{l_j}$, at larger scales, which generally correspond to smoother areas of the image, should be filtered more aggressively, while blocks with small values of the scale l_j, corresponding to more detailed image areas, should not be subjected to strong filtering, in order to preserve the image's details.

A space-variant filter is then used for the deblocking process. The control of the support's length adjusts the filter's strength, according to the detail level of the region being deblocked. This avoids the appearance of blurring artifacts, which are frequently caused by the use of deblocking techniques. Figure F.1 presents a one-dimensional representation of a reconstructed portion of the image, resulting from the concatenation of three basic blocks, $(\mathcal{D}_0^{l_0} \quad \mathcal{D}_1^{l_1} \quad \mathcal{D}_2^{l_2})$, each from a different scale: l_0, l_1 and l_2, respectively. At each filtered pixel, represented in the figure by a vertical arrow, the kernel support of the deblocking filter is set according to the scale l_k used for its representation.

This method proved to be successful in several cases. Nevertheless, some problems were observed when it was used in a predictive-based coding scheme. Accurate predictions result in low energy residues even for regions presenting high activity, that tend to be efficiently coded using blocks from larger scales. As a result, some highly detailed regions would be improperly considered as smooth and filtered with a large aggressive filter.

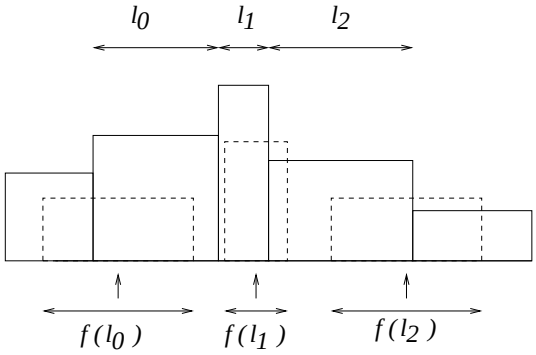


Figure F.1: The deblocking process employs an adaptive support for the FIR filters used in the deblocking.

This may introduce a considerable degradation on the image's detail, forcing the decrease of the overall strength of the filter (one single strength is used for all image), and thus, limiting its effectiveness. Also, the tracking of the information about the original scale of the basic units that compose each codeword is also a cumbersome task. Furthermore, perhaps the most important disadvantage of this method is that it is only appropriate for the MMP algorithm, since it needs segmentation information obtainable from the MMP decoding process.

F.3.2 Generalization to other image encoders

In order to overcome the limitations described for the method from [94], we propose a new mapping approach based on a total variation analysis of the reconstructed image. The new mapping procedure starts by considering that the image was initially segmented into blocks of $N \times M$ pixels. For each block, the total variation of each of its rows and columns is determined, respectively, by:

$$\mathcal{A}_j^v = \sum_{i=1}^{N-1} |\hat{\mathbf{X}}_{(i+1,j)} - \hat{\mathbf{X}}_{(i,j)}|, \tag{F.3}$$

$$\mathcal{A}_i^h = \sum_{j=1}^{M-1} |\hat{\mathbf{X}}_{(i,j+1)} - \hat{\mathbf{X}}_{(i,j)}|. \tag{F.4}$$

Each region is vertically or horizontally segmented if any of the above quantities exceeds a given threshold τ. With this approach, regions presenting a high activity are successively segmented, resulting in small areas that will correspond to narrower filter supports. In contrast, smooth regions will not be segmented, which will correspond to wider filter supports, associated to larger blocks.

It is important to notice that the value of τ has a limited impact on the final performance of the deblocking algorithm. A high value for τ results in fewer segmentations, and consequently, on a larger filter's supports than those obtained using a smaller value

for τ. However, these larger supports can be compensated adjusting the shape of the filter, in order to reduce the weight, or even neglect, the impact of distant samples on the filter support. In other words, the value of τ can be fixed, as this procedure only need to establish a comparative classification of the regions with different variation intensity, with the deblocking strength being controlled through the shape of the filter used.

Figure F.2 shows the filtering map generated for image Lena coded with MMP at two different bitrates, using $\tau = 32$. Lighter areas correspond to regions that will use larger filter supports, while darker regions correspond to regions that will use narrower filter supports. It is important to notice that not only the proposed algorithm was effective in capturing the image structure for both cases, but also it revealed an intrinsic ability to adapt to the different compression ratios. The map for the image coded at a lower bitrate has a lighter tone, that corresponds, on average, to wider supports for the deblocking filter. This is so because as the reconstruction is heavily quantized and larger blocks tend to be used, the sum of the gradient tends to be low in these regions, corresponding to the need for strong filtering.

Figure F.2: Image Lena 512×512 coded with MMP at 0.128bpp (top) and 1.125bpp (bottom), with the respective generated filter support maps using $\tau = 32$.

It is also important to notice that this approach is based on the information present in the reconstructed image only, and is thus independent of the encoding algorithm used to generate it. As a result, the proposed method overcomes the problem of misclassification of well predicted detailed regions when predictive coding is used.

Furthermore, when applied to MMP, it avoids the need for keeping a record of all original scales of the basic units used for each block performed, as was done in [94], resulting in a more effective and less cumbersome algorithm. Note that one of the advantages of the new scale analysis scheme is that it enables the use of the adaptive deblocking method

for images encoded with any algorithm.

F.3.3 Adapting shape and support for the deblocking kernel

For the algorithms proposed in [94], several kernels with various support lengths were tested. In [94], experimental results showed that gaussian kernels are effective in increasing the PSNR value of the decoded image, as well as in reducing the blocking artifacts. Thus, Gaussian kernels were also adopted for the proposed method, with the same $l_k + 1$ samples filter length, where l_k refers the segment support. The filter strength is then controlled by adjusting the gaussian's variance, producing filter kernels with different shapes. Considering a gaussian filter, with variance $\sigma^2 = \alpha L$ and length L, we can express its impulse response (IR) as:

$$g_L(n) = e^{-\frac{\left(n-\frac{L-1}{2}\right)^2}{2(\alpha L)^2}}, \tag{F.5}$$

with $n = 0, 1, ..., L-1$. By varying the parameter α, one adapts the IR of the filter by adjusting the variance of the gaussian function. The IR of the filter may range from almost rectangular to highly peaked gaussians, for different lengths. Furthermore, when α tends to zero, the filter's IR tends to a single impulse, and the deblocking effect is switched off for those cases where filtering is not beneficial.

Figure F.3 represents the shape of a 17 tap filter for the several values of parameter α.

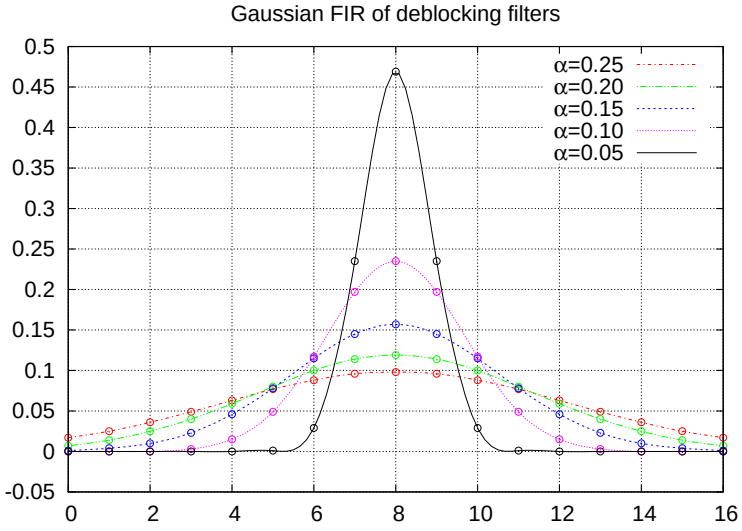


Figure F.3: Adaptive FIR of the filters used in the deblocking.

The analysis of deblocked images revealed that artifacts appeared on some regions, where a concatenation of wide and short blocks with very different intensity values occurred (see Figure F.4). Here a wide dark block A is concatenated with two bright blocks: one narrow block B followed by one wide block C. When blocks A and B are filtered, a smooth transition appears, which eliminates the blocking effect in the AB border. When

the block C is filtered, the pixels near the BC border will suffer from the influence of some of the dark pixels of block A, resulting in the appearance of a dark "valley" in the BC border.

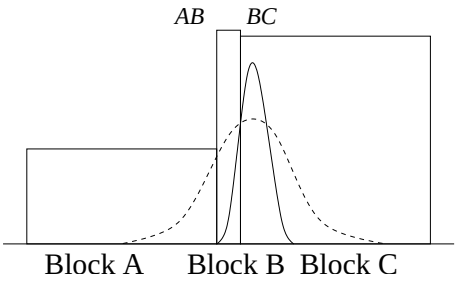


Figure F.4: A case were the concatenation of blocks with different supports and pixel intensities causes the appearance of an image artifact, after the deblocking filtering.

This problem was solved by constraining the filter's support to the pixels of the present block and to those from its adjacent neighbors. In the example of Figure F.4, the length of the filter applied to C block's pixels that are near to the BC border is reduced, so that the left most pixel of the support is always a pixel from block B. This corresponds to use the filter represented by the solid line, instead of the original represented by the dashed line.

Figure F.5 illustrates another common situation, which also results on the introducion of some artifacts in the original method. When two blocks A and B with very different intensity values are concatenated, it is highly probable that the border between the two blocks corresponds to a natural edge. In order to avoid these natural edges to be filtered, a feature similar to that used by the H.264/AVC adaptive deblocking filter [78] was also adopted. The differences in the borders are monitored, and the filter is switched off every time this difference exceeds a defined step intensity threshold, s. For the case represented on Figure F.5, the filter is switched off if $|A_k - B_0| > s$.

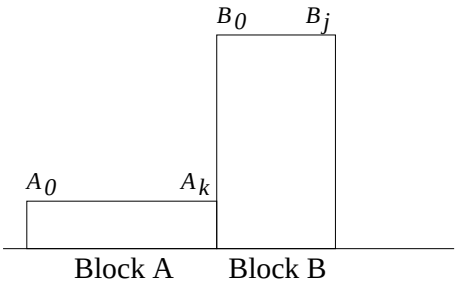


Figure F.5: A case where a steep variation in pixel intensities is a feature of the original image.

F.3.4 Selection of the filtering parameters

The filtering parameters α (gaussian filter variance), s (step intensity threshold at image edges) and τ (segmentation threshold) must be known in the decoder to perform the de-

blocking task. In [94], the parameters' values were exhaustively optimized at the encoder side in order to maximize the objective quality, and appended at the end of the encoded bit-stream. This introduced a marginal additional computational complexity and a negligible overhead, but changed the structure of the encoded bit-stream. Consequently, this approach restricts the use of the deblocking process on standard decoders, such as JPEG and H.264/AVC, that have normalized bitstream formats.

In order to address this problem, we have developed a version of the proposed deblocking method that avoids the transmission of the filter parameters, by estimating their values at the decoder side. The parameter estimation is supported by the high correlation observed between the amount of blocking artifacts and some of the statistical characteristics presented by the encoded image.

The relation between τ and the shape of the used filter was already mentioned on Section F.3.2. For the case of gaussian kernels, the use of a large value for τ, which results on larger filter supports, can be compensated using a lower value for α. This corresponds to a highly peaked gaussian, that results on a filtering effect similar to the one obtained using a shorter support and a larger value for α. For that reason, the value of τ can be fixed without significant losses on the method's performance, with the deblocking effect being exclusively controlled by adjusting the parameter α.

Experimental tests have shown that the performance of the algorithm is considerably more affected by α, than by the step intensity threshold s. Thus, we started by studying the relationship between the optimal α and the statistical characteristics of the input images. The parameter s would then be responsible for the method's fine tuning.

Fixing the parameters $\tau = 32$ and $s = 100$, a large set of test images was post-processed with the proposed method, using several values of α, in order to determine the one which maximizes the PSNR gain for each case. The test set included a large number of images with different characteristics, compressed at a wide range of compression ratios and using different coding algorithms, including MMP, H.264/AVC (compressed as still images and video sequences) and JPEG. Thus, it was possible to evaluate the behavior of the proposed method for a wide range of applications.

For each case, several statistical characteristics of each image were simultaneously calculated, in order to determine their correlation with the optimal value of α. This analysis included:

- The average support size that resulted from the mapping procedure;
- The standard deviation of the distribution of the support lengths;
- The average variation between neighbor pixels;
- The standard deviation of the distribution of the variation between neighbor pixels.

We observed that the optimal value of α increases with the average of the filter's support length and decreases with the value of the average variation between neighbor pixels, as expected.

Images which tend to use large filter supports, have usually a low pass nature and can thus be subjected to aggressive filtering without significant degradation on its overall quality. In the other hand, images which result on narrow filter supports, tend to present highly detailed regions and must be subjected to moderate filtering, in order to not degrade these detailed regions. The same tendency is verified for the average variation between neighbor pixels. However, the average support length presented better results, hence it reflects not only the amount of details on the image, but also includes some information regarding the way these details are distributed on the image.

The standard deviation of the variations was used to characterize the distribution of details across the image. For a given average support length (or variation between neighbor pixels), if the distribution of the pixels' variation presents a high standard deviation, one may assume that the details are more concentrated on limited regions of the input image, than for distributions presenting low standard deviations. In these cases, one may assume that details are homogenously spread across the entire image.

We found the average support length to be a simple and effective estimator for the optimal value of α, by itself. In Figure F.6a, we present a plot which represents the optimal value of α in function of the average support length. The presented points were obtained for a test of 1000 test images, encoded using different codecs. The proposed method was exhaustively applied to each image, using values for α ranging from 0 to 0.25, with a 0.01 step increment. The value which maximized the PSNR of the post-processed image is then plotted as a function of the calculated average support length.

Figure F.6a clearly demonstrates the correlation between the product of the average support lengths in both the vertical and horizontal directions, and the optimal value determined for α. The optimal value for α presents a tendency to increase for images which result in larger supports. Thus, we may approximate the optimal value of α through a linear function which minimize the estimation error for the entire test set, without incurring in significant errors. However, we adopted a conservative approach while defining the linear function, as the reconstructed image's quality is more negatively affected if the α estimated exceeds the optimal one than for the opposite. If the calculated value considerably exceeds the optimal α, the filtering process may degrade the image details. On the other hand, if the value determined for α is lower than the optimal one, there are no risks of degrading the image's quality, and the unique issue is a quality improvement not as significant as it could be. We found the equation:

$$\alpha = 0.0035 \times v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}, \tag{F.6}$$

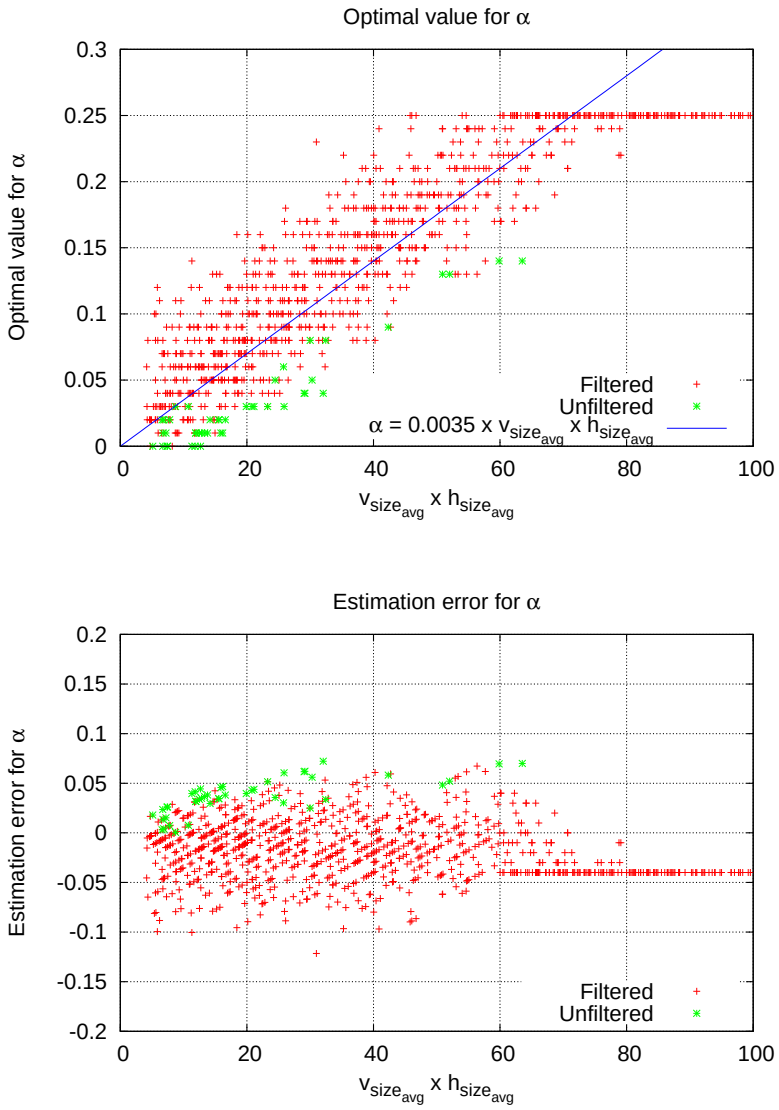


Figure F.6: Best value for α *vs.* the product of the average support lengths both in the horizontal and vertical directions.

where $v_{\text{size}_{\text{avg}}}$ and $h_{\text{size}_{\text{avg}}}$ represent the average length of the filter's support obtained by the mapping procedure, in the vertical and horizontal directions, respectively, to present a good fit to the distribution. The function resulting from F.6 is plotted in Figure F.6a. We have observed that, in general, the product of the average length on both directions allows to obtain better results than the use of a separate optimization for each direction. The combination of features from both directions makes the algorithm less sensitive to cases where the image characteristics tend to differ from the adopted model. In order to avoid excessive blurring on the reconstructed images, the maximum value of α was limited to $\alpha = 0.21$.

Figure F.6b represents the error between the estimated value for α and that obtained through the exhaustive optimization process, plotted in function of the product of the average support length. One may notice that the error is tendentiously negative, for the

reasons mentioned above.

Despite the good results observed for most of the tested images, the model presented some limitations when applied to scanned text images, or, more generally, images presenting a large amount of high frequency details very concentrated in some regions. The short supports associated to detailed regions were in these cases counterbalanced by the long supports from the background regions, and Equation F.6 resulted in too aggressive filtering for these particular cases. Thus, it has been found advantageous to switch off the filter when the standard deviation of the distribution of variations between neighbor pixels (σ_{size_v} and σ_{size_h}) in each direction, is high. We also found appropriate to switch-off the filter when the product of the standard deviations exceeds the product of the average support lengths by a given threshold, that is, when:

$$\frac{\sigma_{\text{size}_v} \times \sigma_{\text{size}_h}}{v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}} > 25. \tag{F.7}$$

In Figure F.6, the points labeled as unfiltered corresponds to those for which the filter is disabled according to Equation F.7. One may notice that the filter tends to be disabled for the cases where the estimated value of α exceeds more significantly its optimal value (the error is positive). Thus, it is possible to identify most cases where the filtering process could be too aggressive and then, not advantageous, as it could blur the image, degrading the existing details.

In order to preserve natural edges on deblocked images, the value of s is adapted in accordance to the strength α found for the filter. If a large α was adopted, the image needs strong deblocking, and the threshold to switch-off the filter must also be high. If α is low, then the image has a large amount of high frequency details, and the value of s needs to be decreased in order to avoid filtering natural edges. The value of s is related to α through the equation:

$$s = 50 + 250\alpha. \tag{F.8}$$

Thus, a minimum threshold of 50 is used and the value of s increases for increasing values of α.

Using Equations F.6 and F.8, the proposed method is able to adapt the filtering parameters for each frame of a given video sequences only based on its local characteristics.

The impact of the initial block dimensions on the objective quality gains was also evaluated for the proposed method. It can be seen that the use of large blocks does not affect negatively the performance of the algorithm, as these blocks tend to be segmented every time they are not advantageous. However, blocks with more than 16 pixels in length are rarely used, and even when the mapping procedure produces such large blocks, there are no quality gains associated to their use. They only contribute to increasing the overall computational complexity of the algorithm. On the other hand, using small initial blocks restricts the maximum smoothing power achievable by the filter, and consequently the

maximum gains that the method can achieve. We have verified that 16×16 blocks are the best compromise for most situations, even for images compressed with HEVC using 64×64 blocks. Therefore, we adopted 16×16 blocks as a default parameter for the method, without a significant impact in its performance.

F.3.5 Computational complexity

The computational complexity of the proposed deblocking filter method mainly depends on two procedures: the creation of the filter map and the deblocking process itself.

The mapping procedure has a very low computational complexity, when compared to the filtering process. It only requires to subtract each pixel from its predecessor neighbor, accumulate the result, and compare the accumulated value with the threshold τ, in order to decide whether to segment or not the current block. This can be regarded as a linear function $\mathcal{C}(n)$, which depends on n, the number of pixels of the entire image. Only integer operations need to be performed.

The filtering process is a direct implementation of bilateral filters, whose computational complexity is also a linear function $\mathcal{C}(n.m)$, that depends on n, the number of pixels in the image, and m, the number of pixels of the filter support, as the deblocked image is obtained through the convolution of the input image with the deblocking kernel. As variable filter support lengths are used, the computational complexity is maximized by the case where the maximum block size is used for all the pixels on the image.

Consequently, the resulting computational complexity is comparable to those from the methods from [78] and [87], and significantly lower to that from [126], which performs an adaptive multipass deblocking. The computational complexity is also considerably lower than that of interactive methods, such as the method presented in [91].

F.4 Experimental results

The performance of the proposed method was evaluated not only for still images, but also for video sequences, through comparison between several state-of-the-art block based encoders.

F.4.1 Still image deblocking

In our experiments, the performance of the proposed method was evaluated not only for images encoded using MMP, but also using two popular transform-based algorithms: JPEG and H.264/AVC, in order to demonstrate its versatility. Furthermore, we present results corresponding to four still images with different characteristics, ranging from smooth

Table F.1: Results for the deblocking of MMP coded images [dB]

Lena	Rate (bpp)	0.128	0.315	0.442	0.600
	No filter	31.38	35.54	37.11	38.44
	Original [94]	31.49	35.59	37.15	38.45
	Proposed	**31.67**	**35.68**	**37.21**	**38.48**
Peppers	Rate (bpp)	0.128	0.291	0.427	0.626
	No filter	31.40	34.68	35.91	37.10
	Original [94]	31.51	34.71	35.92	37.10
	Proposed	**31.73**	**34.77**	**35.95**	**37.11**
Barbara	Rate (bpp)	0.197	0.316	0.432	0.574
	No filter	27.26	30.18	32.39	34.43
	Original [94]	27.26	30.18	32.39	34.43
	Proposed	**27.38**	**30.31**	**32.51**	**34.52**
PP1205	Rate (bpp)	0.378	0.576	0.765	1.005
	No filter	27.13	31.08	34.54	37.67
	Original [94]	27.13	31.08	34.54	37.67
	Proposed	**27.14**	**31.09**	**34.54**	**37.67**

natural images Peppers and Lena, to text image PP1205, in order to illustrate the performance of the proposed method under several operating conditions.

For images compressed using MMP, the proposed method was compared to the method proposed in [94]. As MMP is not a normative encoder, the filter parameters may be optimized by the encoder and transmitted to the decoder, in order to maximize the PSNR of the reconstruction. With this approach, similar to the one used by [94], the best objective quality gains are always achieved, and we have the guarantee that the deblocking filter never degrades the image's PSNR.

Consistent objective image quality gains, as well as more effective deblocking effect were obtained, when compared to the method presented in [94]. The improved mapping procedure used to estimate the block dimensions proposed in this paper eliminates the effects of erroneous consideration of accurately predicted blocks, as smooth blocks observed in [94]. This avoids the exaggerate blurring of some detailed regions, with impact on the PSNR value of the filtered image. Furthermore, the new mapping procedure allows the use of a stronger deblocking in smooth regions, without degrading image details.

The comparative objective results are summarized in Table F.1, while Figures F.7 and F.8 present a subjective comparison between the two methods. We can observe that the proposed method achieves higher PSNR gains than the method from [94], for all cases. Additionally, one can see in Figures F.7 and F.8 that the blocking artifacts are more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed image. High frequency details, like the ones on the headscarf from image Barbara, are successfully preserved by the proposed method.

(a) No deblocking 31.38dB

(b) Method from [94] 31.49dB (+0.11dB)

(c) Proposed method 31.67dB (+0.29dB)

Figure F.7: A detail of image Lena 512×512, encoded with MMP at 0.128 bpp.

(a) No deblocking 30.18dB

(b) Method from [94] 30.18dB (+0.00dB)

(c) Proposed method 30.31dB (+0.13dB)

Figure F.8: A detail of image Barbara 512×512, encoded with MMP at 0.316 bpp.

Table F.2: Results for the deblocking of H.264/AVC coded images [dB]

Lena	Rate (bpp)	0.128	0.260	0.475	0.601
	No filter	31.28	34.48	37.20	38.27
	Original [78]	31.62	34.67	37.24	38.27
	Proposed	**31.63**	**34.72**	**37.31**	**38.31**
Peppers	Rate (bpp)	0.144	0.249	0.472	0.677
	No filter	31.62	33.77	35.89	37.09
	Original [78]	**32.02**	33.99	35.90	37.02
	Proposed	31.98	**33.99**	**35.95**	**37.11**
Barbara	Rate (bpp)	0.156	0.321	0.407	0.567
	No filter	26.36	29.72	31.13	33.33
	Original [78]	26.54	**29.87**	**31.28**	**33.45**
	Proposed	**26.59**	29.84	31.25	33.42
PP1205	Rate (bpp)	0.310	0.586	0.807	1.022
	No filter	23.95	27.61	30.37	32.91
	Original [78]	**24.05**	**27.75**	**30.55**	**33.09**
	Proposed	24.03	27.65	30.35	32.91

The inefficiency of the method from [94] becomes evident in Figure F.8. The high frequency patterns from the headscarf tend to be efficiently predicted, and coded using relatively large blocks. The mapping generated by the deblocking procedure did not reflect the high frequency present in these regions, and the patterns tend to be considerably blurred. As a result, the deblocking filtering is disabled, in order to avoid the image's PSNR degradation.

It is also important to notice that the adaptability of the proposed method allows to disable the deblocking filter for non-smooth images, such as text documents (PP1205), thus preventing highly annoying smoothing effects.

The versatility of the proposed method was evaluated by comparing it with the in-loop filter of H.264/AVC [78]. Images were encoded with JM 18.2 reference software, with and without the use of the in-loop filter. The non-filtered images were then subjected to a post-filtering with the proposed method. In order to preserve the compliance with the H.264/AVC standard bit-stream, the default values for α, s and τ proposed in Section F.3.4 were used.

The objective results presented in Table F.2, for four different images with different natures, demonstrate that the proposed method is able, in several cases, to outperform the objective quality achieved by the H.264/AVC in-loop filter.

Figures F.9 and F.10 present a subjective comparison between the proposed method and the in-loop filter of H.264/AVC [78]. In the reconstructions obtained with the in-loop filter disabled (Figures F.9a and F.10a), blocking artifacts are quite obvious at such compression ratios.

(a) In-loop deblocking [78] disabled 30.75dB

(b) In-loop deblocking [78] enabled 31.10dB (+0.35dB)

(c) Proposed method 31.11dB (+0.36dB)

Figure F.9: A detail of image Lena 512×512, encoded with H.264/AVC at 0.113 bpp.

(a) In-loop deblocking [78] disabled 29.72dB

(b) In-loop deblocking [78] enabled 29.87dB (+0.15dB)

(c) Proposed method 29.84dB (+0.12dB)

Figure F.10: A detail of image Barbara 512×512, encoded with H.264/AVC at 0.321 bpp.

Table F.3: Results for the deblocking of JPEG coded images [dB]

Lena	Rate (bpp)	0.16	0.19	0.22	0.25
	No filter	26.46	28.24	29.47	30.41
	Method from [87]	**27.83**	**29.55**	**30.61**	**31.42**
	Proposed	27.59	29.32	30.46	31.29
Barbara	Rate (bpp)	0.20	0.25	0.30	0.38
	No filter	23.49	24.49	25.19	26.33
	Method from [87]	**24.39**	**25.26**	**25.89**	**26.86**
	Proposed	24.18	25.03	25.52	26.42
Peppers	Rate (bpp)	0.16	0.19	0.22	0.23
	No filter	25.59	27.32	28.39	29.17
	Method from [87]	**27.33**	**28.99**	**29.89**	**30.54**
	Proposed	26.64	28.14	29.10	29.74

It is also interesting to notice that the deblocking artifacts have a different distribution than those presented in Figures F.7a and F.8a, respectively. This happens because, unlike MMP, H.264/AVC only uses a limited set of pre-established block sizes. This results in the appearance of blocking artifacts in less regions, all at predictable locations (the grid that defines those transform block's boundaries), but that tend to be more pronounced for similar compression ratios.

In Figures F.9b and F.10b, it can be seen that the in-loop filter used by H.264/AVC [78] is effective in reducing the deblocking artifacts, at the cost of blurring some details. The reconstruction obtained using the proposed method, presented in Figures F.9c and F.10c, respectively, shows at least an equivalent perceptual quality, with a marginal objective performance advantage in most cases, with all the advantages of using a post-processing filter instead of an in-loop filter. Furthermore, the use of the pre-established parameters' values result in a fully compliant algorithm.

The proposed method was also tested using images encoded with JPEG. Significant quality improvements were also achieved in this case, as seen in Table F.3. In this case, the proposed method was compared with the method from [87]. The method from [87] is specifically optimized for JPEG, since it takes advantage of the knowledge regarding the possible location of artifacts (JPEG uses a fixed 8×8 transform) and the artifact strength (using information from the image's quantization table), unlike the proposed method that does not make any assumption about the image coding structure. The previous consideration justifies the gains presented by [87]. However, the proposed method is still able to achieve a significant objective and perceptual quality improvement for these cases, with results very close from those from the method proposed in [87].

(a) Original 30.41dB

(b) Method from [87] 31.42dB (+1.01dB)

(c) Proposed method 31.29dB (+0.88dB)

Figure F.11: A detail of image Lena 512×512, encoded with JPEG at 0.245 bpp.

(a) Original 26.62dB

(b) Method from [87] 27.13dB (+0.51dB)

(c) Proposed method 26.69dB (+0.08dB)

Figure F.12: A detail of image Barbara 512×512, encoded with JPEG at 0.377 bpp.

In Figures F.11 and F.12, we present the objective results obtained by both methods, for images Lena and Barbara. Blocking artifacts are evident in both the original reconstructions (Figures F.11a and F.12a) at such compression ratios. From Figures F.11b and F.12b, it can be seen that the method from [87] is able to significantly reduce the amount of blocking artifacts, increasing the perceptual quality of the reconstructed images. From Figures F.11c and F.12c, it can be seen that despite the lower quality gain, the proposed method is also able to significantly reduce the amount of blocking artifacts, specially on images with a low pass nature, such as image Lena. For image Barbara, the reduction of blocking artifacts does not work so well, but the method was still able to increase the perceptual quality for this image.

Figure F.13 summarizes the objective results achieved by the proposed method, when used to filter four different images, compressed using the three tested encoders. In order to illustrate the performance of the proposed method for a wide range of image types, the results are presented for images with different levels of details. Images Lena, Goldhill and Barbara are natural images presenting, respectively, low, medium and high levels of detail. Image PP1205 results from a scanned text document, and presents a large amount of high frequency transitions, in order to evaluate how the method performs in extreme conditions.

The figure shows the PSNR gain achieved by the proposed method, over the non-filtered reconstruction. For H.264/AVC and JPEG, the gain is presented using the pre-determined parameters' values, but also the optimal values, that are obtained by testing all possible values, in order to evaluate the impact of the proposed approximation. It can be seen that the PSNR gain obtained using the pre-determined values is close to that obtained using the optimal parameters at high compression ratios. This corresponds to the case where a strong filtering is most needed, as blocking artifacts are in these cases more pronounced. The difference tends to increase for highly detailed images, because the default parameters were defined using a conservative approach, in order to avoid applying an aggressive filtering, which would introduce blurring on high detailed regions.

An interesting phenomenon can be observed in Figure F.13d. Contrarily to the tendency observed for the other images, the PSNR gain for image PP1205, compressed using JPEG, increases when the compression ratio decreases, for the bitrate range presented in the figure. This happens because the PSNR achieved using JPEG at such compression ratios is low for this highly detailed image, with almost all details being degraded. Consequently, the deblocking filter does not have sufficient information to increase the general image's quality. When the compression ratio decreases, the amount of detail increases, and the filter becomes able to more significantly increase the PSNR gain. This tendency is however inverted when the reconstruction achieves a high level of detail, and the filtering process ceases to be beneficial. The gain then starts to decrease, in accordance to the other results. This inflexion point occurs at approximately 1.4 bpp, for the case of image

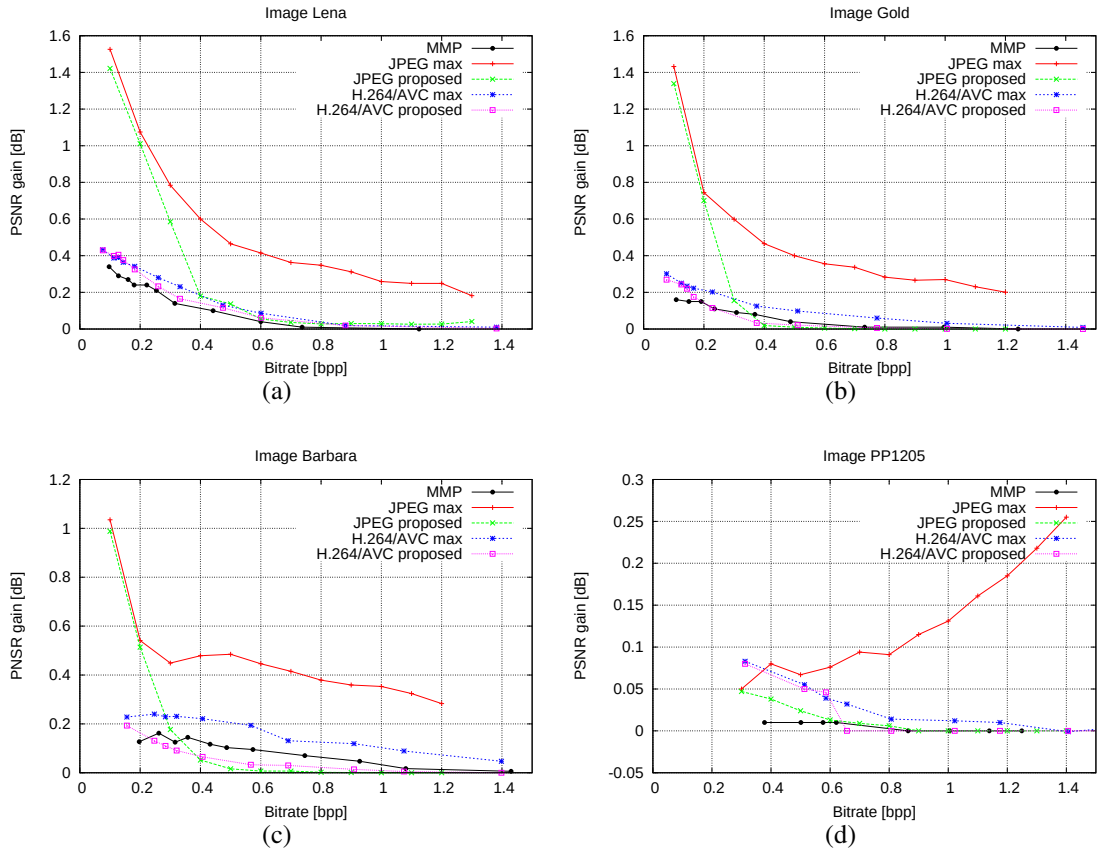


Figure F.13: Comparative results for the images Lena, Goldhill, Barbara and PP1205 (512×512).

PP1205.

F.4.2 Video sequences deblocking

The proposed method was also evaluated for video sequences' deblocking. The objective was to access its performance when applied to Inter-coded frames. This approach imposes the additional challenge of locating blocking artifacts on Inter-frames, as motion compensation using non-deblocked references may introduce blocking artifacts in any location of the reconstructed images. This is different from the case of Intra-frames, where these artifacts only appear at the block boundaries. Additionally, disabling the in-loop filter causes the Inter-frames to be encoded using non-deblocked references, which reduces the probability of finding good matches for the blocks during motion estimation (ME). As a result, the motion compensation residue energy increases, decreasing the compression efficiency of Inter-frames and consequently the overall performance of the compression algorithm. For these reasons, achieving competitive results using a post-processing deblocking algorithm requires higher quality gains than that of an equivalent in-loop method, in order to compensate the lower ME performance.

The experimental tests were performed using the JM18.2 H.264/AVC reference soft-

Table F.4: Results for the deblocking of H.264/AVC coded video sequences [dB]

		In-Loop [78] ON		In-Loop [78] OFF			Proposed		Increase
	QP	Bitrate	PSNR	Bitrate	PSNR	BD-PSNR	PSNR	BD-PSNR	BD-PSNR
	[I/P-B]	[kbps]	[dB]	[kbps]	[dB]	[dB]	[dB]	[dB]	[dB]
Rush Hour	48-50	272.46	30.62	288.24	29.99	-0.85	30.69 (+0.70)	-0.16	0.69
	43-45	478.65	33.62	500.79	32.92		33.67 (+0.75)		
	38-40	865.48	36.38	903.19	35.75		36.42 (+0.67)		
	33-35	1579.27	38.76	1636.38	38.23		38.80 (+0.57)		
Pedestrian	48-50	409.69	28.68	420.79	28.18	-0.59	28.75 (+0.56)	-0.14	0.45
	43-45	711.64	31.93	730.58	31.43		32.01 (+0.58)		
	38-40	1216.63	34.89	1243.96	34.44		34.83 (+0.39)		
	33-35	2080.58	37.43	2107.30	37.09		37.17 (+0.08)		
Blue Sky	48-50	572.47	26.74	583.90	26.40	-0.31	26.71 (+0.30)	-0.11	0.20
	43-45	912.33	30.25	924.21	29.99		30.28 (+0.29)		
	38-40	1557.29	33.77	1566.44	33.54		33.73 (+0.19)		
	33-35	2737.99	37.10	2740.06	36.90		36.82 (-0.08)		

ware, operating at high profile, either enabling or disabling the in-loop deblocking filter [78]. The sequences compressed with the in-loop filter disabled were then subjected to a post-filtering using the proposed method, with the parameter estimation proposed in Section F.3.4. Thus, the filtering parameters were adjusted depending on the features of each frame, according to the Equations F.6 and F.8, using $\tau = 32$.

Consequently, the bitrate presented by the non-filtered sequence must be also considered for the sequence reconstructed with the proposed method. A set of commonly used parameters was adopted for these experiments, namely a GOP size of 15 frames, with an IBBPBBP pattern at a standard frame-rate of 25 fps. For ME, the Fast Full Search algorithm was adopted, with a 32 pixels search range and 5 reference frames. Only the PSNR values of the luma component are presented as references, being representative to the overall results. Table F.4 summarizes the results obtained for the first 128 frames of three high definition (1920×1080 pixels) well known test sequences.

Unlike the case of still images, which are compressed as Intra-frames, the difference in the ME efficiency contributes to a significant difference between the achieved bitrates with the various filters, making a direct comparison of the results difficult. In order to improve this comparison, we computed the Bjøntegaard delta (BD) PSNR [84] for the two sets of results. The BD-PSNR presented for the in-loop OFF indicates the average PSNR loss incurred from disabling the in-loop deblocking filter described in [78], in the interval of overlapping bitrates of both sets of results. The BD-PSNR presented for the proposed method provide the comparison between the results obtained using the proposed method, over the sequence compressed using the H.264/AVC in-loop deblocking filter [78]. In order to indicate the objective quality gains achieved by the proposed method, the BD-PSNR presented on the last column indicates the average PSNR gains achieved by the proposed method over the non-filtered reconstructed video sequence.

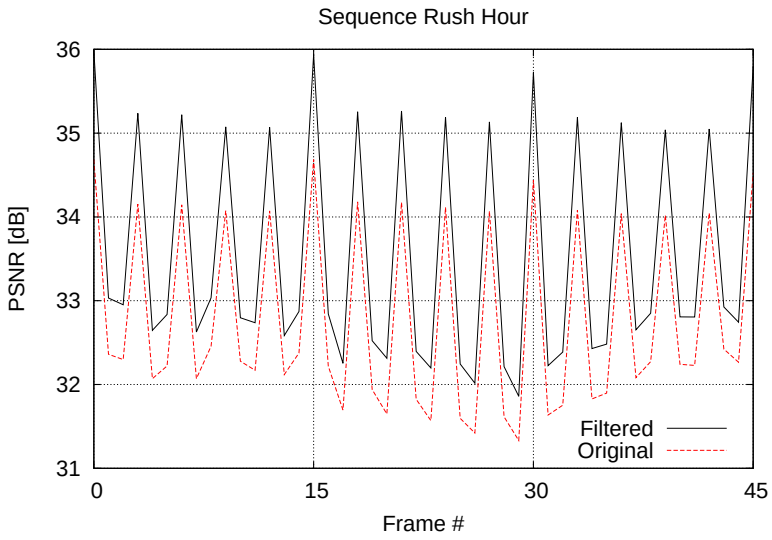


Figure F.14: PSNR of the first 45 frames of sequence Rush Hour, compressed using QP 43-45, with the H.264/AVC in-loop filter disabled, and the same 45 frames deblocked using the proposed method.

As can be seen in Table F.4, the post-filtering using the proposed method is able to significantly increase the average objective quality of the reconstructed video sequences, achieving global results close to those obtained enabling the H.264/AVC in-loop filter [78]. For the sequence Rush Hour, the average PSNR has a BD-PSNR decrease of up to 0.85dB when the H.264/AVC in-loop filter is disabled, but the proposed method is able to reduce the performance gap to just 0.16dB, which represents an average PSNR gain of 0.69dB on the interval of the tested bitrates.

One may also observe that the PSNR gains are approximately constant for all types of frames, as shown in Figure F.14. This figure presents the PSNR both for the original (non-filtered) and post-processed first 45 frames from sequence Rush Hour, encoded using QP 43-45 (for I/P and B frames). These results demonstrate that the proposed method is able to efficiently identify blocking artifacts also on Inter-frames, enhancing both the subjective and objective quality for all frames from the video sequence, independently of the coding tools used in their compression.

The independence observed relatively to the compression tools used in the encoding process motivated another experimental test. It was observed that disabling the in-loop filter affects considerably the performance of H.264/AVC, as it results in reference frames with lower quality, affecting the ME efficiency. Thus, the proposed method was tested for sequences encoded with H.264/AVC, disabling the in-loop filter only for the non-reference frames. With this approach, there is no performance degradation on ME, allowing a more direct comparison between the gains resulting from the proposed method and those from the H.264/AVC in-loop filter. Figure F.15 presents the obtained results, for the first 45 frames of sequence Rush Hour, compressed using QP 43-45.

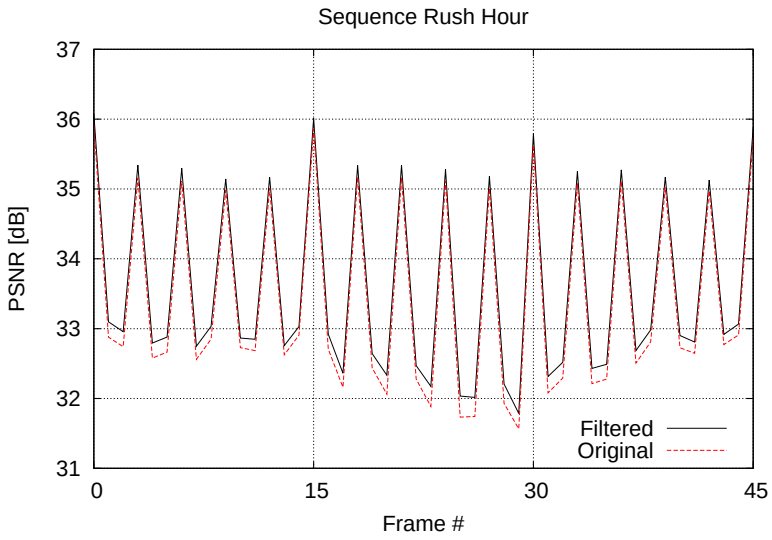


Figure F.15: PSNR of the first 45 frames of sequence Rush Hour, compressed using QP 43-45, with the H.264/AVC in-loop filter disabled only for B frames, and the same 45 frames deblocked using the proposed method.

In this case, the proposed method was able to outperform the H.264/AVC in-loop filter, with a BD-PSNR gain of 0.10 dB, when applied to all frames of the video sequence. Figure F.15 shows that the proposed method was not only able to improve the objective quality of the Inter-frames where the in-loop filter was not applied, but also to increase the objective quality of the reference frames that were already filtered using the in-loop filter. In this case, the increase of the PSNR is not significant, but the details and the objective quality are preserved. This was also the case for when the proposed method was applied to video sequences encoded with the H.264/AVC in-loop filter enabled for all frames, with gains up to 0.08 dB on the average PSNR. This can be important while post-processing a given video sequence, as it works regardless of the use or not of the loop filter, corroborating the fact that the proposed post-deblocking filter does not require any knowledge regarding the way it was encoded.

It is important to notice that, unlike the case of the H.264/AVC in-loop filter [78], where its in-loop nature imposes that both the encoder and the decoder must perform the same filtering tasks in order to avoid drift, the proposed method only requires the decoder to filter the reconstructed video sequence. Additionally, the filter can be switched on and off for arbitrary frames. It can be switched off when the computational resources are of large demand, and can be switched on when more computational resources are available. Such adaptability is not possible with the techniques from [78], where disabling the filters at some point of the decoding process results on the loss of synchronism between the encoder and the decoder.

The proposed method was also used to deblock video sequences encoded with the up-coming, highly efficient HEVC video coding standard [16]. For that purpose, we used the

Table F.5: Results for the deblocking of HEVC coded video sequences [dB]

		In-Loop [78] ON		In-Loop [78] OFF			Proposed		Increase
	QP	**Bitrate**	**PSNR**	**Bitrate**	**PSNR**	**BD-PSNR**	**PSNR**	**BD-PSNR**	**BD-PSNR**
	[I/P-B]	[kbps]	[dB]	[kbps]	[dB]	[dB]	[dB]	[dB]	[dB]
Rush Hour	48	214.14	34.41	216.27	34.00	-0.47	34.27 (+0.26)	-0.19	0.28
	43	404.23	36.69	410.19	36.25		36.53 (+0.28)		
	38	785.83	38.75	798.08	38.32		38.60 (+0.28)		
	33	1655.04	40.67	1683.73	40.28		40.53 (+0.25)		
Pedestrian	48	322.32	32.68	321.74	32.25	-0.39	32.49 (+0.24)	-0.21	0.18
	43	576.87	35.20	575.34	34.76		35.00 (+0.24)		
	38	1040.16	37.50	1036.70	37.11		37.28 (+0.17)		
	33	2003.40	39.68	1995.11	39.36		39.39 (+0.02)		
Blue Sky	48	414.47	32.10	422.10	31.52	-0.62	31.54 (+0.02)	-0.67	-0.05
	43	715.92	35.01	727.25	34.45		34.44 (-0.01)		
	38	1261.90	37.80	1284.25	37.26		37.20 (-0.05)		
	33	2327.27	40.41	2369.07	39.95		39.78 (-0.17)		

HM5.1 reference software, disabling both the in-loop and the ALF deblocking filter [126]. The unfiltered sequences were then subjected to a post-processing using the proposed algorithm, and the results are compared with those obtained by HEVC with both filters enabled.

The main objective of these tests was to evaluate the performance of the proposed method for the upcoming video standard, that uses a new set of coding tools, such as 64×64 unit blocks *vs.* the 16×16 blocks used on H.264/AVC and 8×8 blocks from JPEG. Some default parameters for HM5.1 were used in the experiments, such as a hierarchical-B (with an intra-period of 8) configuration. A gradual 1 QP increment was used for Inter-frames at higher levels. Motion estimation used the EPZS algorithm, with a 64 pixels search range.

The results are summarized in Table F.5, for the same video sequences used to evaluate the deblocking in H.264/AVC (Table F.4), in order to allow a direct comparison between the deblocking performance for both video codecs. As in H.264/AVC, disabling the filters has a significant impact on the compression efficiency of the algorithm. PSNR losses of up to 0.5dB can be observed in some cases. Despite being outperformed by the HEVC filtering tools, the proposed method was able to significantly enhance the objective quality of the reconstructions in most cases, with increases of up to 0.28dB in the average PSNR of the reconstructed sequences. This demonstrates once more the versatility of the proposed post-processing algorithm. Additionally, the subjective quality of the video signal was globally increased, with a considerable reduction of blocking artifacts.

These results demonstrate that, despite not being as efficient as the highly complex and algorithm-specific HEVC deblocking filters [126], the proposed method is still able to present a consistent performance when applied to signals encoded using this algorithm. This corroborates the high versatility of the proposed method and its independence rela-

tively to the encoding tools used to compress the input images. Furthermore, the higher performance presented by [126] comes at the expense of a higher computational complexity resultant from the multipass adaptive filter. Such as in the case of the H.264/AVC in-loop filter, activating HEVC filters [126] imposes that both the encoder and the decoder need to perform this task, in order for them to remain syncronized, avoiding drift. Therefore, it is also not possible to switch the HEVC filters on and off arbitrarily in the decoder, according to the availability of computational resources.

F.5 Conclusions

In this appendix we present an image post deblocking scheme based on adaptive bilateral filters. The proposed method performs a total variation analysis of the encoded images in order to build a map which defines the filter's shape and length for each region on the image. Regions with low total variation are assumed to have a smooth nature and are strongly filtered, using filters with wide support regions to eliminate the artifacts in the blocks' boundaries. Regions with high total variation are assumed to contain a high level of detail, and are only softly filtered, or not filtered at all. The ability to reduce the length of the filter's support region or even to disable the filtering minimizes the blurring effect caused by filtering these regions.

Unlike other approaches, the proposed technique is universal, not being specifically tailored for any type of codec, being applicable both to still images and video sequences. This is confirmed by the objective and subjective image quality gains that have been observed for several tested codecs, namely MMP, JPEG, H.264/AVC and the upcoming standard HEVC. Additionally, the method is a post-processing technique, which does not impose the transmission of any side information, resulting in a fully compliant algorithm regardless of the codec used to compress the image.

Appendix G

Compression of volumetric data using MMP

G.1 Introduction

Image and video compression algorithms based on two-dimensional multiscale reccurent patterns have been widely investigated over the past few years. Several compression schemes were proposed for a wide range of applications, as discussed in Appendix B, with some of them achieving state-of-the-art compression performances. This allowed to demonstrate the potential of such approach, motivating to spend more time in further researches to improve its performance and to increase the number of applications.

It is now important to search for new insights of the MMP algorithm, exploiting new tools and new approaches for multimedia signal compression. In this appendix, we investigate a new compression framework based on a three-dimensional extension of the MMP algorithm.

In [95], a three-dimensional extension of MMP was proposed, for meteorological radar signals compression. Despite the high compression efficiency achieved for this particular application, the method proposed in [95] was based on an early version of MMP, where some compression techniques that considerably improved the performance of two-dimensional MMP-based compression algorithms were not available yet. Since then, new techniques have been introduced, such as the use of a hierarchical predictive scheme [15], the flexible partition [5] or the improved dictionary design techniques proposed in [49].

The main interest of developing a volumetric compression layout lies in the multitude of applications which can benefit from such approach. Many signals are volumetric by nature, such as meteorological radar signals, tomographic scans or multispectral images, among many others. Furthermore, some other types of signals whose compression traditionally relies into two-dimensional techniques could also benefit from such approach, such as the case of video sequences.

A video sequence is a temporal succession of image frames, and can thus be conceived as a three-dimensional signal, with one temporal and two spatial dimensions. Generally, a high level of spatial and temporal correlation exists, and the success in exploiting such redundancies is the key feature for the rate-distortion performance of a video encoder.

As discussed in Appendix D, most modern video compression schemes rely on a hybrid architecture, using a frame by frame motion compensation to exploit the temporal correlation, and some two-dimensional compression methods to encode the generated residue and extract the remaining spatial redundancy. This approach has been the basis for the successful H.26x [45] family of standards and is expected to be used in the upcoming HEVC video coding standard [16]. Despite the high compression efficiency achieved by hybrid video codecs, motion compensation presents some impairment for certain applications. It is a very computationally demanding operation, and it does not perform well for some types of movements, such as non-translational motion, which includes rotations, zoom and shearing of objects. This motivated the research for alternative approaches to efficiently exploit the temporal redundancy.

In the literature, several works already suggested to approach video signals from a volumetric point-of-view, using straightforward 3D extensions of well-known 2D compression methods, to reduce the spatiotemporal redundancy. This corresponds to process the video data directly as a three-dimensional volumetric signal, instead of using a frame-by-frame approach. For example, the use of a 3D fractal for video compression was proposed in [96], and several researchers suggested using 3D extensions of the Discrete Cosine Transform (DCT) [97–100] and Discrete Wavelet Transform (DWT) [101–104] for video compression purposes.

Earlier proposals [97–99, 101] suggested to apply the 3D transforms directly on the input video data. Despite being able to achieve a very efficient representation of slow movements, where the energy concentrates on the low frequency temporal coefficients, the performance of such methods degrades considerably in the presence of complex and non-uniform motion. In this case, the energy spreads along the higher frequency temporal coefficients, restricting the energy compaction property of the transforms. This motivated the study of an alternative class of algorithms, which perform some kind of motion compensation before applying the transform [100, 102, 104]. Several solutions have been proposed, either through filtering along motion trajectories [102], by projecting all frames onto a reference coordinate system [104], or by explicitly using motion information during scanning and quantization of transform coefficients [100]. However, despite the excellent computational complexity *vs.* compression performance ratio achieved by some of these algorithms, none of them resulted in a competitive alternative to the state-of-the-art hybrid video codecs.

In this appendix, we propose a new volumetric compression framework, named 3D-MMP, supported by the MMP algorithm and volumetric prediction tools. The proposed

framework is intended to be used to encode several types of volumetric signals, so we start by describing a generic three-dimensional predictive MMP-based encoding algorithm. Next, we present some optimizations specifically oriented towards video compression. The performance of the resulting algorithm is then evaluated for this particular application.

In order to replace the traditional frame by frame intra and inter prediction techniques, we adopted least squares [47] and directional predictions to perform the spatiotemporal de-correlation. The remaining residue is then encoded with a volumetric MMP, using a three-dimensional extension of the flexible partition scheme [5]. The adaptation to local image features is inherent to the flexible partition scheme [5] and is improved by the use of weighted spatial and temporal predictions such as the ones proposed in [47], where a backward adaptive spatiotemporal predictor was proposed for video compression.

Based on the duality between edge contours and motion trajectories, this method generates a least squares prediction for each pixel, using the behavior of its spatial and temporal neighbors. This allows to simultaneously exploit spatial and temporal redundancies, through an implicit approach that does not require the transmission of any overhead. Experimental results have shown that this method is able to generate predictions with a lower error than motion compensation with quarter pixel accuracy [128]. Particularly, this method reveals an intrinsic ability to adapt to recursive complex patterns and textures, for which most of the other prediction methods tend to fail.

In [46], an enhancement of this method was proposed, in order to adapt the algorithm for two-dimensional block-based still image coding. This approach showed that the least squares prediction is able to achieve significant performance gains, using suitable adaptive filter's supports and causal neighborhood training regions. Furthermore, it demonstrated that the use of the pixel's prediction instead of its encoded value does not compromise the prediction accuracy, when the pixel is needed on the filter's support and its corresponding residue value is not yet available. Competitive results for spatial [46], inter-component [123] and inter-view [129] redundancy exploitation, have demonstrated the potential of the least squares prediction approach for block-based image compression. Thus, the use of the least squares prediction method on a 3D block-based layout may be a promising approach in the development of unified spatiotemporal predictors. Experimental results have shown that the developed volumetric prediction scheme may be able to deal with a wide range of possible motion types, including non-linear motions, where the classical approaches tend to fail.

This appendix is organized as follows. Section G.2 presents a description of the proposed compression architecture, based on volumetric MMP and prediction. The main modifications performed to adapt the MMP algorithm to volumetric signal compression are described, as well as the prediction tools adapted to work on a volumetric layout. In Section G.3, we present some additional modifications specifically oriented to dapt the

proposed layout for video coding applications. Experimental results for video compression are presented in Section G.4, and Section G.5 summarizes the main conclusions of this appendix.

G.2 A volumetric compression architecture

The architecture adopted for the proposed volumetric data compression framework relies in a volumetric extension of the MMP algorithm, which uses a hierarchical prediction [15] and the three-dimensional extension of the flexible partition scheme [5].

In this section, we describe the main adaptation performed on the former techniques, in order to adapt them for a volumetric framework.

G.2.1 3D-MMP

In [95], the use of MMP was already proposed to encode volumetric signals (meteorological radar signals), but with some major differences relatively to the objectives defined for the present work. The algorithm from [95] was based on an earlier version of MMP, and lacked some improvements that significantly increased the algorithm's performance for natural images compression. In this work we will study the influence of such methods for volumetric signals compression. As examples of such improvements, we may refer to the flexible partition scheme [5], the use of the original block scale as a context for the arithmetic coder [49], the use of block transforms to improve the dictionary's approximation power [49], or the dictionary growth control techniques [4], which avoided the insertion of codewords similar to the existing ones.

When compared with the conventional 2D MMP algorithm described on Appendix B, the first major difference is that the basic unit is no longer a generic 2D rectangle $X^l_{m,n}$ with $N \times M$ pixels, but a 3D parallelepiped $X^l_{m,n,k}$ with $N \times M \times K$ pixels. As a direct consequence, the amount of possibilities to divide a given block increases significantly. Considering the flexible partition scheme, a given block from scale l with $N \times M \times K$ pixels, where $N \neq 1$, $M \neq 1$ and $K \neq 1$, can be divided along the three axis that comprise the 3D space, as can be seen on Figure G.1.

The observation of Figure G.1 suggests the existence of an additional segmentation flag, in order to signal the block segmentation along the k_3 axis. In the algorithm presented in [3], only two flags were necessary to signal whether to segment or not a given block. In [15], the adoption of a predictive scheme resulted in an additional flag, in order to distinguish situations where both the prediction and residue blocks were segmented, and situations where the prediction was no further segmented, but the residue block was. The number of flags increased once more with the flexible segmentation scheme. In this case, each block can typically be either vertically or horizontally segmented, with the residue

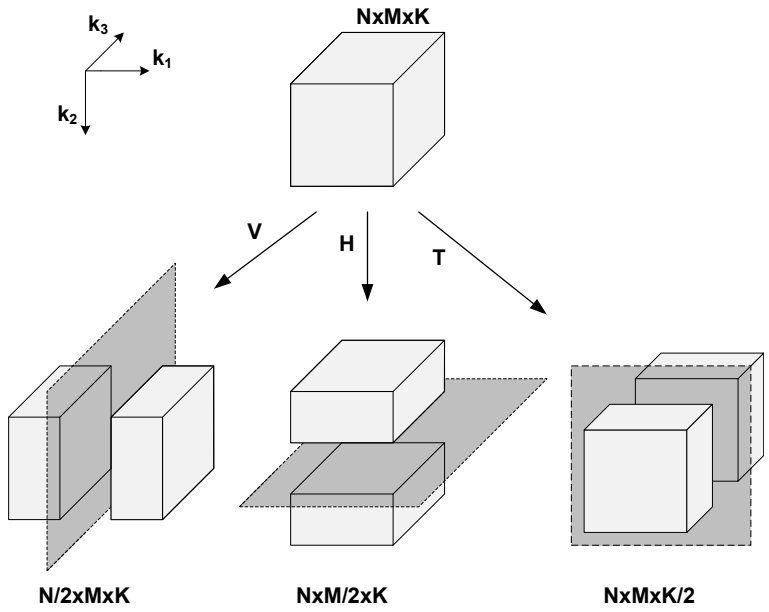


Figure G.1: Triadic flexible partition.

or both the residue and prediction segmentations generically possible on each case. As an extension of the former case, the existence of an additional direction to segment the block introduces two more flags, totalizing seven different flags to signal the segmentation pattern on the bitstream:

- NS - The node is a tree leaf (the original block is not segmented);
- PV - The node corresponds to a vertical segmentation of both the residue and the prediction blocks;
- PH - The node corresponds to a horizontal segmentation of both the residue and the prediction blocks;
- PT - The node corresponds to a transversal segmentation of both the residue and the prediction blocks;
- RV - The node corresponds to a vertical segmentation of the residue block only;
- RH - The node corresponds to a horizontal segmentation of the residue block only;
- RT - The node corresponds to a transversal segmentation of the residue block only.

Note that, alternatively, only two ternary flags could be used to identify the possible occurrences. One flags would be used to signalize that the block is not segmented, that only the residue is segmented or that both the prediction and residue are segmented, respectively, with the second flag indicating the optimal direction for that segmentation (if it occurred). However, we have observed that the adopted scheme favored the arithmetic encoder's adaptation process [48], presenting marginal performance gains.

As more combinations of block sizes become possible, the number of scales on the volumetric framework MMP dictionary also increases considerably. In Equation B.6, we have seen that the total number of scales is given by the product of the possible block sizes along each of the block's dimensions. Thus, extending Equation B.6 to a 3D layout, the total number of scales can be obtained by the following equation:

$$N_{scales} = (1 + \log N) \times (1 + \log M) \times (1 + \log K), \tag{G.1}$$

where N, M and K are powers of two, and define the size of the initial blocks used by MMP.

For example, consider a $16 \times 16 \times 16$ pixels block. In the volumetric layout, the block can be segmented in 5 different locations along each of the 3 dimensions, corresponding to a 16, 8, 4, 2 or 1 pixel width. Thus, an initial block size of $16 \times 16 \times 16$ pixels results in 125 different scales in the volumetric layout. Note that a 16×16 pixels block only could be segmented along 2 directions on the 2D layout, resulting in a total of 25 different scales. It is also important to notice that the increase in the number of block scales has a significant impact in many practical aspects of the MMP algorithm. For example, it impacts on the performance of the arithmetic coder, as the block scale is used as a context while compressing each symbol from the bitstream. Additionally, it determines the computational complexity of the algorithm, as it increases the amount of possible segmentations, and consequently the number of matching procedures that need to be performed.

In Appendix B, we performed a formal derivation of the computational complexity of MMP-FP, for the case where no redundant node are optimized. Here we will review the formal derivation presented on Equation B.16 in order to extend this equation for the volumetric case.

Generically, we have shown that the computational complexity of the MMP matching procedure can be obtained by multiplying the full search vector quantization complexity for blocks with the initial block size used on MMP, by the total number of existing dictionary scales.

For volumetric blocks with $2^m \times 2^n \times 2^k$ pixels, the full search vector quantization complexity, using a codebook composed by S elements is given by $(2^m \times 2^n \times 2^k) \times S$. In Equation G.1, we have shown that the total number of scales for $2^m \times 2^n \times 2^k$ pixels blocks when a flexible partition scheme is used can be given by $((m+1) \times (n+1) \times (k+1))$. Thus, combining the two equations, we obtain:

$$\mathfrak{C}_{3\mathrm{D}-\mathrm{MMP}}(2^m, 2^n, 2^k) = (2^m \times 2^n \times 2^k) \times S \times ((m+1)(n+1)(k+1)), \tag{G.2}$$

for the computational complexity of a volumetric MMP with an initial block size of $2^m \times 2^n \times 2^k$ pixels, and three-dimensional flexible segmentation scheme.

Similarly to Equation B.16, the proof of Equation G.2 can also be done by induction. Once more, the formula clearly holds for blocks with size $(1 \times 1 \times 1)$, since the elements of the dictionary will be tested only once, that is:

$$\begin{aligned} \mathfrak{C}_{\text{3D}-\text{MMP}}(2^0, 2^0, 2^0) &= (2^0 \times 2^0 \times 2^0) \times S \times ((0+1) \times (0+1) \times (0+1) \\ &= S \end{aligned} \tag{G.3}$$

Using the inductive hypothesis, the formula holds for blocks of dimension $(2^m, 2^n, 2^k)$. For blocks of dimension $(2^{m+1}, 2^n, 2^k)$, the algorithm need to perform extensive optimizations of the two $(2^m, 2^n, 2^k)$ blocks, which compose the original $(2^{m+1}, 2^n, 2^k)$ block, plus the optimization of all the non-redundant nodes, which correspond to those from dictionary scales with dimensions $(2^{m+1}, 2^i, 2^j)$, with $(i = 0...n)$ and $(i = 0...k)$. Thus:

$$\begin{aligned} \mathfrak{C}_{\text{3D}-\text{MMP}}(2^{m+1}, 2^n, 2^k) &= 2 \times \mathfrak{C}_{3D-MMP}(2^m, 2^n, 2^k) \\ &\quad + \sum_{i=0}^{n} \sum_{j=0}^{k} 2^{(n-i)(k-j)} \times (2^{m+1} \times 2^i \times 2^j) \times S \\ &= 2 \times \Big((2^m \times 2^n \times 2^k) \times S \times ((m+1)(n+1)(k+1)) \Big) \\ &\quad + \sum_{i=0}^{n} \sum_{j=0}^{k} (2^{m+1} \times 2^n \times 2^k) \times S \\ &= (2^{m+1} \times 2^n \times 2^k) \times S \times ((m \times n + m + n + 1)(k+1)) \\ &\quad + (2^{m+1} \times 2^n \times 2^k) \times S \times (n+1)(k+1) \\ &= (2^{m+1} \times 2^n \times 2^k) \times S \times \\ &\quad (mnk + mn + mk + m + 2nk + 2n + 2k + 2) \\ &= (2^{m+1} \times 2^n \times 2^k) \times S \times ((m+2)(nk + n + k + 1)) \\ &= (2^{m+1} \times 2^n \times 2^k) \times S \times ((m+2)(n+1)(k+1)) \,. \end{aligned} \tag{G.4}$$

The induction for the other coordinates is entirely analogous.

The particular case where $k = 0$ corresponds to the 2-dimensional MMP-FP algorithm, so the simplification of Equation G.4 obviously results on Equation B.16:

$$\begin{aligned} \mathfrak{C}_{\text{3D}-\text{MMP}}(2^m, 2^n, 2^0) &= (2^m \times 2^n \times 2^0) \times S \times ((m+1)(n+1)(0+1)) \\ &= (2^m \times 2^n) \times S \times ((m+1)(n+1)) \,. \end{aligned} \tag{G.5}$$

This also allows us to perform a computational complexity comparison between the two-dimensional MMP-FP algorithm and 3D-MMP:

$$\mathfrak{C}_{\text{3D}-\text{MMP}}(2^m, 2^n, 2^k) = \mathfrak{C}_{MMP-FP}(2^m, 2^n) \times (2^k \times (k+1)). \tag{G.6}$$

Equation G.6 shows that the introduction of a new coordinate results on an exponential

increase on the computational complexity. This increase becomes much more relevant if a hierarchical prediction is used. All the prediction modes will be tested for each block dimension allowed for prediction, and as referred in Appendix B, there is no redundant nodes in the prediction level. Thus, all the combinations of block partitions need to be optimized on the prediction level.

G.2.2 3D-MMP dictionary design

The new block scales possibilities have a direct impact in the dictionary design. The first challenge is to deal with the existence of a multiscale dictionary. Two approaches are possible for a practical implementation of such a codebook:

- All the code-vectors are stored in a single dictionary, as well as the scale where it was originally created. While optimizing the representation of a given block of the input signal, the algorithm needs to perform a scale transform of each codeword before performing the match. This is also the case when testing the insertion of new codewords in the dictionary: the algorithm needs to perform a scale transform of each existing codeword, in order to avoid the replication or the insertion of similar codewords.

- Multiple copies of scaled versions of the codebook are stored in the memory, resulting in the existence of several sub-dictionaries. When the algorithm needs to perform a match for a given scale, or need to verify if no similar codewords already exist, it only needs to check the corresponding sub-dictionary. The scale transform is only applied to new created codewords, before determining their insertion on each scale.

The first approach requires less memory, as a single copy of each codeword is stored in the dictionary. However, performing the scale transform from each codeword before performing a match is a cumbersome task, which is computationally prohibitive. The second approach reduces significantly the computational complexity of the algorithm, but significantly increases the memory requirements, as the need of storing the several sub-dictionaries is imposed. This revealed however the best practical approach, as the processing power is the bottleneck for such a computationally complex algorithm.

This approach was adopted also for the proposed framework. However, the increase in the number of scales also results in an increase on the number of the sub-dictionaries which need to be stored. For this reason, despite the same trend verified for the 2D layout, where the increase in the dictionary size tends to increase the coding efficiency of the algorithm, most of our tests have been performed using a maximum size of 5000 elements per scale, instead of the 50000 used on [49].

Other dictionary parameters were directly inherited from the 2D version of the MMP still image coding algorithm [5]. For example, a new codeword is only inserted on scales where each corresponding dimension is half or twice that of the original scale, and the original scale where the block was originated is used as a context for the arithmetic encoder, exploiting the difference on probabilities of matching blocks becoming from different scales. Similarly to the 2D case, geometric transforms of each new codeword are also generated and inserted in the dictionary. These geometric transforms include the additive symmetrical, 90^o, 180^o and 270^o rotations, along each of the 3 axis.

The redundancy control tool proposed in [49] was also used in our implementation. In this case, a new codevector is only inserted in the dictionary if its position in the 3D space does not fall inside the volume defined by a hypersphere of radius d, centered at each of the previously existing codevectors. Experimental tests demonstrated that the same model for $d(\lambda)$ presented in [4] and traduced by Equation B.12, where d is the hypersphere value and λ the lagrangian operator, is also suited for the volumetric framework, that is:

$$d(\lambda) = \begin{cases} 5, & \text{if} \quad \lambda \leq 15; \\ 10, & \text{if} \quad 15 < \lambda \leq 50; \\ 20, & \text{otherwise.} \end{cases} \tag{G.7}$$

Using Equation G.7, the encoder is able to determine the value of the hypersphere of radius d based on the input parameter λ.

G.2.3 The use of a CBP-like flag

Such as for the case of MMP-video, described in Section D.3.4, the proposed framework also adopted the use of a CBP-like flag, to signal null residue patterns.

A binary flag is transmitted for every tree leaf, indicating if a given block is better represented by a zero pattern, or by using a dictionary code-vector. In the first case, the flag '0' is transmitted for the tree leaf, omitting the transmission of a dictionary index associated to the corresponding block. In the second case, the flag '1' is transmitted, followed by the index corresponding to the best code-vector found in the dictionary to represent the block.

A lagrangian cost function is used to determine the best representation for each block. The cost of the null residue pattern corresponds to sum of the energy of the residue block with the rate associated to the transmission of the flag '0', multiplied by the lagrangian operator λ. The cost of the non-null residue pattern corresponds to the sum of the distortion between the residue block and the optimal code-vector found for its representation, and λ times the sum of the rate corresponding to the transmission of the flag '1' and to the transmission of the best dictionary index, which represents the selected code-vector.

Such as on the case of MMP-video, this approach tends to increase the rate needed to encode non-zero patterns, but tends to decrease significantly the rate required to encode null residue patterns, which are expected to occur very often if the algorithm is able to generate accurate predictions. Despite the ability presented by the adaptive arithmetic encoder to adapt to the most frequent occurrence of null residual patterns, the use of a CBP-like flag considerably accelerates the time needed to adapt to this statistic, increasing the compression performance of the algorithm for most cases.

G.2.4 3D least squares prediction

In Section B.2.2, the least squares block-based prediction proposed in [46] was described. This method was adapted from [47] but some modifications were proposed in order to solve some causal neighborhood issues, which appear in a block-based layout. The use of fixed filter support and training window was proposed in [47] in a pixel-by-pixel basis, where all the pixels on the left and on the top of the one being encoded, as well as all the pixels from the previous frames are already available. This is not the case for a block-based encoder.

In a block-based layout, the left neighbors that belong to the same block of the pixel being predicted are not yet available. The method described in [46] suggested to use the predicted pixels' values for the non-available neighbors, instead of their reconstructed values. This approach allowed to solve the causal neighborhood issues and to maintain the prediction on a pixel-by-pixel basis, with minimal performance losses.

However, for pixels on the right frontier of a given block, the neighbors from the upper line located on the right side of the pixel being predicted are available, as they belong to another block which has not yet been encoded. Thus, for these cases, both the filters support and the training window are modified, in accordance to Figures B.6-b and B.7-b, in order to only include causal neighbors.

The causality issues become much more relevant in a volumetric layout. As the method proposed on [47] suggested the use of LSP for spatiotemporal prediction, it can be seen as a generic case of a three-dimensional compression layout, where the third dimension is the temporal axis. Thus, in order to exploit the temporal redundancy, both the filter support and the training window must only include pixels from previous frames which have already been encoded.

Therefore, for our particular compression scheme, the three-dimensional block-based approach implies that some of those previous pixels may not have been encoded yet. Without losing the generality, one may refer the slices along the k_3 axis as frames, in order to simplify the comparison with the case from [47], where k_3 is the temporal axis. Thus, pixels from previous frames which belong to the same spatiotemporal block as the pixel being encoded, have also not yet been encoded. Then, they need to be replaced by their

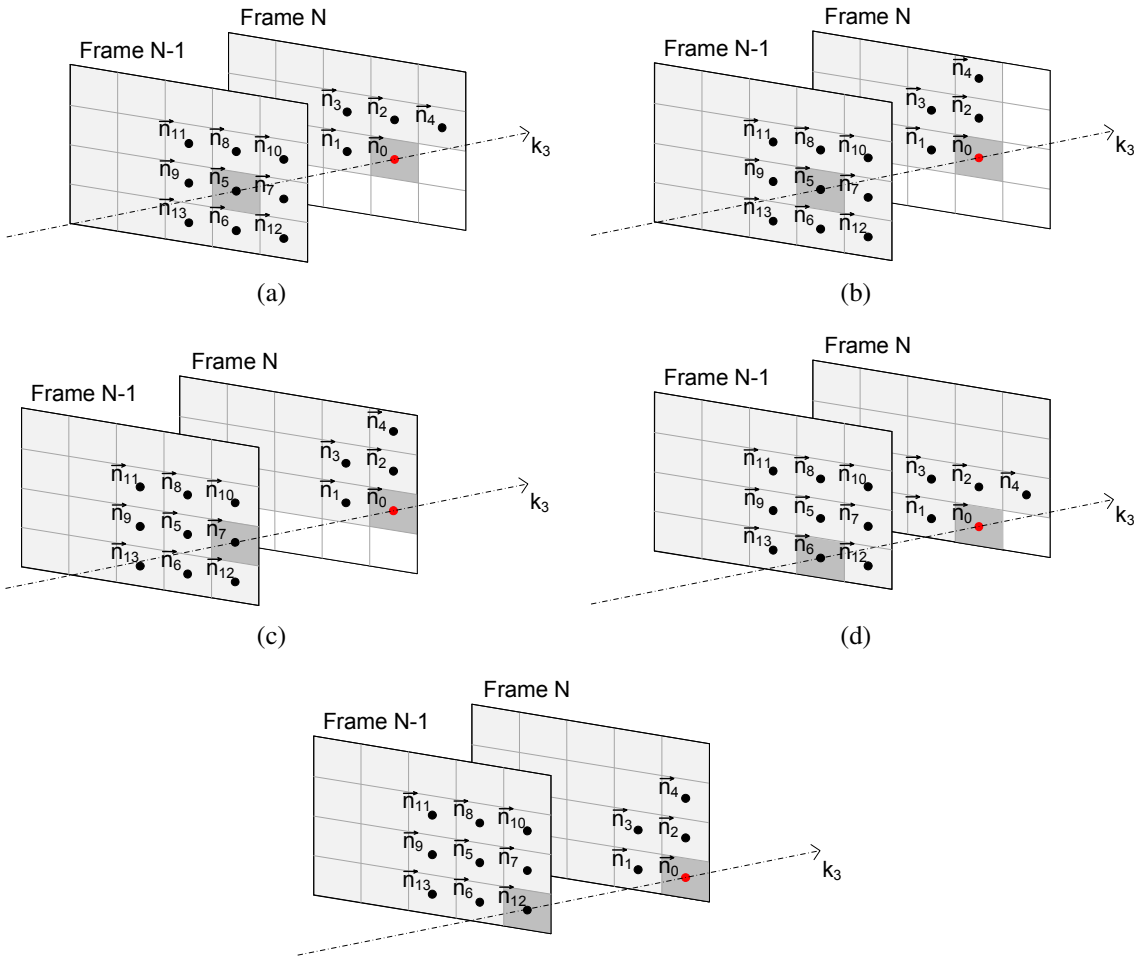


Figure G.2: Spatiotemporal neighborhood used on (a) default (b) rightmost column of first layer of the block (c) rightmost column subsequent layers of the block (d) bottommost row (e) bottom-right corner.

predicted values. Pixels located on the block boundary may also not have the temporal neighbors on its right, as these pixels can belong to the next block to be encoded. This way, the modifications performed on the shape of the filter support and the training region need to be extended to the additional dimension.

As a result, we adopted a filter support similar to the one proposed on [47], as illustrated on Figure G.2a, with its support being subjected to some modifications, in order to be adapted to the block-based approach constraints.

Similarly to the case of [47], the four nearest neighbors in space, plus the nine closest in time [130] are used in the support of a thirteen order linear predictor. Note that the pixel ordering does not affect the prediction result, but obviously the same order must be used for all the pixels from the training window. The choice of such filter support relies in the little-motion assumption, as this approach assumes that the correspondence for the pixel being predicted, $X(\vec{n}_0)$, with $\vec{n}_0 = (k_1, k_2, k_3)$, is likely to be located within a 3×3 pixels window of the previous frame $(k_3 - 1)$, centered in (k_1, k_2). In [131], the authors

proposed to use pixels from the two previous frames, on a least squares based predictor selection for lossless video compression framework. However, the use of pixels from two previous frames is of little utility to estimate the pixels values, as a large window will be needed to comprise the pixels correspondence, for most motion cases.

As previously referred, in the block right boundary, the pixel $\vec{n}_4$ belongs to the next block to be encoded, and consequently is not yet available. In these cases, the position of $\vec{n}_4$ is displaced, as illustrated on Figure G.2b. The pixel located on the position (k_1, k_2-2) is used instead of the pixel from (k_1+1, k_2-1). Furthermore, in this situations, the pixels $\vec{n}_7$, $\vec{n}_{10}$ and $\vec{n}_{12}$ will also belong to the next block if the pixel being predicted is located in the second layer of pixels along k_3. Thus, in these situations, the temporal neighborhood is displaced 1 pixel to the left, as illustrated on Figure G.2c. This is also the case for the right boundary of the frame, where $\vec{n}_4$, $\vec{n}_7$, $\vec{n}_{10}$ and $\vec{n}_{12}$ from Figure G.2a are not available.

Similarly, on the bottom edge of the blocks and the bottom boundary of the frames, pixels $\vec{n}_6$, $\vec{n}_{12}$ and $\vec{n}_{13}$ may also not be available to be used on the filter's support. For these cases, the neighborhoods from Figure G.2a and Figure G.2c are displaced to the top, resulting in Figure G.2d and Figure G.2e, respectively.

For pixels on the first slice along the k_3 axis, the absence of the references from the previous slice is solved by using only the spatial neighbor pixels in the filter support, reducing the order of the linear predictor to four.

The prediction for a given pixel $X(\vec{n}_0)$, with $\vec{n}_0 = (k_1, k_2, k_3)$, can be then obtained using the equation:

$$\hat{X}(\vec{n}_0) = \sum_{i=1}^{N} a_i X(\vec{n}_i), \tag{G.8}$$

where $\vec{n}_i$ with $i = 1, 2, ..., N$, are the spatiotemporal causal neighbors presented on Figures G.2, and $\vec{a} = [a_1, ..., a_n]^T$ is the prediction coefficient vector field. Assuming the Markov property, the optimal prediction coefficients $\vec{a}$ are trained from a local causal neighborhood in space-time, such as proposed in [47].

The volumetric causal training neighborhood we have adopted is a three-dimensional extension of the spatial training window proposed in [46]. Figure G.3a presents the generic case, and Figure G.3b presents the training region used for pixels located on the block's right boundary, where the pixels on the right of the one currently being predicted are not yet available.

Similarly to the two-dimensional case, all the samples from the training region are placed into an $M \times 1$ column vector $\vec{y}$, with M being the number of pixels on the training region. If we put the N causal neighbors for each training sample (13 for the case presented in Figure G.2) into a $1 \times N$ row vector, then all training samples generate a data matrix C, of size $M \times N$.

It is important to note that the changes in the filter support for the referred exception cases implies that the same support is used for all the pixels in the training region. Fur-

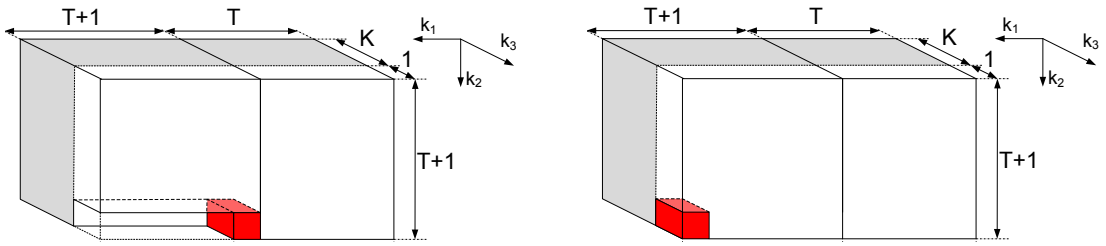


Figure G.3: Spatiotemporal training region (a) standard (b) rightmost column.

thermore, it can be seen that if a filter support is causal for a given pixel, it will also be the case for all pixels inside the defined training region.

The optimal prediction coefficients $\vec{a}$ can be determined by solving the following least squares problem:

$$\min(||\vec{y} - \mathbf{C}\vec{a}||^2), \tag{G.9}$$

which has the well-known closed form solution:

$$\vec{a} = (\mathbf{C}^T\mathbf{C})^{-1}(\mathbf{C}^T\vec{y}). \tag{G.10}$$

While encoding the first slices of a given volumetric signal, there is not only the need of adapting the filter's support but also the training region. As previously referred, only the spatial neighbors are used while encoding the first slice resulting in a fourth order linear predictor. Furthermore, for the first slice, there are no temporal neighbors available for the training region ($K = 0$), so the training region only comprises pixels from the current frame. Thus, the least squares predictor is in this case a spatial predictor, such as in [46, 132]. For the second slice, all the pixels from the filter's support become available. In this case, the proposed method uses the thirteen order predictor, but the training region remains confined to only the current slice, with $K = 0$. For subsequent K_3 slices, the value of K is gradually increased, until reaching the maximum defined value.

G.2.5 3D Directional prediction

H.264/AVC [51, 51] adopted a set of directional prediction modes to exploit Intra-frame spatial redundancy. This concept was further extended for larger 64×64 pixels blocks on the upcoming HEVC coding standard [16], which increased the possibilities of directional prediction from 8 to up to 33 directional modes. Directional prediction modes generally achieve good results on most natural images, because the intensity field tend to be constant along the edge orientation. Thus, if the directional prediction is able to match the edge orientation, it will be able to generate an accurate prediction block for most cases.

In some volumetric signals, intensity fields present a similar behaviour in the three-

dimensional domain. For example, in Section G.3.1, we will demonstrate the similarity between the behavior of the motion trajectories in space and edge contours in the spatiotemporal domain, for the case of video signals. Based on such observations, we may assume that prediction techniques successful in exploiting the spatial redundancy, may also be suited to predict intensity fields trajectories on a volumetric framework. Thus, the concept of directional prediction for 3D blocks was extended for the proposed framework.

Consider a generic 3D block $X^l(k_1, k_2, k_3)$ with $N \times M \times K$ pixels, where k_1 and k_2 are the spatial coordinates and k_3 is assigned to the temporal axis. Consider also a bi-dimensional directional vector $\vec{v} = (v_1, v_2)$, with coordinates v_1 and v_2 corresponding to the vector components along k_1 and k_2 respectively. Assuming the reconstructed frame from the temporal position immediately before the the frame being encoded is available, a directional 3D prediction block $\hat{X}^l(k_1, k_2, k_3)$ can be defined as:

$$\hat{X}^l(k_1, k_2, k_3) = X^l(k_1 - v_1, k_2 - v_2, k_3 - 1). \tag{G.11}$$

In other words, if we consider that the block $X^l(k_1, k_2, k_3)$ is sliced along the k_3 axis, each slice of the prediction block will assume the values of the same coordinates of the previous frame, displaced by a vector $\vec{v}$.

Note that motion estimation is itself a particular case of this approach, where $K = 1$ and a vector is required for each slice. Using motion estimation, the prediction for each block is a portion of the previously encoded frames, displaced by a given distance and represented by a motion vector. Thus, the volumetric directional prediction can be seen as an extension of motion estimation, where several frames can be motion estimated by the same vector.

Intuitively, one may argue that for linear trajectories, this approach should allow to motion estimate several frames using a single vector. Furthermore, for the case of non-linear trajectories, it can be considered that the trajectory remains approximately linear for sufficiently small intervals, so the method can still be useful. Note that the hierarchical prediction adopted on MMP allows the prediction to be successively segmented along any of the coordinate axis, including k_3, so the algorithm is able to approximate non-linear motion trajectories with linear segments every time this is beneficial from a rate-distortion point-of-view. In the limit, successive prediction blocks with $K = 1$ will be encoded, each using its own bi-dimensional directional vector, in a particular case where the proposed approach converges to the traditional motion estimation.

However, the block-based approach arises some limitations in relation to the causality of the previous frames. As each block may comprise several frames, which are encoded together, portions of the previous frame may belong to other blocks that are not still encoded when the prediction for a pixel is performed. This problem is solved by using references from the closest available frame, instead of the frame immediately before of

the one being predicted, that is $(k_3 - 1)$ in Equation G.11 may be replaced by a more generic $(k_3 - p)$, with $p > 1$.

Lets start with the simpler case, where $\vec{v} = (0, 0)$, and consider a given block as a set of slices along the k_3 axis, varying in k_1 and k_2. Consider that the first corner pixel of the block is located on coordinates (K_1, K_2, K_3), so the pixels from the first slice will have generic coordinates (k_1, k_2, K_3). The prediction generated for a pixel from the slice (k_1, k_2, K_3) will assume the values from the contiguous pixels located on $(k_1, k_2, K_3 - 1)$. These pixels belong to a previous block which was already entirely encoded.

For the second slice $(k_1, k_2, K_3 + 1)$, a generic pixel should use as reference the pixels (k_1, k_2, K_3), which belong now to the same block that is being encoded. More precisely, the reference pixels are located on the previously processed slice of the block and thus, the predicted pixels for the current slice will be equal to those from the previous layer. In other words, the algorithm performs the padding of the pixels from the frontier with the block being encoded, to the entire block.

Nevertheless, a causality problem arises if any of the vector components (v_1 or v_2) is negative. In this case, the prediction for a pixels located on (k_1, k_2, K_3) (from the first slice) will assume the values of the displaced correspondence in the previous slice, more precisely $(k_1 - v_1, k_2 - v_2, K_3 - 1)$. In the first slice, these pixels should be available as they belong to a previous block already encoded, but for the the second layer the reference pixels $(k_1 - v_1, k_2 - v_2, K_3)$ may not be still available (note than v_1 and v_2 are negative, so the pixels are on a non-causal region of the image, corresponding to the right or bottom relatively to the pixels being predicted). The same applies for the subsequent slices of the block.

This problem is overcome by moving the reference pixels to the closest temporal pixels available for each layer, so that pixels from the K_3 frame are used to predict all slices from the block. For the first layer, the pixels from (k_1, k_2, K_3) will be predicted using the values from pixels $(k_1 - v_1, k_2 - v_2, K_3 - 1)$. For the second layer pixels $(k_1, k_2, K_3 + 1)$, instead of using the pixels from $(k_1 - v_1, k_2 - v_2, K_3)$, which are not all available, the prediction is generated using the pixels $(k_1 - 2 \times v_1, k_2 - 2 \times v_2, K_3 - 1)$. The same approach applies to each of the K layers on the block, with the n^{th} layer being predicted using the pixels in $(k_1 - n \times v_1, k_2 - n \times v_2, K_3 - 1)$. The only constraint of such approach is that $K_1 + n \times v_1 \leqslant W$ and $K_2 + n \times v_2 \leqslant H$, with W and H being the signal's width and height, respectively.

Note that these causality problems do not arise when the vector components are positive. In this case, the prediction only needs causal reference pixels, located on the left and top of the pixels being predicted. These pixels may be already completely encoded, so that their reconstructed value is used on the prediction, or in some cases, only the prediction value can be available for those pixels, with the residue for those pixels not encoded yet. In this last case, predicted values are used to generate predictions for further layers,

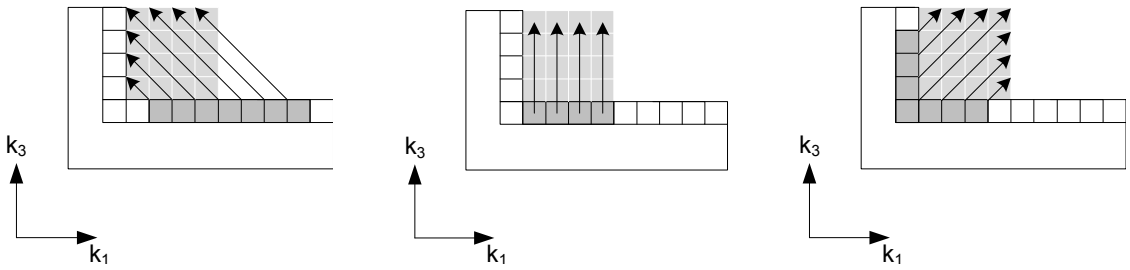


Figure G.4: Diagram of directional prediction along a single coordinate (a) $v_1 < 0$ (b) $v_1 = 0$ (c) $v_1 > 0$.

such as proposed in [46] for the spatial case. The unique constraints for positive vectors are $K_1 - v_1 \geqslant 0$ and $K_2 - v_2 \geqslant 0$.

The directional prediction is graphically represented on Figure G.4. In order to improve the figure's legibility, only one of the spatial coordinates is represented, but the same approach is extendable to the other spatial coordinate. The light grey pixels use the dark grey ones as references for the directional prediction.

Figure G.4a shows the case where the vector norm along k_1 is negative (v_1<0). In this case a given pixel being predicted will assume the value of a pixel from the closest causal temporal reference frame, which has been totally encoded, located on its right. When v_1=0 (Figure G.4b), the prediction for each pixel assume the value of the corresponding pixel on the closest temporal reference neighborhood. When v_1<0, the block based approach allows to trade the reference pixels from the totally reconstructed frame, for some spatial neighbors which are already available and closer to the pixel being predicted, as shown on Figure G.4c.

In our first approach, the algorithm tested all the possible directional vectors on a pre-established interval, and the direction which minimized the resulting residue block energy was chosen. However, it is important to notice that unlike transform based algorithms, there is no strong correlation between the block's energy and the amount of bits needed to encode it. A given residue block can present a high energy but being very similar to an existing codeword, and another residue block can present a low energy but being not able to find a proper match. Thus, the lowest energy approach is sub-optimal in a rate-distortion point-of-view, but is considerably less cumbersome than computing the MMP cost for each residue block. Furthermore, it can be seen that this approach only results in marginal performance losses.

The chosen directional vector is then encoded using an adaptive arithmetic coder [48], with independent probability models for each component of the vector and for each block level. Thus, the best vector is selected using a lagrangian cost function [44] which weights the energy of the resulting residue block, over the rate required to encode the corresponding vector.

This approach revealed however some inefficiencies while encoding directional vectors, as it does not take into account the directional behavior from neighbor blocks. For example, in the case of video compression, it is a well known fact that for some motion classes, the motion direction of a given block is strongly correlated with those from other blocks located on its proximity. This way, the optimal vector can generally be successfully estimated using the neighbor block's vectors. This correlation is successfully exploited by many hybrid video codecs, such as [45].

This led us to perform a direction estimation based on the block's causal neighborhood. A directional vector is estimated based on a template located on the block's causal neighborhood, and the algorithm chooses between using the estimated vector or transmitting the best determined vector. The choice is performed once more using a lagrangian cost, which weights the rate required for each case, and the residue's energy associated to each option. If the estimated vector is considered to be suited for the block, a single flag 0 is transmitted to the decoder, which is able to perform the same estimation in order to replicate the vector. Otherwise, the flag 1 is transmitted, followed by each of the two vector's components.

G.2.6 H.264/AVC based prediction modes

Additionally to the volumetric least squares prediction mode described on Section G.2.4, and to the volumetric directional prediction presented on Section G.2.5, the proposed framework adopted some complementary prediction modes. Those resulted from 3D extensions of the prediction modes used by H.264/AVC [45] and the MMP based still image encoder [5].

One of these modes is the volumetric extension of the most frequent value (MFV), a substitute on MMP-based still image encoders of the DC mode used by H.264/AVC. In this case, a homogenous prediction block is generated, with a pixel intensity equal to the most frequent values among the causal neighbors of the block being predicted. In the volumetric framework, the reference pixels are located on the three planes α, β and γ, on causal boundaries of the block being predicted, as shown on Figure G.5.

This prediction mode revealed to be particularly useful for the first slices along the k_3 axis, for which the least squares and the directional predictions are not still available, due to the lack of previous reference slices. For the first block of the input data, for which no prediction references are available, a default uniform prediction block with a pixel intensity level of 128 is generated by this prediction mode.

The other prediction modes presented on Figure B.5 were also adapted for volumetric blocks. Considering that a volumetric block can be sliced along the k_3 axis, resulting in planes which vary along k_1 and k_2, those prediction modes are applied to each of these resulting layers. It is important to notice that this situation can be seen as a particular

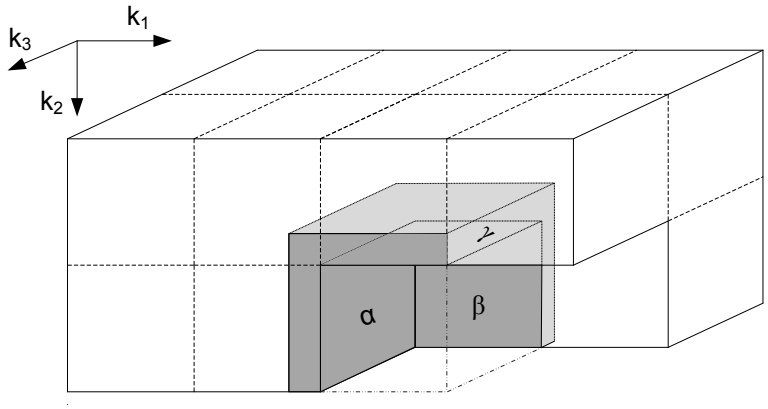


Figure G.5: Block neighborhood.

case of the directional prediction used in [15], where only the information present on the current slice is needed.

All the prediction modes presented in this section, as well as those described in Sections G.2.4 and G.2.5, are applied to each block, when the corresponding reference pixels are available. The algorithm then selects the prediction mode which minimizes the lagrangian cost function, that weights the rate associated to the transmission of each mode, over the energy of the resulting residue blocks.

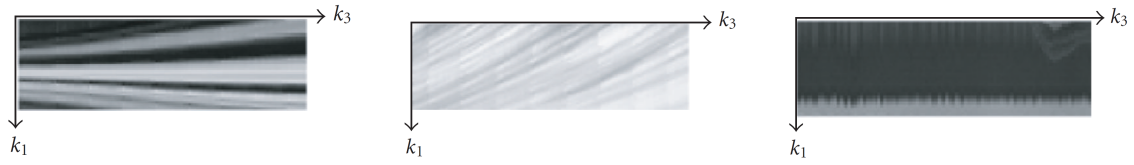
G.3 3D-MMP for video compression

The algorithm presented in the previous section is a generic approach to volumetric signals compression. In this section, supported on the knowledge gathered from previous investigations related to video signals compression, we propose an optimized video compression architecture.

G.3.1 The edge contour/motion trajectory duality

The duality between edge contours in 2D image and motion trajectories in video sequences, has already been well described in the literature [47]. If one replaces one of the spatial coordinates by the temporal axis, it can be seen that the result is a 2D signal composed by parallel rows or columns, which presents many characteristics common to those observed on natural images. Thus, we may argue that if we intentionally confuse spatial coordinates with the temporal axis, the resulting signal is dual to a video sequence consisting of parallel slices in the temporal domain.

Particularly, if we take as an example some edge from a given natural image, we may observe that the intensity field tends to be constant along the edge orientation. Similarly, if a given video sequence is conceived as a volumetric signal, the motion trajectories in the 3D space will also be characterized by iso-intensity levels sets in that continuous space.



(a) (b) (c)

Figure G.6: Examples of spatiotemporal under camera (a) zoom (b) panning (c) jittering.

Thus, conceptually, the contour of an edge in 2D is equivalent to a motion trajectory in the 3D space, and such duality suggests that the redundancy reduction tools useful for exploiting the geometric constraint of edges, lend themselves to be useful while exploiting motion-related temporal redundancy as well.

As we consider more general motion, such as camera jittering, rotation or zoom, motion trajectory of an object becomes more complicated curves in the spatiotemporal slices. However, locally within a small spatiotemporal cube, the flow directions of motion trajectory is still approximately constant.

Figure G.6 presents some examples that help to illustrate this concept for several motion types in a spatiotemporal slice. Figure G.6a shows a portion of a spatiotemporal slice from a video sequence subjected to a camera zoom. The object contours are augmenting their dimensions on the scene along the temporal (k_3) axis, due to the camera zoom. The iso-intensity levels are visible along those edge contours. Figure G.6b shows a spatiotemporal slice obtained from a video sequence subjected to camera panning. The flow-like pattern visible on the figure also shows the motion trajectories along which the iso-intensity constraint is satisfied. In Figure G.6c, a spatiotemporal slice from a video sequence subjected to camera jittering is presented, demonstrating once more the iso-intensity levels defined along the edge regions.

G.3.2 Video compression architecture

In our first approach, the input video sequence was processed sequentially in groups of N frames. Each group of N frames was then encoded in a $N \times N \times N$ block basis, using a raster scan order, as illustrated in Figure G.7.

Considering k_1 and k_2 the spatial coordinates, where $1 \leq k_1 \leq H$ and $1 \leq k_2 \leq W$, and t is the temporal coordinate, this approach corresponds to process the temporal coordinate as a generic k_3, which results in a 3D volumetric signal $X(k_1, k_2, k_3)$. Thus, a 3D hierarchical prediction scheme is used to simultaneously exploit the spatiotemporal correlation of the input signal. The generated 3D residue is directly compressed using a 3D extension of the MMP algorithm with a flexible partition scheme, as described in Section G.2.1. Note that both spatial and temporal references are in this case available to generate the 3D prediction for each block.

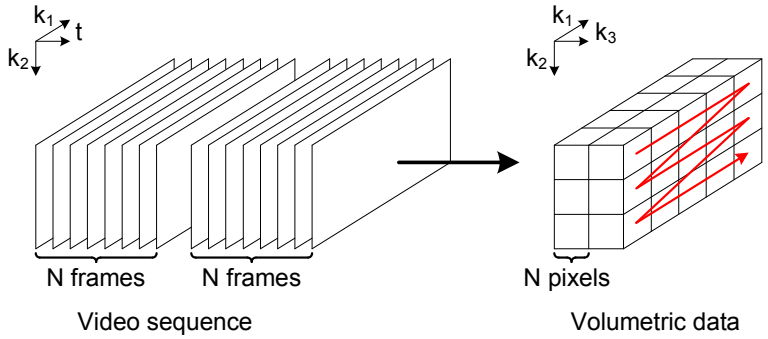


Figure G.7: Sequential codec architecture.

A strong correlation can be established between each group of N frames and the GOP on generic hybrid video codecs. Similarly, both are the minimum temporal unit which can be decoded independently without the need of decoding previous or future segments of the video sequence. Thus, if both the MMP dictionary, the arithmetic coder statistics and the temporal prediction references are reset periodically, the proposed framework can provide the same aleatory access presented by hybrid codecs, which is essential for most practical applications. Consequently, N will determine the minimum GOP size, but it is still possible to reset this information for any multiple of N frames.

Such as for hybrid video codecs, increasing the minimum temporal coding unit will impact negatively on the ability to randomly access the video sequences, and will increase the amount of memory needed to store the temporal references. However, it usually tends to increase the rate-distortion performance of the video compression algorithm, hence the existence of a richer set of temporal references generally results on more accurate predictions. Furthermore, the arithmetic coder has more time to adapt to the signals statistical characteristics, and previous works have suggested that the increase in the input signal length tends to present a positive impact on the approximation power of the MMP dictionary, as a richer set of code-vectors. Thus, the choice of the minimum temporal coding unit is a trade-off between random access ability and computational resources, for compression efficiency.

Typically, for practical applications, it is possible to use a GOP size corresponding to a multiple of N. In such case, when the second group of N frames is encoded, the temporal references from the previous group are available to be used while generating the prediction for the block being coded.

It is also important to notice that despite the adoption of cubic blocks with $N \times N \times N$ pixels, illustrated in the example, this approach can be generalized to any block dimension with $N \times M \times K$ pixels.

However, a second approach for video compression revealed a higher rate-distortion performance. Instead of sequentially processing groups of successive frames from the video sequence, each group of frames is composed by alternate frames of the video se-

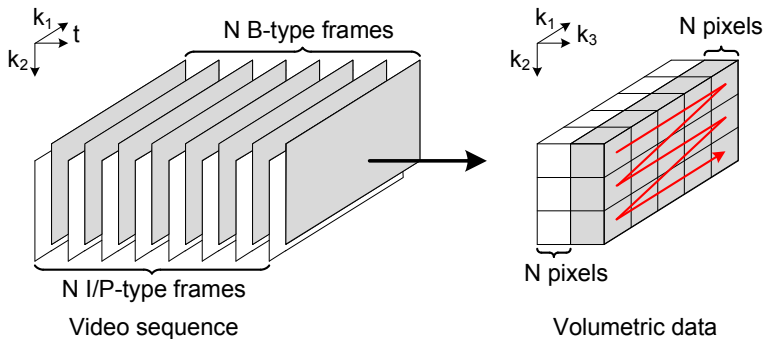


Figure G.8: Hierarchical codec architecture.

quence. Figure G.8 illustrates the case where the first group of frames comprises only by the even frames of the video sequence, while the second group is composed only by the odd frames. In this case, the frames encoded in the first group can be used as prediction reference for the frames from the second group. As each frame from the second group is temporally located between two frames already encoded, both past and future neighbors can be used to generate its prediction, which results in more accurate predictions for these frames.

Note the similarity of this approach with the I/P and B slices compression on hybrid video codecs. We may consider that the frames from the first group of frames are similar to the I/P slices from hybrid video codecs, where only spatial or past references are used while predicting them, while the frame from the second group are similar to B frames, for which a bi-prediction using past and future references is possible. For that reason, blocks from the first group of frames will be referred in the future as I/P blocks, while blocks coded using the bi-predictive approach will be referred as B blocks.

The practical impact of such approach is similar to the one observed in hybrid video codecs. The compression efficiency while compressing I/P type blocks tend to degrade relatively to the case where simultaneous frames are encoded. This can be explained by the lower temporal correlation that results for the higher temporal distance between the frames being encoded. However, this performance degradation is compensated while encoding B type blocks. The high amount of temporal references allied to spatial references allow to generate accurate predictions on most cases, resulting in low energy residues, that can frequently be discarded, such as in B slices on hybrid codecs. In other words, the increase on the compression efficiency for B type blocks is more significant than the slight performance loss shown for I/P type slices.

It is also important to notice that such approach allows a straightforward implementation of a temporally scalable video codec. By simply discarding B type blocks, it is possible to decrease the frame rate if needed. Furthermore, this approach can be extended hierarchically, creating several layers of B type frames, and allowing several levels of temporal scalability.

G.3.3 3D least squares prediction for video compression

The adoption of a hierarchical codec architecture resulted in the creation of two class of blocks, referred to as I/P-type and B-type blocks.

For the case of I/P-type blocks, the application of the least squares prediction method is a straightforward application of the method described in Section G.2.4. The unique difference is that in this case, the temporal neighbor is not the previous closest frame, but the previous frame of the same type.

It can be expected that this approach results in a lower prediction accuracy for the LSP method, due to the smaller amount of redundancy in the temporal information, but this architecture allows to improve the prediction for B-type frames. In this case, each group of frames encoded using I/P-type blocks can be used as prediction reference for the corresponding frames encoded using B-type blocks. Each of the frames from the B-type block, with the exception of the last of the N frame which composes the block, have their past and future frame available, and thus, the LSP filter's support can be modified in order to take advantage from this additional information.

Nine reference pixels from the neighbor future frame were included in the filter's support, resulting in the extension of the least squares prediction to a twenty two order bi-predictive implicit prediction scheme. Similarly to the past frame reference pixels, the future reference pixels are locate within a 3×3 pixels window centered in (k_1, k_2). Figure G.9a illustrates the resulting filter support for B-type blocks. Note that, in this case, an order twenty two linear predictor is used.

Similarly to the generic case presented in G.2, in a block right boundary, the pixel $\vec{n}_4$ belongs to the next block to be encoded and thus cannot be used in the filter support. Just like the previous case, $\vec{n}_4$ is displaced from $(k_1 + 1, k_2 - 1)$ to $(k_1, k_2 - 2)$, as illustrated in Figure G.9b. Note that for B-type frames, both the past and future frames were already encoded as P frames, so $\vec{n}_7$, $\vec{n}_{10}$, $\vec{n}_{12}$, $\vec{n}_{16}$, $\vec{n}_{19}$ and $\vec{n}_{21}$ are available to be used on the predictor. Consequently, there is no need to modify the filter support.

An exception can be found in the image's right boundary. In this case, the pixels from the past and previous frames are displaced to the left, as shown on Figure G.9c. In the bottom boundary of the frame, the past and future pixels also need to be displaced to the top, resulting in the cases illustrated on Figure G.9d and Figure G.9e, respectively. Figure G.9d corresponds to a generic bottom boundary pixel, and Figure G.9e to a bottom pixel also located in a block's right frontier.

The absence of a future reference for the pixels belonging to the last frame from a group encoded using B-type blocks is solved by using the same order thirteen predictor presented in Section G.2.4, which is used also for I/P-type frames.

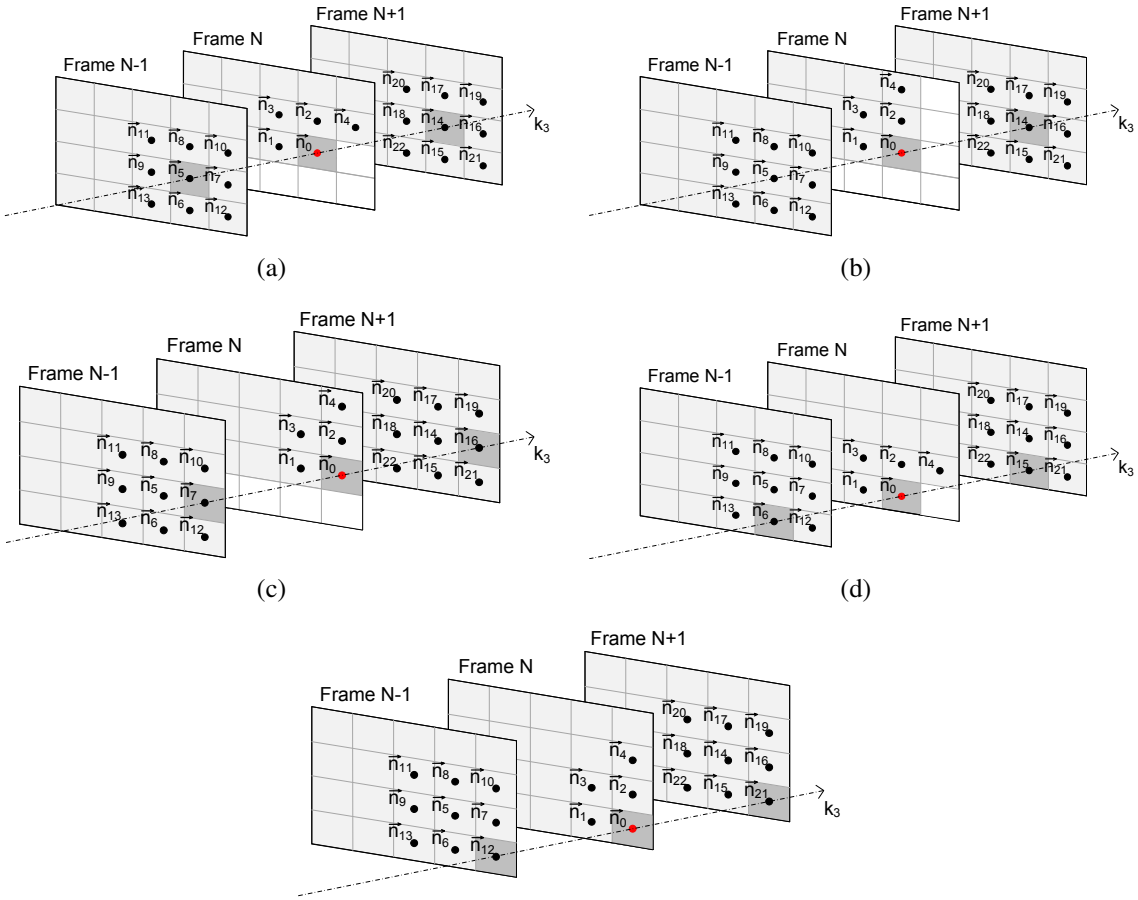


Figure G.9: Spatiotemporal neighborhood for B-type frame pixels (a) default (b) rightmost column of first layer of the block (c) rightmost column subsequent layers of the block (d) bottommost row (e) bottom-right corner.

G.3.4 3D directional prediction for video compression

For the case of B-type blocks, it is possible to take advantage from the closest references from previously encoded I/P-type frames, just like described for the case of the LSP prediction mode. The alternate encoding order means that a reference frame located between each of the B-type frames has already been coded and is thus available to be used as the closest reference for the prediction. Figure G.10 schematizes the references used for the directional prediction of the B-type blocks.

The case where the vector norm along k_1 is negative (v_1<0) is presented in Figure G.10a. Similarly to the case presented on Figure G.4a, the used references are located on the closest frame already reconstructed. Figure G.10b represents the case where v_1=0, which is similar to a block copy on the traditional motion estimation approach. Figure G.10c represents the case where v_1<0, where similarly to the case presented on Figure G.4c, temporal references are exchanged by closer spatial references.

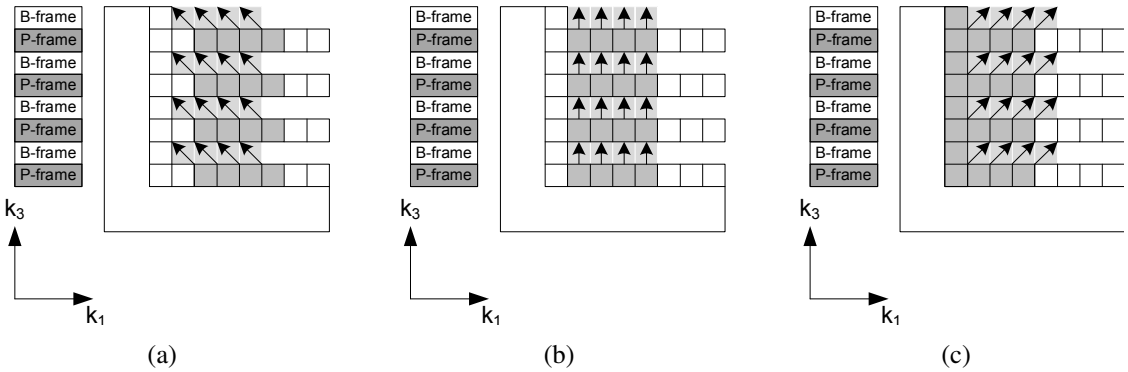


Figure G.10: Diagram of directional prediction for B frames, along a single coordinate (a) $v_1 < 0$ (b) $v_1 = 0$ (c) $v_1 > 0$.

G.4 Experimental results

This section presents a performance evaluation of the proposed framework, when used for video compression purposes. The experimental results obtained with the proposed method are compared with those from the JM17.1 H.264/AVC reference software.

We adopted the same set of parameters used in Appendix D for H.264/AVC, which included a GOP size of 15 frames with an *IBBPBBP* pattern, at a standard frame-rate of 30fps. H.264/AVC was operating at the high profile, with the RD optimization and the use of Intra MB in inter-predicted frames enabled. The context-based adaptive arithmetic coder (CABAC) was also adopted, while disabling the error resilience tools and the weighted prediction for B frames. The ME was performed using the Fast Full Search algorithm, with ± 16 search range from 5 reference frames.

The tests were performed using the variable bit rate mode for H.264/AVC, and the QP parameter was also set separately for the I/P and B slices [83]. Four distinct combinations of QP values were used, namely 23-25, 28-30, 33-35 and 38-40, the sames values used on the experimental results presented in Appendix D.

For 3D-MMP, we adopted $8 \times 8 \times 8$ blocks, in order to restrict the computational complexity, with a maximum dictionary size of 5000 elements per scale ($8 \times 8 \times 8$ blocks result in a total of 64 different scales). The hierarchical frame architecture was adopted, intercalating one B-type and one I/P-type frame, sequentially. Such as for H.264/AVC, spending more bits while encoding the I/P slices revealed to be beneficial, as it results in better predictions for B slices [83], so the λ used while coding the B-type blocks is set to be 50% larger than the one used for the I/P-type blocks.

In order to obtain rate-distortion points in the same bitrate range than those obtained with H.264/AVC, four distinct combinations of λ values where used in the experimental tests, namely 20-30, 75-112, 200-300 and 500-750, respectively for the I/P and B-type blocks.

The proposed method uses the same dictionary redundancy control technique rule

proposed in [49], defined by Equation G.7, and new blocks are inserted in scales which each dimension corresponds to half or double the respective dimension of the original scale where the block was created.

The use of the CBP-like flag was enabled, but only for the B-type blocks. Similarly to the case of MMP video, discussed on Appendix D, the CBP-like flag are also not beneficial when used for I/P-type blocks, for basically the same reasons. The CBP-like flag makes the transmission of null residue patterns much more attractive from a lagrangian cost point-of-view, resulting in a rate-conservative representation of the input signal. This rate-conservative representation results on reference blocks encoded with higher distortions, and consequently, decreases the quality of the predictions for the B-type blocks. In other words, unlike the case of B-type blocks, the local choice of representing I/P-type blocks with higher distortion propagates that distortion thought the prediction of subsequent blocks, and thus have a negative impact on the codec's overall compression efficiency. Furthermore, as more residue blocks tend to be encoded using null patterns when the CBP-like flag is used, the number of new patterns generated through concatenation decreases, conditioning the dictionary growing process and increasing the time needed by the algorithm to adapt to the local signal's statistical characteristics.

The directional prediction is performed using vectors with integer precision, with each component being restricted to the interval [-4;4].

After encoding each group of eight frames, the deblocking filter proposed in Appendix F is applied in the reconstructions. The filtering parameters τ and s are fixed as 32 and 100, respectively, and α is exhaustively optimized over a set of eight pre-established values, which includes 0, 0.05, 0.08, 0.10, 0.12, 0.15, 0.17 and 0.20. This optimization is performed targeting the maximization of the PSNR from the reconstructions, and the selected values are appended to the final bitstream, after being encoded using an adaptive arithmetic encoder.

Figures G.11 to G.13 present the average PSNR of the first 64 frames *vs.* the global bitrate, for each colour component of three different CIF video sequences: Akiyo, which presents slow natural motion, Container which presents a uniform translational motion, and Coastguard, which presents several types of motion, including zoom, pan and jittering. The results are summarized in Table G.1, which also presents the BD-PSNR [84] for each colour component, reflecting the average PSNR gain of the proposed method relatively to JM17.1.

As one can see, the proposed method is able to outperform the state-of-the-art H.264/AVC for the sequence Container. This video sequence presents a uniform translational motion, which is efficiently predicted using the proposed directional prediction mode. Thus, the proposed algorithm is able to predict several frames with the same vector, achieving a more efficient representation than that obtained using the traditional ME.

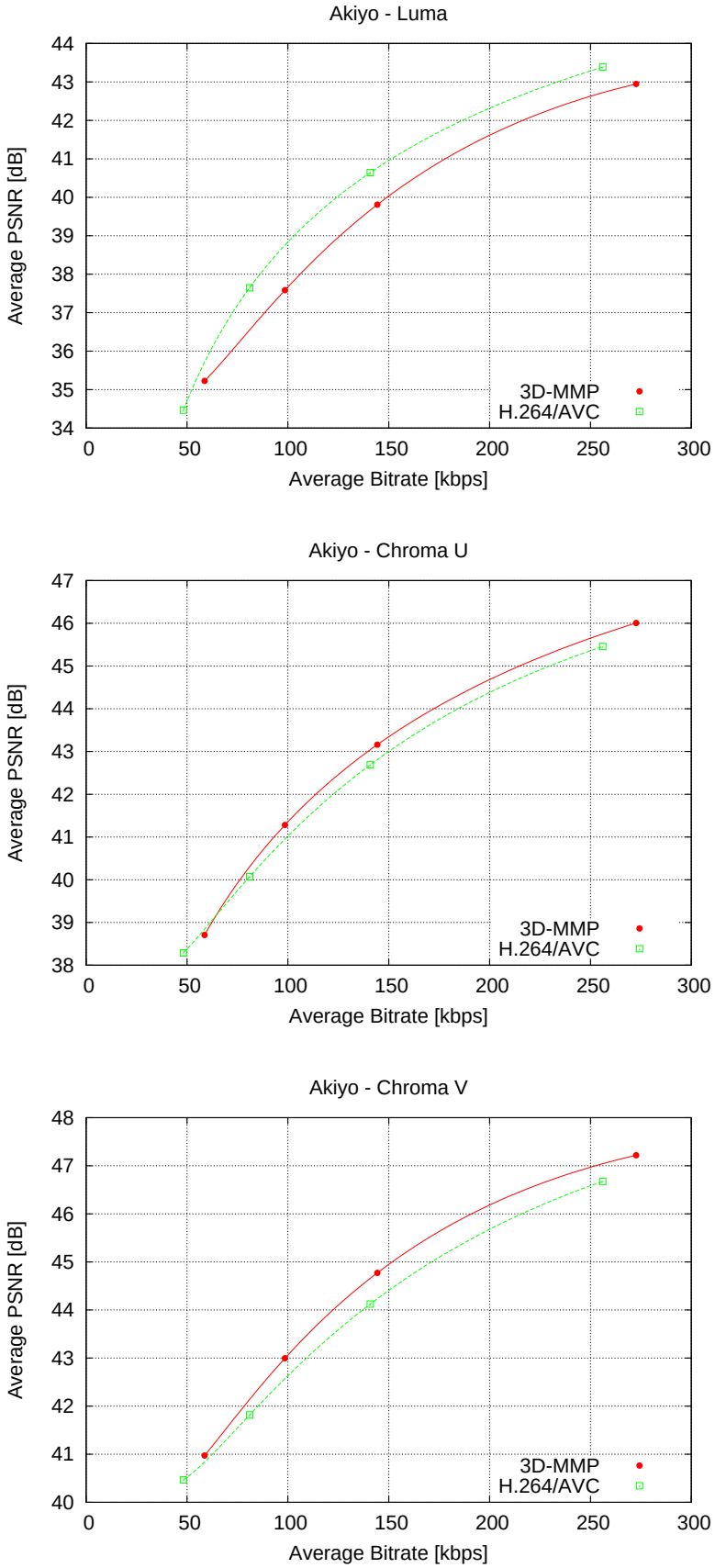


Figure G.11: Comparative results for the 3D-MMP video encoder and the H.264/AVC high profile video encoder, for the Akiyo sequence (CIF).

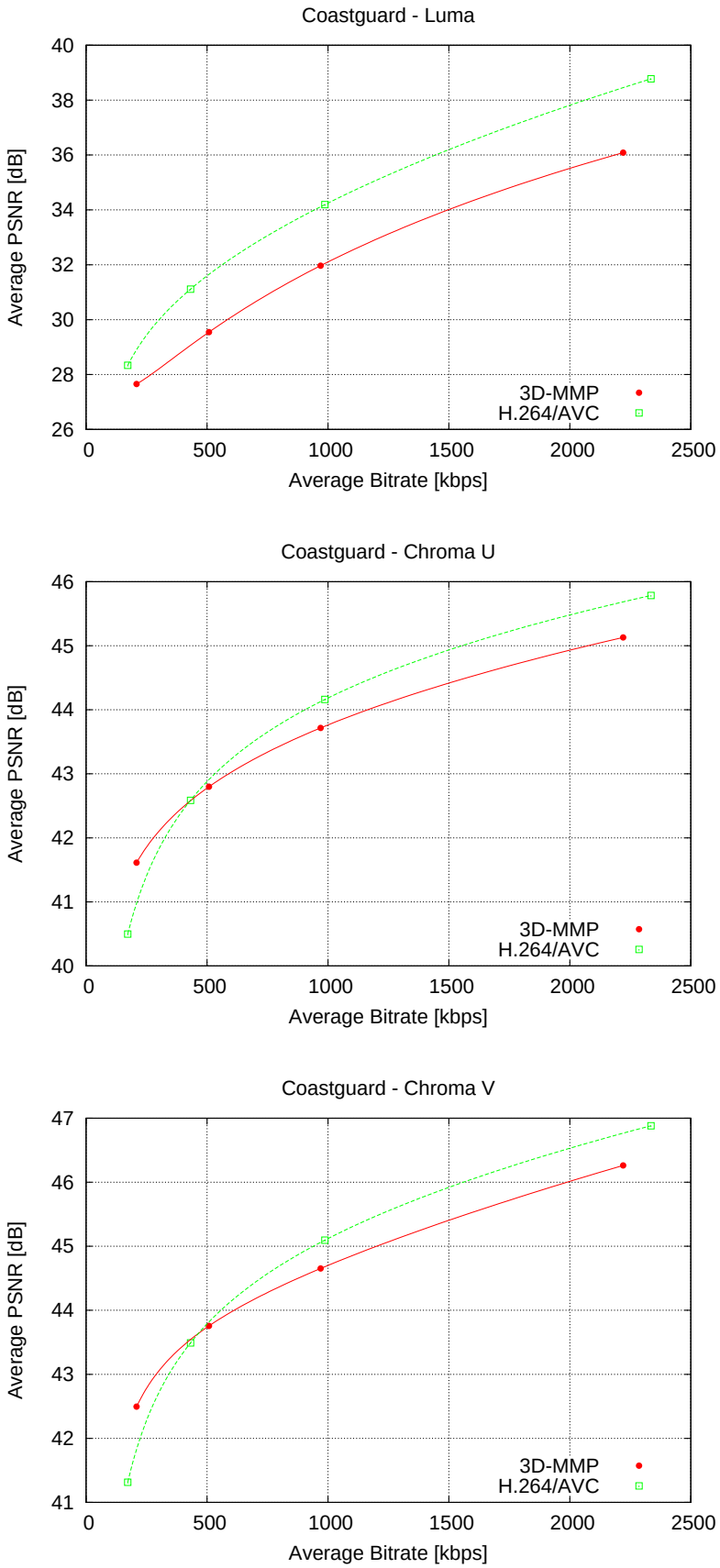


Figure G.12: Comparative results for the 3D-MMP video encoder and the H.264/AVC high profile video encoder, for the Coastguard sequence (CIF).

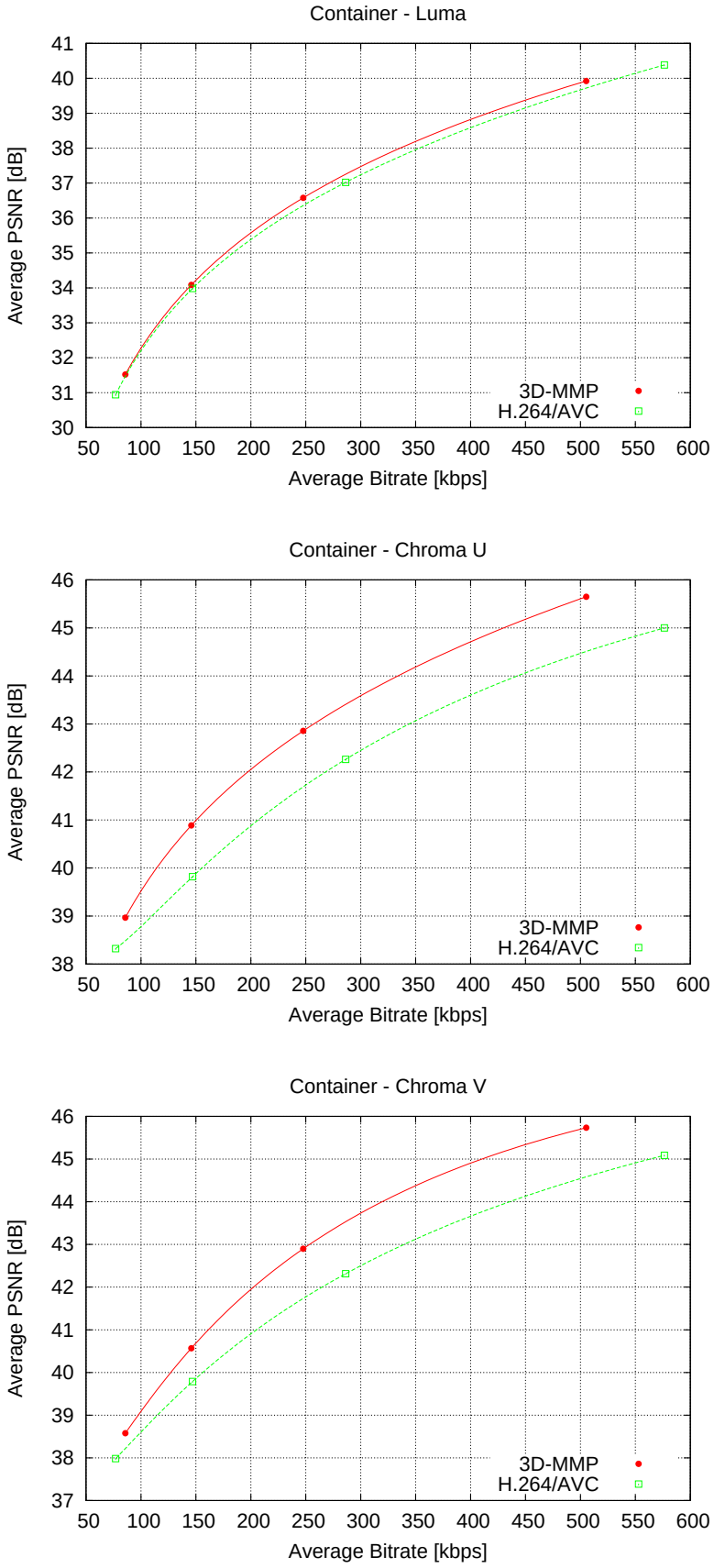


Figure G.13: Comparative results for the 3D-MMP video encoder and the H.264/AVC high profile video encoder, for the Container sequence (CIF).

Table G.1: Comparison of the global R-D performances of 3D-MMP and H.264/AVC JM 17.1. The BD-PSNR corresponds to the performance gains of 3D-MMP over H.264/AVC.

		H.264/AVC				3D-MMP				BD-PSNR		
	QP [I/P-B]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**BR** [kbps]	**Y** [dB]	**U** [dB]	**V** [dB]	**Y** [dB]	**U** [dB]	**V** [dB]
Akiyo	23-25	256.12	43.39	45.46	46.68	272.66	42.95	46.01	47.22	**-0.91**	**0.27**	**0.42**
	28-30	140.84	40.64	42.69	44.12	144.35	39.81	43.16	44.77			
	33-35	81.06	37.65	40.07	41.82	98.56	37.59	41.28	43.00			
	38-40	48.22	34.47	38.29	40.47	58.69	35.23	38.71	40.97			
Coastguard	23-25	2335.07	38.78	45.79	46.88	2220.32	36.09	45.13	46.26	**-2.03**	**-0.17**	**-0.14**
	28-30	987.19	34.19	44.16	45.10	969.46	31.97	43.72	44.65			
	33-35	431.83	31.11	42.59	43.49	507.94	29.54	42.80	43.76			
	38-40	172.47	28.34	40.50	41.31	208.84	27.66	41.61	42.50			
Container	23-25	576.35	40.38	45.00	45.09	505.28	39.92	45.65	45.73	**0.17**	**1.05**	**0.96**
	28-30	286.29	37.02	42.26	42.31	247.71	36.58	42.85	42.90			
	33-35	146.99	33.99	39.82	39.79	145.82	34.08	40.89	40.57			
	38-40	76.99	30.94	38.32	37.98	85.78	31.52	38.97	38.58			

For example, for $\lambda = 200$, the proposed method uses the directional prediction mode to predict 99.6% of the pixels from the B-type frames, and the prediction is so efficient that 96.9% of those pixels are encoded using the null residue pattern (through the use of the 0 CBP flag). Furthermore, the high correlation between the best directional vector (DV) for each block and those from its neighbors, allows the algorithm to efficiently predict these vectors, avoiding its transmission for most cases.

This contributes to a low average entropy observed for the vectors. In average, 0.33 bits are required to transmit the k_1 component, and 0.15 bits for the k_2 component.

As a result, the rate required to encode the B-type blocks corresponds to only 10% of the rate required for the I/P-type blocks, demonstrating the efficiency of the hierarchical architecture for video compression. Note that H.264/AVC needs to transmit a vector corresponding to each block for each frame, resulting in a less efficient representation for this case.

For the case of sequence Akiyo, the almost static background is also efficiently predicted by the proposed directional prediction. In this case, 99.8% of the B-type blocks are encoded using the directional prediction mode, and no residue is transmitted for 99.8% of the pixels from these frames. The average entropy for the DVs are respectively 0.49 and 0.89 bits, for the k_1 and k_2 components. However, H.264/AVC is also very efficient while encoding this static background, as it uses mostly the copy and skip modes, which require the transmission of very little information. Thus, H.264/AVC is able to outperform the proposed in 0.9dB, for this particular sequence.

Sequence Coastguard presents a larger quantity of motion from several types. A fast panning and several moving objects are accompanied by some camera jittering. This

Table G.2: Rate used by each type of symbol, for the first 64 frames of sequences encoded using $\lambda = 200$.

	Akiyo		**Coastguard**		**Container**	
Symbols	**Bits**	**%**	**Bits**	**%**	**Bits**	**%**
Bits for indices	65880	31.4%	237858	21.9%	121102	39.0%
Bits for dicsegs	36072	17.2%	115256	10.6%	57368	18.5%
Bits for CBP	536	0.3%	3816	0.4%	1744	0.6%
Bits for flags	55552	26.4%	207320	19.1%	85984	27.7%
Bits for modes	13536	6.4%	55864	5.2%	23408	7.5%
Bits for DV flag	1544	0.7%	3920	0.4%	1896	0.6%
Bits for DV	36888	17.6%	459296	42.4%	19320	6.2%
Bits for α	16	0.0%	48	0.0%	8	0.0%
Total bits	210024	100%	1083378	100%	310830	100%

erratic movement difficult the finding for adequate matches for multiple frames. Thus, the algorithm needs to segment the block along the temporal axis in order to achieve proper matches, converging to the traditional frame-by-frame ME. However, for a frame-by-frame estimation, the proposed algorithm is not able to achieve the same prediction performance obtained using the more complex quarter pixels ME used in H.264/AVC, which benefits from a larger searching window than that provided by reference resulting from the adopted range for the DVs.

Furthermore, the high detail presented by this particular sequence imposes the need of performing more segmentations than for the other test sequences, resulting in average, in smaller blocks, and consequently in a larger number of DVs which need to be transmitted. These vectors are also more difficult to predict using the block's neighborhood, due to the erratic motion observed in this sequence, increasing the average entropy to, 1.78 and 0.51 bits, respectively for the k_1 and the k_2 components.

A total of 99.8% of the B-type blocks' pixels are encoded using the directional prediction, and a null residue pattern is transmitted for 98.8% of those pixels. This demonstrates that the directional prediction is still able to adapt to the more complex motion presented by this sequence, but at the expense of more segmentations and more rate to transmit the information related with the DVs.

Table G.2 presents the amount of rate used by each symbol type, for the 64 frames of each of the three video sequences, and its corresponding percentage in the final bitstream. Note that the context conditioning technique proposed in [49] was adopted in our method, so each code-vector is identified by a dictionary partition (dicseg), which corresponds to the original scale were the code-vector was created, and by the index of the that code-vetor (index) inside that partition.

One may observe that for the case of sequence Coastguard, the rate required to trans-

mit the DVs corresponds to almost half of the total bitrate used to encode the video sequence. This is explained by the use of smaller blocks, which increases the number of DVs to be transmitted, and by the higher average entropy for these blocks. Thus, it is important to notice that the cases for which H.264/AVC outperforms the proposed method, are exactly the same cases where the DV related information corresponds to a more significant portion of the final bitstream. Thus, the investigation for improved DVs' compression and estimation techniques may result in significant performance increases for the proposed 3D-MMP video codec.

The approach adopted to encode the DVs information is still simple, and the use of more sophisticated entropy coding methods, such as the CABAC [82] used by H.264/AVC [51], can result in a significant reduction of the rate required to transmit this information. Furthermore, the directional prediction can be improved by the use of information simultaneously from both the past and future neighbors, in order to generate averaged predictions, or even estimate the DVs for the B-type frames. The directional prediction can also be further adapted to use a half or quarter pixel accuracy, in order to increase the prediction's efficiency, at the expense of a larger computational complexity.

It is important to refer that the introduction of the vector estimated directional prediction mode reduced considerably the use of the LSP mode. The LSP is able to provide a good directional implicit prediction when the behavior of the block is correlated with that from its neighborhood. However, when this situation occurs, the DV is also efficiently predicted based on the block's neighborhood an does not need to be transmitted, so the originally explicit directional prediction also performs as an implicit method. Thus, both modes tend to perform well for these cases, but the directional prediction tends to be advantageous because of the lower rate needed to transmit this mode, which results from its most frequent use.

However, the LSP prediction mode is still useful in many cases, and its usage has the tendency to increase for lower compression ratios, where more accurate predictions are required. When a low distortion is required, the LSP mode may be able to generate an implicit non-integer accuracy prediction and adapt to the periodic variations both in time and space, justifying the eventual increase in the mode signaling. In some cases, the LSP mode is used to predict more than 10% of the pixels from the video sequence, as is the case for the sequence Container encoded with $\lambda = 10$. Its usage is more significant for the I/P-type frames, for which the directional prediction tends to perform worse. In this case, the LSP mode was used to predict over 16% of the pixels from I/P type blocks.

In order to demonstrate the contribute of some of the proposed techniques in the overall performance of the presented video compression algorithm, Figure G.14 presents the experimental results obtained while encoding the sequence Container, with some of these techniques disabled. As the results are consistent for the other tested sequences, only the results for the sequence Container are presented as a reference.

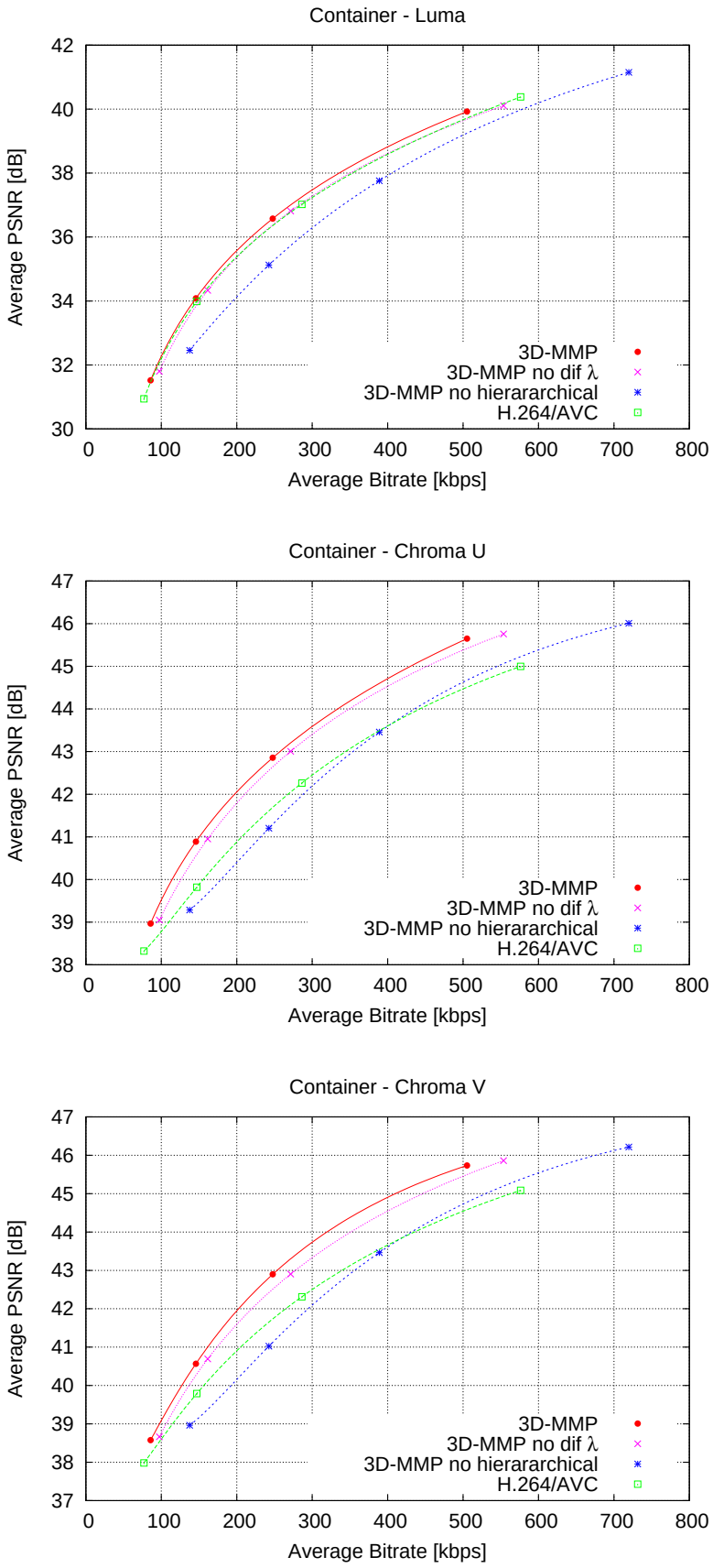


Figure G.14: Comparative results for the 3D-MMP encoder with and without the hierarchical prediction and the use of different values for the λ P and B-type blocks, and the H.264/AVC high profile video encoder, for the Container sequence (CIF).

The use of different values for the lagrangian operator λ, respectively for the I/P-type and the B-type blocks, contributes to a BD-PSNR increase of 0.23 dB, 0.26 dB and 0.34 dB, respectively for the Y, U and V components, over the version of the algorithm which uses the same value of λ for both block types. The plot corresponding to the use of the same value of λ for all block types is referred to as "3D-MMP no dif λ". Furthermore, from Figure G.14, one may notice that the increase on the compression efficiency of the algorithm resulting from this method is consistent for all the tested compression ratios.

The use of the hierarchical frame coding illustrated in Figure G.8, interleaving one B-type frame between each 2 previously encoded I/P-type frames, contributes to a BD-PSNR increase of 1.20 dB, 1.40 dB and 1.55 dB, for the Y, U and V components, respectively. Note that these performance gains also rely on the use of different values for the lagrangian operator λ, only possible when the use of different block types is enabled. The results obtained using the hierarchical frame coding disabled, corresponds to the plot marked as "3D-MMP no hierarchical" in Figure G.8. One may notice that the performance gains in this case have the tendency to increase for increasing bitrates. Reference frames presenting a higher quality allow to generate better predictions for B-type frames. Thus, less segmentations tend to occur and more residue blocks are encoded using the zero CBP, contributing to a more efficient representation for these frames.

The results presented in this section demonstrate the potential of the proposed method for video compression applications, and allowed to identify the topics which deserve further researches in the future. Thus, several optimized techniques for the DVs prediction and encoding shall be investigated in the future, as well as other techniques which can improve the spatiotemporal prediction. The use of 3D-MMP will also be investigated for other types of input signals.

The competitive performance of the proposed prediction techniques also provides good expectations regarding the use of this compression architecture with a three-dimensional DCT instead of MMP, in order to develop competitive low complexity video compression algorithms.

G.5 Conclusions

In this appendix, we presented a new MMP-based volumetric signal compression framework. The proposed framework adopted an hierarchical volumetric prediction, with the 3D resulting residue being encoded using a three-dimensional extension of the MMP algorithm.

For that purpose, several functional implementations were proposed for MMP, in order to adapt the algorithm for volumetric signal compression, and the parameters which define the algorithm's performance where evaluated and optimized for the new framework. Furthermore, we proposed several prediction modes adapted for volumetric signals, such

as three-dimensional block-based least squares prediction and directional predictions.

The proposed framework was evaluated for video compression, with results that outperform state-of-the-art hybrid video codecs in some cases. However, we believe that several optimizations can still be performed on the developed framework, targeting a more efficient prediction for the volumetric data. For example, more sophisticated directional vectors prediction can contribute to decrease the rate required for their transmission, which is responsible for almost half the required bit-rate on some sequences. A multi-level hierarchical bi-prediction can also be helpful while improving the algorithm's performance, such as in H.264/AVC [51] and HEVC [16].

In the future, we intend to test the proposed compression architecture for other types of input signals, such as tomographic scan signals, multispectral images, meteorological radar images of multiview images. All these input signals types present a large degree of correlation between their several dimensions, which can be efficiently exploited by the proposed prediction tools.

Furthermore, we also intend to test the replacement of the MMP algorithm [3] in the developed framework by some other residue compression algorithms, such as a 3D extension of fractals [96], or 3D transforms, such as [97–104]. The use of a 3D transform for residue coding purposes can be a viable solution to develop efficient and low computational complexity algorithms for volumetric signals compression.

Appendix H

Conclusions and perspectives

H.1 Final considerations

In the previous appendices, we have described the fundamental topics related to the work developed during this thesis. The specific conclusions, regarding each of the covered topics, as well as the corresponding results, are presented in the last sections of each appendix.

The multidimensional multiscale parser algorithm was studied in detail, and several contributions were proposed, focusing the improvement of the rate-distortion performance and perceptual quality of the reconstructed images, and the reduction of the computational complexity of the proposed algorithms. As a result, new optimized frameworks were developed, both for text images, scanned compound documents and video compression. Each of the proposed methods achieved results competitive with those from the state-of-the-art algorithms, for the considered application.

Additionally, a new research line was initiated, resulting from the combination of a volumetric extension of MMP with a three dimensional hierarchical prediction scheme. The new framework was tested for video compression applications, presenting some interesting results presented for this particular application. These results demonstrated the potential of such approach, which will justify further investigations and its extension for other applications involving three-dimensional data sources.

H.2 Original contributions

In this section, we present a summary of the most relevant original contributions of the research work described in this thesis. The contributions are not only related with the MMP algorithm, since some of the proposed methods are extensive to another pattern matching methods, or even to other block-based image coding algorithms.

The validation of the developed work within the scientific community has been con-

sidered very important in order to access its relevance. As a consequence, most of the results achieved have either been published on international journals or in the proceedings of national and international conferences. The complete list of the papers published to this date is presented, for reference, in Appendix J.

The most relevant contributions of this thesis can be summarized in the following topics:

- **The MMP-Compound algorithm: a MMP based encoder for scanned compound document encoding**

  The investigation on optimizing the coding efficiency of MMP both for smooth and for text and graphics images resulted in two algorithms. Both of these methods are able to outperform state-of-the-art encoders for its application fields. The combination of both encoders in a segmentation-driven framework, described in Appendix C, resulted in an efficient scanned compound document encoder that proved to be robust and efficient.

  Experimental results demonstrated that the proposed algorithm considerably outperformed, both perceptually and objectively, other state-of-the-art segmentation-driven compound document encoders, as well as generic still image encoders.

  The work developed under this topic resulted in the publication: "Scanned Compound Document Encoding Using Multiscale Recurrent Patterns", published on the *IEEE Transactions on Image Processing*.

- **The MMP-Video algorithm: an efficient fully pattern-matching-based video compression algorithm**

  The development of a fully MMP-based video compression algorithms was one of the main objectives of this thesis.

  The conducted investigation resulted on MMP-video, a hybrid video coder framework that uses MMP to compress both the intra and the motion-compensated residues. The use of the multiscale recurrent pattern paradigm for video compression was optimized, based on the previous experience with still image encoders, and new methods, specifically designed to exploit the video signal's particular features, were studied and developed.

  The resulting video coding framework is totally based on the pattern matching paradigm, and was able to achieve a considerable compression performance advantage over the state-of-the-art H.264/AVC video coding standard, for medium to low compression rates. These results demonstrated that the pattern matching paradigm can present a viable alternative to the ubiquitous transform-based paradigm.

  These results validate the use of the multiscale recurrent pattern matching paradigm also for video compression, and resulted on the paper "Efficient Recurrent Pattern

Matching Video Coding", published on the *IEEE Transactions on Circuits and Systems for Video Technology*. Further researches will be developed based on this framework, in order to extend the application range to high-definition video sequences or even 3D and multiview video signals.

- **Study of computational complexity reduction methods to apply on MMP-based encoders**

  The major issue associated to the practical use of dictionary/pattern matching-based encoders is their high computational complexity. In the case of MMP, this problem is still aggravated by the fact that the decoder also presents a considerable computational complexity, limiting the use of MMP even for encode-once-decode-many times application scenarios.

  We investigated some complexity reduction methods which can be applied to generic dictionary-based compression method. When applied to MMP-base enconders, these techniques were able to decrease up to 86% and 95% the time required by the encoder and decoder, respectively, without any considerable losses in the rate-distortion performance. The developed techniques can be used in conjunction with previously studied methods, allowing to further increase the time saving for the MMP codec.

  In spite of these gains, this topic will probably be included in further researches, as the computational complexity associated to MMP-based codecs is still considerably higher than that of most transform-based algorithms, and may still be limitative for many applications.

  The achieved results were described in the paper: "Computational Complexity Reduction Methods for Multiscale Recurrent Pattern Algorithms", published on the proceedings of the international conference *Eurocon2011 - International Conference on Computer as a Tool*.

- **Improving multiscale recurrent pattern image coding with post-processing deblocking filtering**

  Like most block-based algorithms, MMP has the tendency to introduce some blocking artifacts in the reconstructed images, specially for high compression ratios. This motivated previous studies on deblocking methods applied to MMP. However, the existing methods revealed several compromising inefficiencies. The previous methods were also specific for MMP, and could not be applied to any other algorithm.

  A new deblocking method was then investigated, in order to overcome these inefficiencies, and improve the overall perceptual quality of the reconstructions obtained for image and video sequences not only encoded using MMP, but also JPEG, H.264/AVC or HEVC.

The proposed deblocking algorithm is a a post-processing method which uses an adaptive FIR filter to process each image block. The filter shape is adaptively defined, in accordance to the local features of the image region that is being processed. A total variation analysis of the reconstructed image allows to determine the optimal filters support for each region.

The developed deblocking filter can either work as an interactive filter, optimized for each particular image type, or operate using pre-established parameter values, that do not need to be sent to the decoder. This approach allows to use the filter as a post-processing method, preserving the compliance of the bitstream with coding standards. The method has been successfully used on images compressed with various image encoders, namely the H.264/AVC coding standard, the upcoming HEVC and JPEG.

The achieved results were described in the paper: "A Generic Post Deblocking Filter for Block Based Image Compression Algorithms", published on the *Elsevier Signal Processing : Image Communications*.

- **Development of a volumetric multiscale recurrent pattern based compression framework**

  In order to investigate the applicability of the multiscale recurrent patterns for several types of three-dimensional data sources, such as video sequences, 3D video sequences, tomographic and meteorological radar scans or multispectral images, we developed a volumetric compression framework, based on hierarchical prediction, that uses a three-dimensional version of the MMP algorithm to encode the resulting residue.

  Several volumetric prediction modes have been investigated, including 3D extensions of several H.264/AVC prediction modes, a volumetric least squares based prediction and a directional volumetric prediction. Furthermore, an intensive evaluation of the impact of each of the MMP parameters on a 3D framework has been performed.

  The developed algorithm has been tested for video sequences compression. The experimental results demonstrated the potential from such approach, opening several new research lines. Thus, further improvements will be implemented on the described encoding method, and it will be evaluated also for other three-dimensional input signals.

  The results achieved for video compression were described in the paper: "Video Compression Using 3D Multiscale Recurrent Patterns", submitted to the *IEEE International Symposium on Circuits and Systems 2013*.

H.3 Future perspectives

The work presented in this thesis, as well as other related works, has demonstrated that MMP is a versatile tool for image compression, achieving state-of-the-art results under several coding scenarios. However, at the actual stage, despite the algorithm's large computational complexity reduction, MMP is still far from being a practical solution for image coding. Thus, a number of open questions still remains. Why to invest time developing such a computationally complex algorithm? Do we really need new image and video compression approaches, or should we stick to successful existing schemes?

The search for different solutions for known-problems is the best way of achieving disruptive ways to deal with those problems, and the resulting capability to "think out of the box" can be determinant while achieving improved solutions. Therefore, the knowledge gathered investigating the MMP algorithm can reveal useful when applied to other compression paradigms, such as transform-based techniques. Understanding MMP can also help to understand how images are formed, allowing to develop more efficient ways of representing them. Therefore, the research for methods other than those from the mainstream, such as the ones proposed on this thesis, have the potential of enlarging the image compression understanding, and thus should continue.

Furthermore, the computational complexity burden seems to be less and less important as time goes by, with the development of machines with more and more computational power. Additionally, the development of highly efficient processing hardware, such as GPUs, may contribute to make MMP a practical solution for image and video compression in the future. This leads us also to another open question, which is the impact of the increasing hardware capabilities on the proposed encoding algorithms. How can MMP be improved to leverage on the potential of hardware specific solutions, is an interesting open question.

Among the proposals presented in this thesis, several research topics are candidates for further future investigations. The new insights on image and video compression tools, hardware resources and content demands, make visual signals compression a permanently open research topic.

On the future, we expect to extend the proposed post-deblocking filter described on Appendix F to a volumetric layout, in order to develop a joint spatiotemporal filtering method. This approach can allow to simultaneously attenuate two of the most annoying artifacts on highly compressed video sequences: blocking artifacts and uniform block flickering, which results from aggressive quantization. The information regarding both the spatial and the temporal dimensions can be useful while distinguishing natural objects edges from the block boundaries introduced in the compression stage.

The work presented on Appendix G is the research line which presents more open topics. Several improvements can still be done to enhance not only the spatiotemporal

prediction, but also the directional vector entropy coding. Furthermore, we intend to develop an alternative compression scheme based on our framework, where 3D transforms are used to compress the generated residue. The best contributions from such approach can be the development of a low complexity general purpose volumetric signal encoding scheme.

Appendix I

Test signals

I.1 Test images

Figure I.1: Grayscale natural test image Lena (512×512).

Figure I.2: Grayscale natural test image Barbara (512×512).

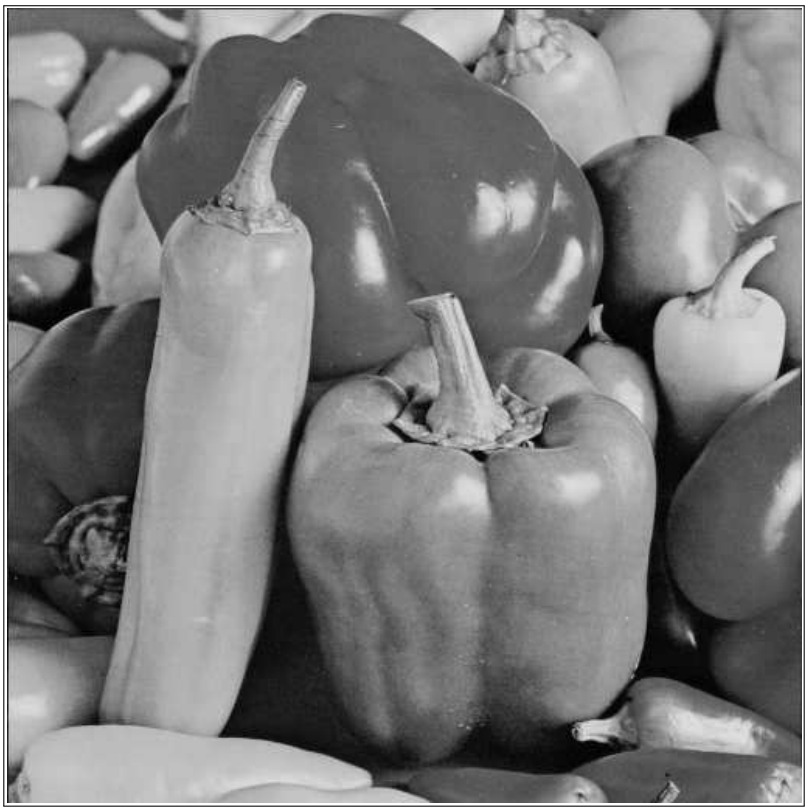


Figure I.3: Grayscale natural test image PEPPERS512 (512×512).

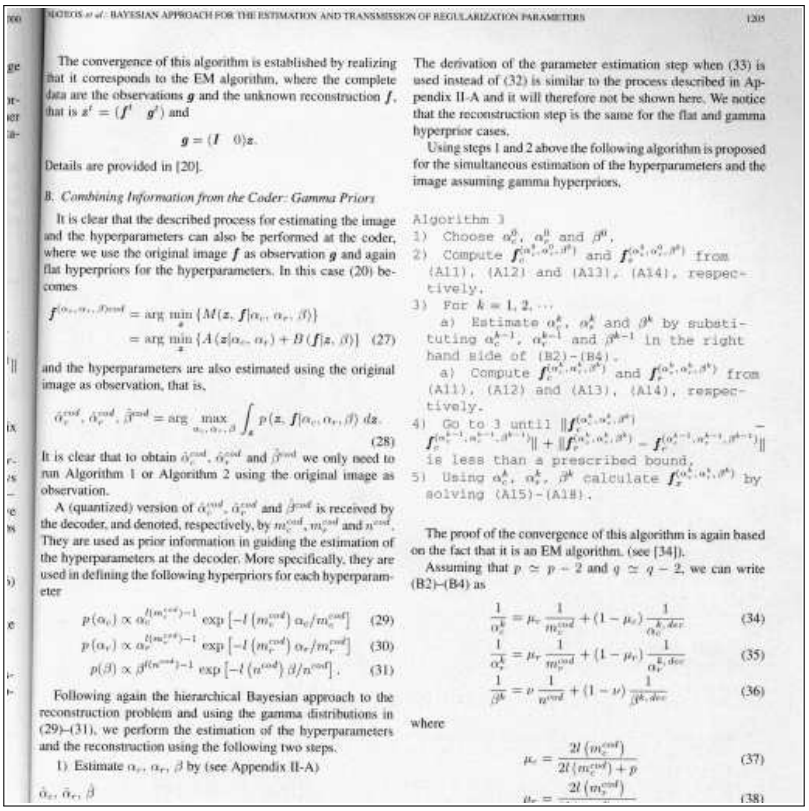


Figure I.4: Grayscale text test image PP1205 (512×512).

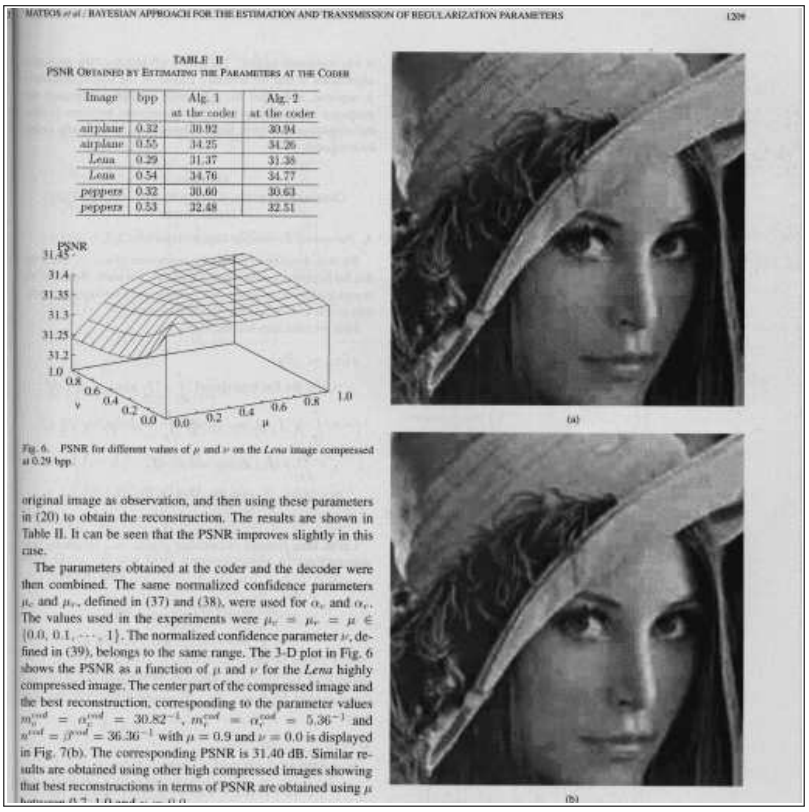


Figure I.5: Grayscale compound test image PP1209 (512×512).

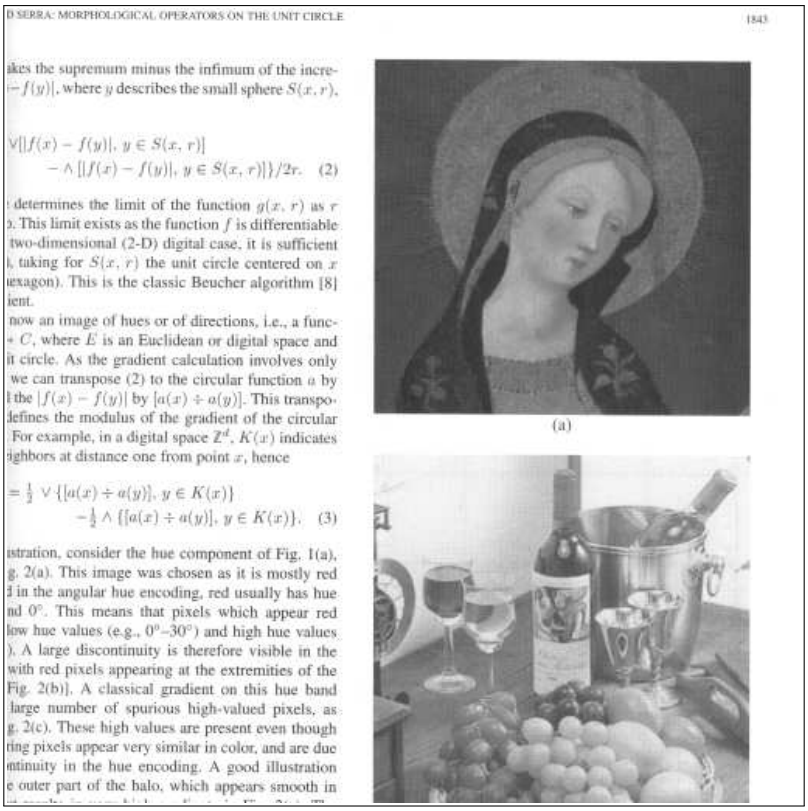


Figure I.6: Grayscale compound test image SCAN0002 (512×512).

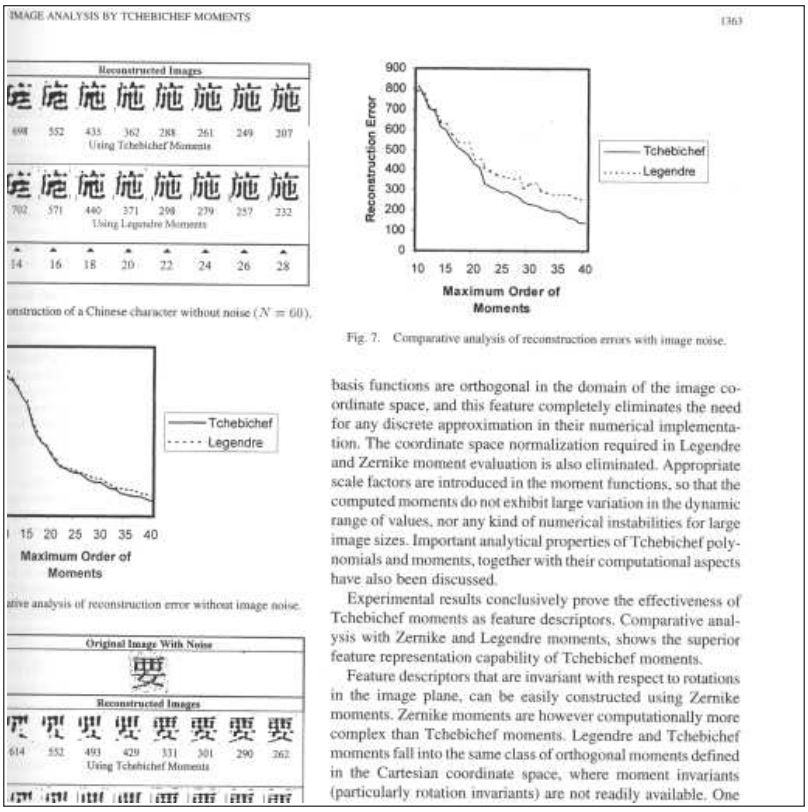
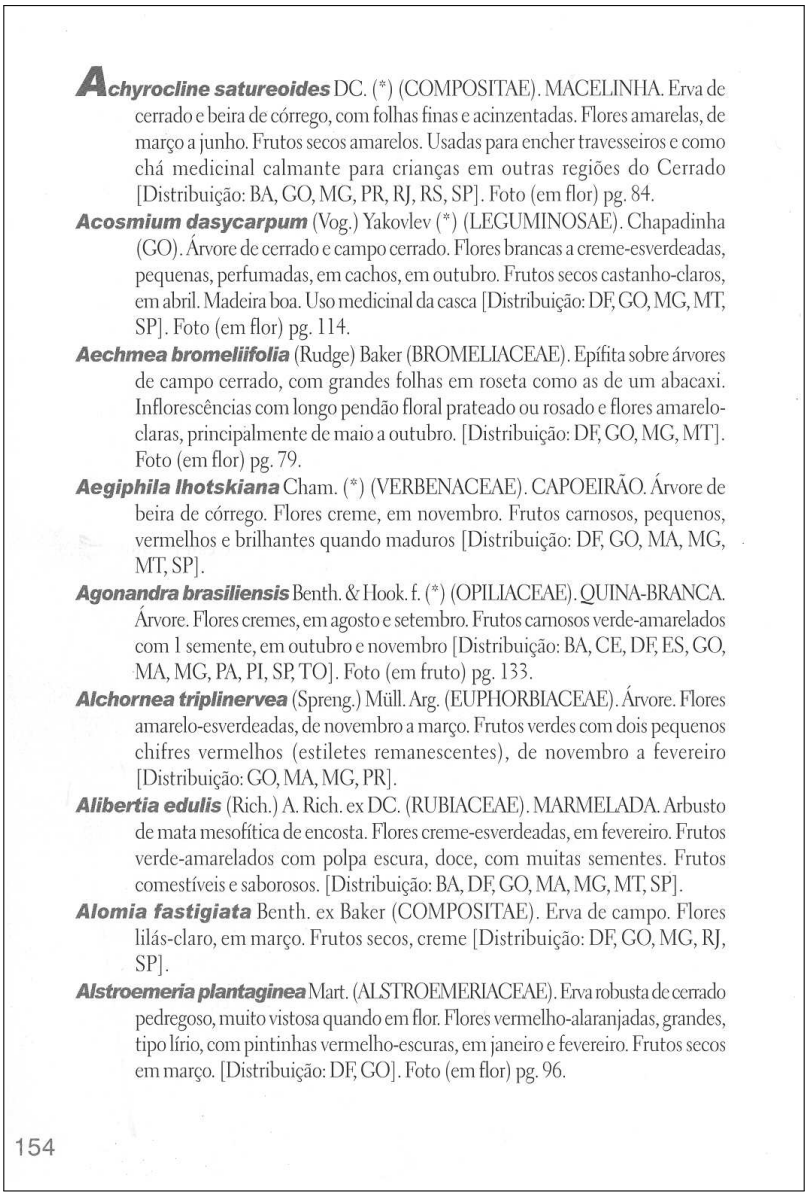


Figure I.7: Grayscale text test image SCAN0004 (512×512).



Achyrocline satureoides DC. (*) (COMPOSITAE). MACELINHA. Erva de cerrado e beira de córrego, com folhas finas e acinzentadas. Flores amarelas, de março a junho. Frutos secos amarelos. Usadas para encher travesseiros e como chá medicinal calmante para crianças em outras regiões do Cerrado [Distribuição: BA, GO, MG, PR, RJ, RS, SP]. Foto (em flor) pg. 84.

Acosmium dasycarpum (Vog.) Yakovlev (*) (LEGUMINOSAE). Chapadinha (GO). Árvore de cerrado e campo cerrado. Flores brancas a creme-esverdeadas, pequenas, perfumadas, em cachos, em outubro. Frutos secos castanho-claros, em abril. Madeira boa. Uso medicinal da casca [Distribuição: DF, GO, MG, MT, SP]. Foto (em flor) pg. 114.

Aechmea bromeliifolia (Rudge) Baker (BROMELIACEAE). Epífita sobre árvores de campo cerrado, com grandes folhas em roseta como as de um abacaxi. Inflorescências com longo pendão floral prateado ou rosado e flores amarelo-claras, principalmente de maio a outubro. [Distribuição: DF, GO, MG, MT]. Foto (em flor) pg. 79.

Aegiphila lhotskiana Cham. (*) (VERBENACEAE). CAPOEIRÃO. Árvore de beira de córrego. Flores creme, em novembro. Frutos carnosos, pequenos, vermelhos e brilhantes quando maduros [Distribuição: DF, GO, MA, MG, MT, SP].

Agonandra brasiliensis Benth. & Hook. f. (*) (OPILIACEAE). QUINA-BRANCA. Árvore. Flores cremes, em agosto e setembro. Frutos carnosos verde-amarelados com 1 semente, em outubro e novembro [Distribuição: BA, CE, DF, ES, GO, MA, MG, PA, PI, SP, TO]. Foto (em fruto) pg. 133.

Alchornea triplinervea (Spreng.) Müll. Arg. (EUPHORBIACEAE). Árvore. Flores amarelo-esverdeadas, de novembro a março. Frutos verdes com dois pequenos chifres vermelhos (estiletes remanescentes), de novembro a fevereiro [Distribuição: GO, MA, MG, PR].

Alibertia edulis (Rich.) A. Rich. ex DC. (RUBIACEAE). MARMELADA. Arbusto de mata mesofítica de encosta. Flores creme-esverdeadas, em fevereiro. Frutos verde-amarelados com polpa escura, doce, com muitas sementes. Frutos comestíveis e saborosos. [Distribuição: BA, DF, GO, MA, MG, MT, SP].

Alomia fastigiata Benth. ex Baker (COMPOSITAE). Erva de campo. Flores lilás-claro, em março. Frutos secos, creme [Distribuição: DF, GO, MG, RJ, SP].

Alstroemeria plantaginea Mart. (ALSTROEMERIACEAE). Erva robusta de cerrado pedregoso, muito vistosa quando em flor. Flores vermelho-alaranjadas, grandes, tipo lírio, com pintinhas vermelho-escuras, em janeiro e fevereiro. Frutos secos em março. [Distribuição: DF, GO]. Foto (em flor) pg. 96.

154

Figure I.8: Grayscale text test image CERRADO (1056×1568).

EDIÇÃO ALEXANDRE VERSIGNASSI (aversignassi@abril.com.br)

SUPERFETICHE

OBJETOS DE DESEJO

DARWIN NO PC

Agora você é Deus

A evolução das espécies está nas suas mãos. Chegou *Spore*, o jogo mais aguardado dos últimos anos.

TEXTO PEDRO BURGOS

Você começa o jogo como um micróbio e vai evoluindo até que seus descentes se transformem numa espécie superinteligente, capaz de explorar todos os cantos da galáxia. Essa é a essência de *Spore*, uma das maiores superproduções da história dos games de computador, que demorou 5 anos para ser produzido. Mas dizer isso é pouco. Esse game junta o que tem de melhor nos outros jogos de estratégia. E coloca o jogador no comando de um jeito inédito. Em *Spore* você não apenas gerencia seu império mas também vê as cidades em todos os detalhes. Não só monta tropas como controla os movimentos de cada um dos soldados. Desenhando a armadura, inclusive. Tudo isso não é para menos: *Spore* é o projeto mais ambicioso do britânico Will Wright, que tem as credenciais de ter criado dois jogos que mudaram a indústria dos jogos eletrônicos. Foi ele quem bolou sozinho há quase 20 anos o *SimCity*, deixando os jogadores como prefeitos superpoderosos de uma cidade, e mais recentemente fez *The Sims*, o jogo mais vendido de todos os tempos, em que você comanda pessoas. *Spore* de uma certa forma tenta juntar os dois conceitos. E, mais do que isso, busca agradar os jogadores casuais. No nível fácil de dificuldade dá para jogar de maneira bem descompromissada, montando bichos absurdos e testando suas possibilidades de sobrevivência e de evolução. Tipo: uma criatura herbívora que dança e canta para fazer amizade teria mais ou menos chances que um predador de dentes afiados, veneno e espinhos? Jogue e verá. S

ASSIM CAMINHA A MONSTRUOSIDADE

Você começa como micróbio e, se for competente, acaba como Dart Vader.

1 FAÇA-SE A LUZ

Um cometa se choca contra um planeta e... surpresa: surge o milagre da vida em *Spore*. Você começa o jogo como um projeto de verme na sopa primordial, o caldo marinho onde apareceram as primeiras células, há 4 bilhões de anos.

2 LEI DO MAIS FORTE

Depois de escapar dos vermes inimigos e comer bastante nutrientes, você ganha o direito de partir para a fase seguinte da evolução e desenhar seu corpo do zero. Se o seu animal for bem-sucedido, ele vai ficar cada vez mais esperto.

3 CIVILIZAÇÃO

Os bichos mais inteligentes conseguem formar tribos, depois cidades. Aí é hora de conquistar o mundo. Mais: toda vez que você inicia o jogo, ele busca criaturas e prédios criados por outros jogadores para povoar os cenários.

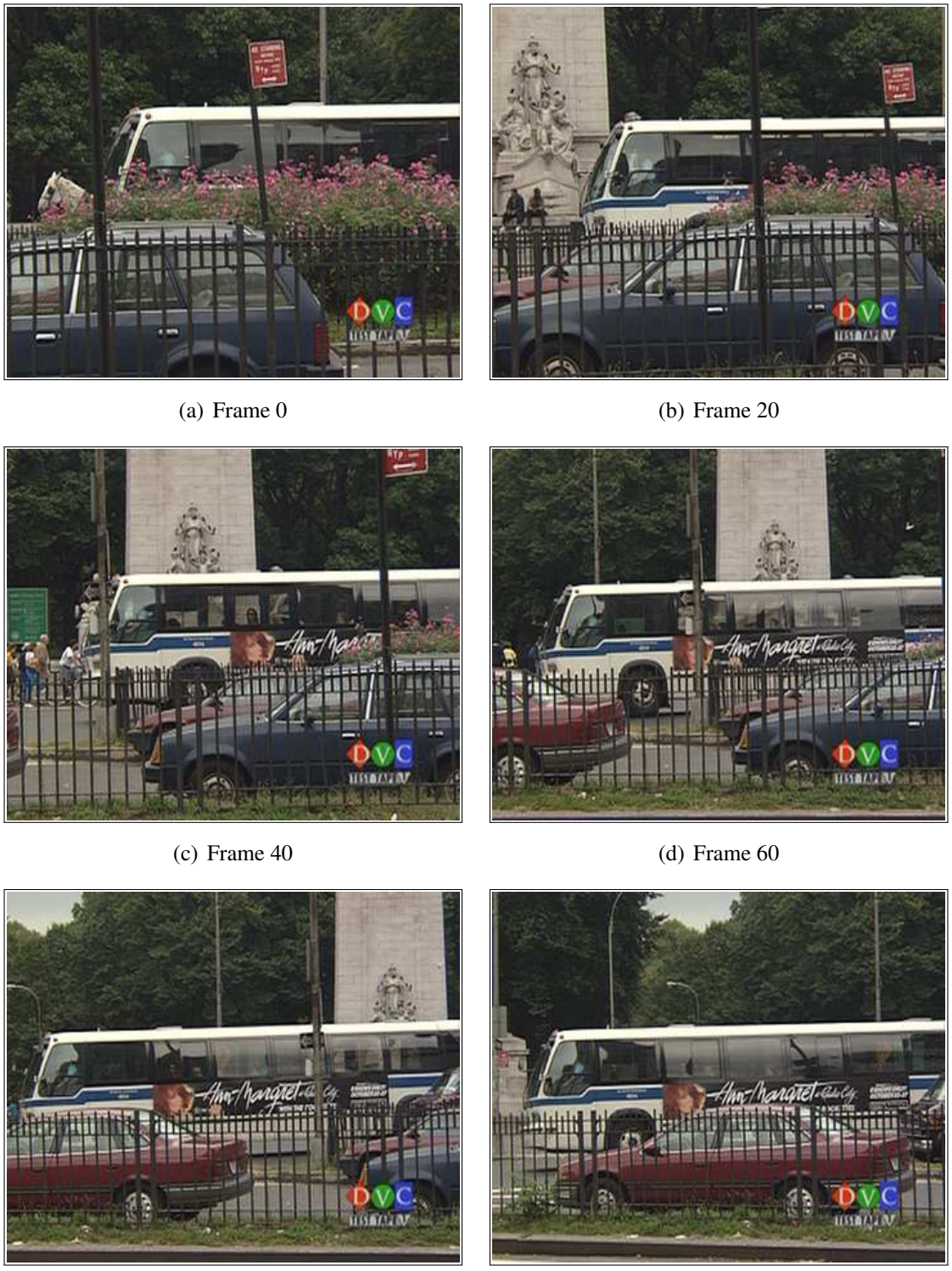
4 STAR WARS

Depois de crescer e aparecer no planeta, sua civilização pode construir uma nave espacial. Só aí dá para ver o tamanho real do jogo. Há incontáveis planetas, cada um com paisagens diferentes. E muitos são cheios de vida... Hora de conquistá-los.

Imagens Divulgação

Figure I.9: Grayscale compound test image SPORE (1024×1360).

I.2 Test video sequences



(a) Frame 0 (b) Frame 20

(c) Frame 40 (d) Frame 60

(e) Frame 80 (f) Frame 100

Figure I.10: Frames from the Bus video sequence (CIF:352×288).

(a) Frame 0

(b) Frame 20

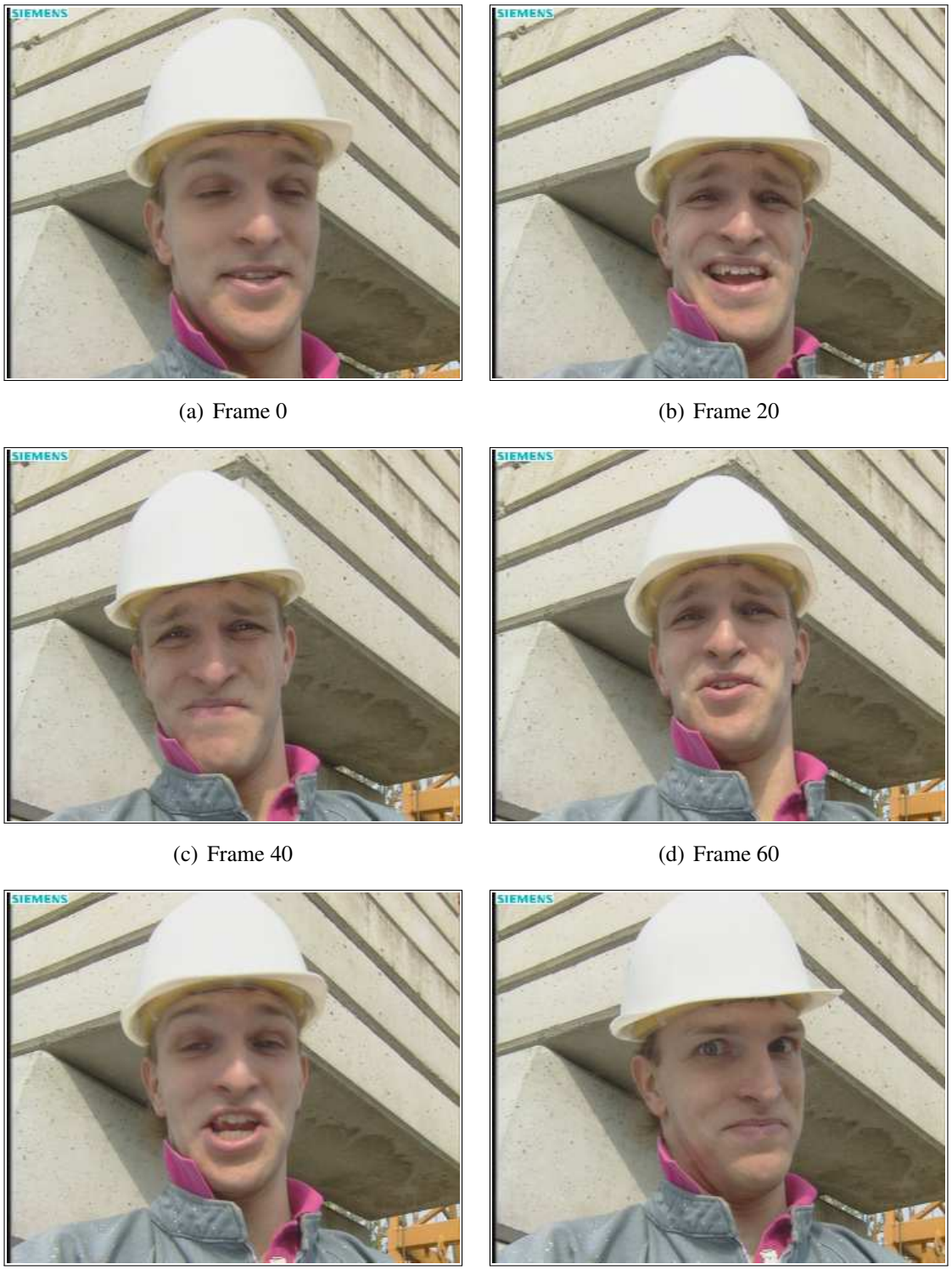
(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.11: Frames from the Calendar video sequence (CIF:352×288).



(a) Frame 0 (b) Frame 20

(c) Frame 40 (d) Frame 60

(e) Frame 80 (f) Frame 100

Figure I.12: Frames from the Foreman video sequence (CIF:352×288).

(a) Frame 0

(b) Frame 20

(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.13: Frames from the Tempete video sequence (CIF:352×288).

(a) Frame 0

(b) Frame 20

(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.14: Frames from the Akiyo video sequence (CIF:352×288).

(a) Frame 0

(b) Frame 20

(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.15: Frames from the Coastguard video sequence (CIF:352×288).

(a) Frame 0

(b) Frame 20

(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.16: Frames from the Container video sequence (CIF:352×288).

(a) Frame 0 (b) Frame 20

(c) Frame 40 (d) Frame 60

(e) Frame 80 (f) Frame 100

Figure I.17: Frames from the Mobcal video sequence (720p:1280×720).

(a) Frame 0

(b) Frame 20

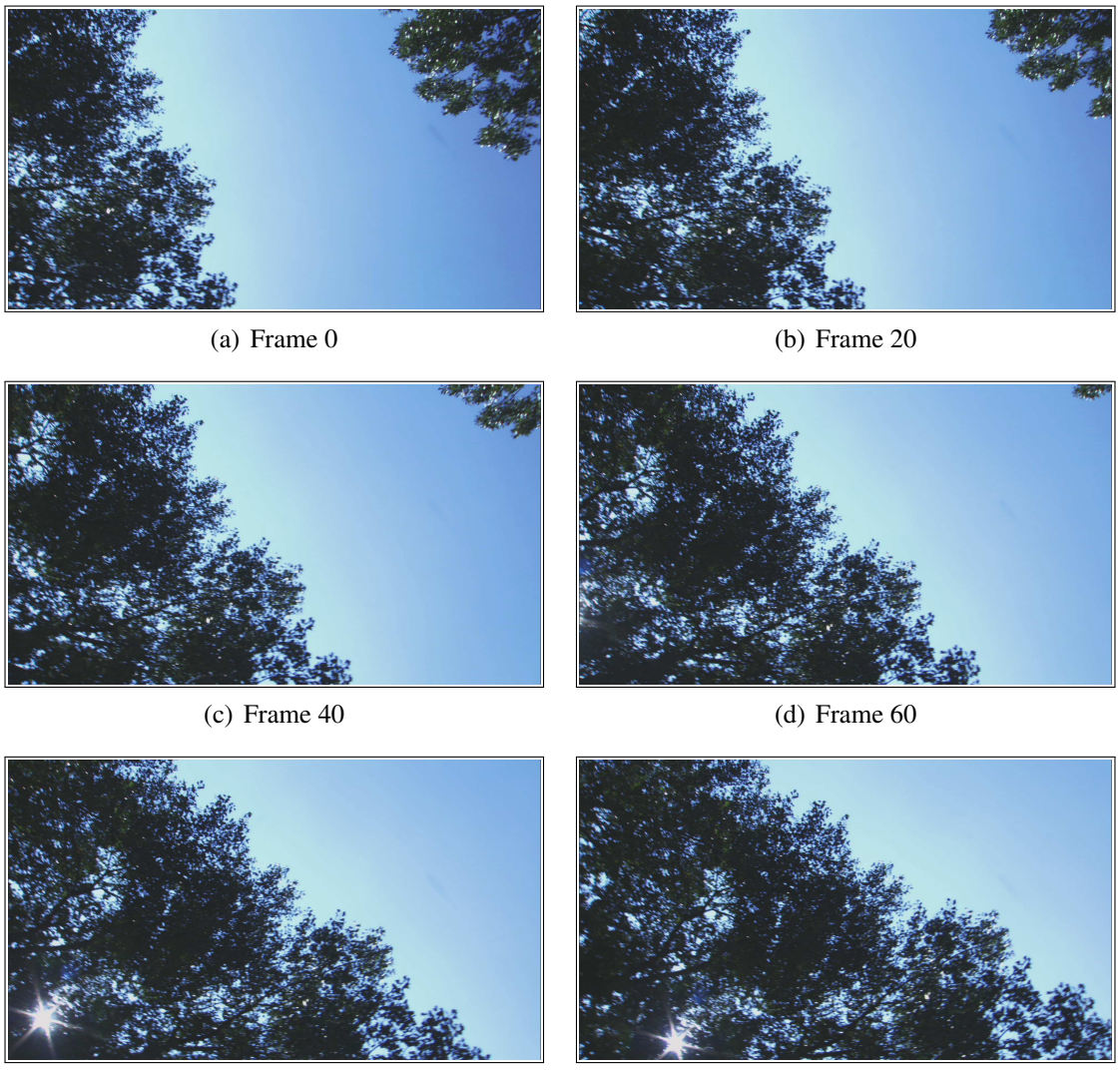
(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.18: Frames from the Old Town Cross video sequence (720p:1280×720).

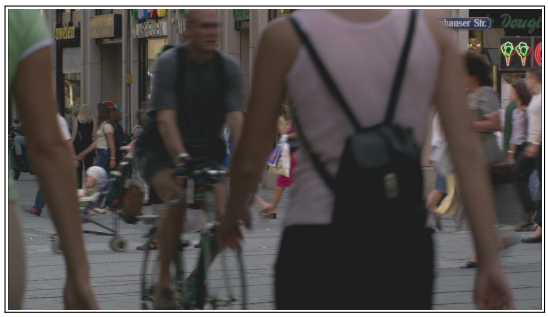


(a) Frame 0 (b) Frame 20

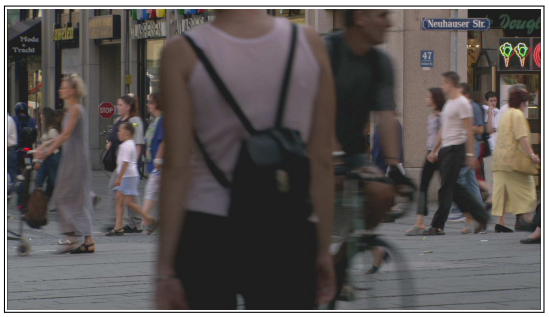
(c) Frame 40 (d) Frame 60

(e) Frame 80 (f) Frame 100

Figure I.19: Frames from the Blue Sky video sequence (1080p:1920×1080).



(a) Frame 0



(b) Frame 20

(c) Frame 40

(d) Frame 60

(e) Frame 80

(f) Frame 100

Figure I.20: Frames from the Pedestrian video sequence (1080p:1920×1080).

(a) Frame 0 (b) Frame 20

(c) Frame 40 (d) Frame 60

(e) Frame 80 (f) Frame 100

Figure I.21: Frames from the Rush Hour video sequence (1080p:1920×1080).

Appendix J

Published papers

J.1 Published papers

J.1.1 Published journal papers

- Francisco, N.C.; Rodrigues, N.M.M.; Da Silva, E.A.B.; De Carvalho, M.B.; De Faria, S.M.M.; Silva, V.M.M.; "Scanned Compound Document Encoding Using Multiscale Recurrent Patterns ", *Image Processing*, *IEEE Transactions on*, Vol. 19, No. 10, pp. 2712 - 2724, October, 2010. doi: 10.1109/TIP.2010.2049181

- Francisco, N.C.; Rodrigues, N.M.M. ; Da Silva, E.A.B.; De Carvalho, M.B.; De Faria, S.M.M.; "Efficient Recurrent Pattern Matching Video Coding", *Circuits and Systems for Video Technology*, *IEEE Transactions on*, Vol. 22, No. 8, pp. 1161 - 1173, August, 2012. doi: 10.1109/TCSVT.2012.2197079

- Francisco, N. C.; Rodrigues, N.M.M. ; Da Silva, E.A.B.; De Faria, S.M.M.; "A Generic Post Deblocking Filter for Block Based Image Compression Algorithms", *Signal Processing: Image Communication*, Vol. 27, No. 9, pp. 985-997, October 2012. doi: 10.1016/j.image.2012.05.005

J.1.2 Published conference papers

- Francisco, N. C.; Rodrigues, N. M. M.; da Silva, E. A. B.; de Carvalho, M. B.; de Faria, S. M. M.; da Silva, V. M. M.; Reis, M. J. C. S.; , "Multiscale recurrent pattern image coding with a flexible partition scheme", *Proceedings of the IEEE International Conference on Image Processing, ICIP'08*, pp.141-144, S.Diego, California, USA, October 2008. doi: 10.1109/ICIP.2008.4711711

- Francisco, N. C.; Rodrigues, N. M. M. ; Da Silva, E. A. B.; De Carvalho, M. B.; De Faria, S. M. M.; Silva, V. M. M.; Reis, M. C. R.; "Casamento Aproximado de

Padrões Multiescala com Segmentação Flexível e Treino do Dicionário", *Proceedings Simpósio Brasileiro das Telecomunicações, SBrT'08* Rio de Janeiro, Brazil, September 2008.

- Francisco, N. C.; Sardo, R. R.; Rodrigues, N. M. M.; Da Silva, E. A. B.; De Carvalho, M. B.; De Faria, S. M. M.; Silva, V. M. M.; Reis, M. C. R.; "A compound image encoder based on the multiscale recurrent pattern algorithm", *Proceedings International Conference on Signal Processing and Multimedia Applications, SIGMAP'08*, Porto, Portugal, July 2008.

- Da Silva, E. A. B.; Lovisolo, L.; De Carvalho, M. B.; Rodrigues, N. M. M.; Filho, E. B. L.; Tcheou, M. P.; De Faria, S. M. M.; Francisco, N. C.; Graziosi, D. B.; "Compressão de Sinais Além das Transformadas" - Mini-curso ministrado no *XXVII Simpósio Brasileiro de Telecomunicações - SBrT 2009*, Blumenau, Brasil, November, 2009.

- Francisco, N. C.; Rodrigues, N. M. M.; da Silva, E. A. B.; de Carvalho, M. B.; de Faria, S. M. M.; , "Computational complexity reduction methods for multiscale recurrent pattern algorithms", *Proceedings IEEE International Conference on Computer as a Tool, EUROCON 2011*, pp.1-4, 27-29 April 2011. doi: 10.1109/EUROCON.2011.5929396

- Francisco, N. C.; Zaghetto, A.; Macchiavello, B.; da Silva, E. A. B.; Lima-Marques, M.; Rodrigues, N. M. M.; de Faria, S. M. M.; , "Compression of touchless multiview fingerprints," *Proceedings IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications, BIOMS 2011*, pp.1-5, 28-28 September 2011. doi: 10.1109/BIOMS.2011.6052380

J.1.3 Submitted conference papers

- Francisco, N. C.; Rodrigues, N. M. M.; da Silva, E. A. B.; de Carvalho, M. B.; de Faria, S. M. M.; , "Video Compression Using 3D Multiscale Recurrent Patterns," Submitted to *IEEE International Symposium on Circuits and Systems, ISCAS 2013*.

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] DE CARVALHO, M. B. *Compression of Multidimensional Signals Based on Recurrent Multiscale Patterns*. Tese de Doutorado, COPPE - Univ. Fed. do Rio de Janeiro, April 2001, *http://www.lps.ufrj.br/profs/eduardo/teses/murilo-carvalho.ps.gz*.

[3] DE CARVALHO, M. B., DA SILVA, E. A. B., FINAMORE, W. "Multidimensional signal compression using multiscale recurrent patterns", *Elsevier Signal Processing*, v. 82, n. 11, pp. 1559–1580, November 2002. ISSN: 0165-1684. doi: 10.1016/S0165-1684(02)00302-X.

[4] RODRIGUES, N. M. M. *Multiscale Recurrent Pattern Matching Algorithms for Image and Video Coding*. Tese de Doutorado, Faculdade de Ciências e Tecnologia - Universidade de Coimbra, October 2008.

[5] 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, S.Diego, CA, USA, October 2008. doi: 10.1109/ICIP.2008.4711711.

[6] GRAZIOSI, D. B. *Contribuições a Compressão de Imagens Com e Sem Perdas Utilizando Recorrência de Padrões Multiescalas*. Tese de Doutorado, COPPE - Univ. Fed. do Rio de Janeiro, April 2011.

[7] 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. doi: 10.1109/ICIP.2006.312585.

[8] DUARTE, M. H. V., DE CARVALHO, M. B., DA SILVA, E. A. B., et al. "Multiscale recurrent patterns applied to stereo image coding", *Circuits and Systems*

for Video Technology, IEEE Transactions on, v. 15, n. 11, pp. 1434–1447, November 2005. ISSN: 1051-8215. doi: 10.1109/TCSVT.2005.856926.

[9] FRANCISCO, N. C., ZAGHETTO, A., MACCHIAVELLO, B., et al. "Compression of touchless multiview fingerprints". In: *Proceedings of the IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications, BIOMS '11*, pp. 1–5, Milan, Italy, September 2011. doi: 10.1109/BIOMS.2011.6052380.

[10] FILHO, E. B. L., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "ECG Signal Compression Based on Dc Equalization and Complexity Sorting", *Biomedical Engineering, IEEE Transactions on*, v. 55, n. 7, pp. 1923–1926, July 2008. ISSN: 0018-9294. doi: 10.1109/TBME.2008.919880.

[11] FILHO, E. B. L., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "On ECG signal compression with one-dimensional multiscale recurrent patterns allied to pre-processing techniques", *Biomedical Engineering, IEEE Transactions on*, v. 56, n. 3, pp. 896–900, March 2009. ISSN: 0018-9294. doi: 10.1109/TBME.2008.2005939.

[12] FILHO, E. B. L. *Aplicações em Codificação de Sinais: O Casamento Aproximado de Padrões Multiescalas e a Codificação Distribuída de Electrocardiograma*. Tese de Doutorado, COPPE - Univ. Fed. do Rio de Janeiro, November 2008.

[13] 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", *VLBV05 - International Workshop on Very Low Bitrate Video*, September 2005.

[14] 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". In: *Proceedings of SPIE - Applications of Digital Image Processing XXIX*, v. 6312, August 2006. doi: 10.1117/12.680355.

[15] 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. 245–248, Genoa, Italy, September 2005. doi: 10.1109/ICIP.2005.1530037.

[16] SULLIVAN, G. J., OHM, J.-R. "Recent developments in standardization of high efficiency video coding (HEVC)". In: *Proceedings of SPIE - Applications of*

Digital Image Processing XXXIII,, v. 7798, August 2010. doi: 10.1117/12.863486.

[17] ZIV, J., LEMPEL, A. "A Universal algorithm for sequential data compression", *Information Theory, IEEE Transactions on*, v. 23, n. 3, pp. 337–343, 1977. ISSN: 0018-9448. doi: 10.1109/TIT.1977.1055714.

[18] ZIV, J., LEMPEL, A. "Compression of individual sequences via variable-rate coding", *Information Theory, IEEE Transactions on*, v. 24, n. 5, pp. 530–536, September 1978. ISSN: 0018-9448. doi: 10.1109/TIT.1978.1055934.

[19] RODEH, M., PRATT, V., EVEN, S. "Linear algorithm for data compression via string matching", *Journal of the ACM*, v. 28, n. 1, pp. 16–24, January 1981. ISSN: 0004-5411. doi: 10.1145/322234.322237.

[20] STORER, J., SYZMANSKI, T. "Data compression via textual substitution", *Journal of the ACM*, v. 29, n. 4, pp. 928–951, October 1982. ISSN: 0004-5411. doi: 10.1145/322344.322346.

[21] BELL, T. "Better OPM/L text compression", *Communications, IEEE Transactions on*, v. 34, n. 12, pp. 1176–1182, December 1986. ISSN: 0090-6778. doi: 10.1109/TCOM.1986.1096485.

[22] BRENT, R. "A linear algorithm for data compression", *Australian Computer Journal*, v. 19, n. 2, pp. 64–68, May 1987.

[23] WELCH, T. A. "A technique for high performance data compression", *IEEE Computer*, v. 17, n. 6, pp. 8–19, June 1984. ISSN: 0018-9162. doi: 10.1109/MC.1984.1659158.

[24] MILLER, V., WEGMAN, M. "Variations on a scheme by Ziv and Lempel", *Combinatorial Algorithms on Words, NATO ASI Series*, v. F12, pp. 131–140, 1984.

[25] JAKOBSSON, M. "Compression of character strings by an adaptive dictionary", *BIT Computer Science and Numerical Mathematics*, v. 4, n. 25, pp. 593–603, December 1985. ISSN: 0006-3835. doi: 10.1007/BF01936138.

[26] TISCHER, P. "A modified Lempel-Ziv-Welch data compression scheme", *Australian Computer Science Communications*, v. 9, n. 1, pp. 262–272, 1987.

[27] FIALA, E., GREENE, D. "Data compression with finite windows", *Communications of the ACM*, v. 32, n. 4, pp. 490–505, April 1989. ISSN: 0001-0782. doi: 10.1145/63334.63341.

[28] GERSHO, A., GRAY, R. M. *Vector quantization and signal compression*. Norwell, MA, USA, Kluwer Academic Publishers, 1991. ISBN: 0-7923-9181-0.

[29] BRITTAIN, N. J., EL-SAKKA, M. R. "Grayscale true two-dimensional dictionary-based image compression", *Journal of Visual Communication and Image Representation*, v. 18, n. 1, pp. 35–44, February 2007. ISSN: 1047-3203. doi: 10.1016/j.jvcir.2006.09.001.

[30] ISO/IEC JTC1/SC29/WG1 N1545. "JBIG2 Final Draft International Standard", December 1999.

[31] YE, Y., COSMAN, P. "Dictionary design for text image compression with JBIG2", *Image Processing, IEEE Transactions on*, v. 10, n. 6, pp. 818–828, June 2001. ISSN: 1057-7149. doi: 10.1109/83.923278.

[32] ATALLAH, M. J., GENIN, Y., SZPANKOWSKI, W. "Pattern matching image compression: algorithmic and empirical results", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 21, n. 7, pp. 614–627, July 1999. ISSN: 0162-8828. doi: 10.1109/34.777372.

[33] DUDEK, G., BORYS, P., GRZYWNA, Z. J. "Lossy dictionary-based image compression method", *Image and Vision Computing*, v. 25, n. 6, pp. 883–889, June 2007. ISSN: 0262-8856. doi: 10.1016/j.imavis.2006.07.001.

[34] 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*, v. 4, pp. 2491–2494, Detroit, Michingan, USA, May 1995. doi: 10.1109/ICASSP.1995.480054.

[35] EFFROS, M., CHOU, P. A., GRAY, R. M. "One-pass adaptive universal vector quantization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '94*, v. 5, pp. 625–628, Adelaide, South Australia, April 1994. doi: 10.1109/ICASSP.1994.389437.

[36] NEFF, R., ZAKHOR, A. "Matching pursuit video coding .I. Dictionary approximation", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 12, n. 1, pp. 13–26, January 2002. ISSN: 1051-8215. doi: 10.1109/76.981842.

[37] NEFF, R., ZAKHOR, A. "Matching-pursuit video coding .II. Operational models for rate and distortion", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 12, n. 1, pp. 27–39, January 2002. ISSN: 1051-8215. doi: 10.1109/76.981843.

[38] CAETANO, R., DA SILVA, E. A. B., CIANCIO, A. G. "Matching pursuits video coding using generalized bit-planes". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '02*, v. 3, pp. 677–680, Rochester, NY, USA, September 2002. doi: 10.1109/ICIP.2002.1039061.

[39] ALZINA, M., SZPANKOWSKI, W., GRAMA, A. "2D-pattern matching image and video compression: theory, algorithms, and experiments", *Image Processing, IEEE Transactions on*, v. 11, n. 3, pp. 318–331, March 2002. ISSN: 1057-7149. doi: 10.1109/83.988964.

[40] FISCHER, Y. *Fractal Image Compression*. 1st ed. New York, NY, Springer Verlag, 1992. ISBN: 0-3879-4211-4.

[41] LUCAS, L. F. R., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Stereo image coding using dynamic template-matching prediction". In: *Proceeding of the IEEE International Conference on Computer as a Tool, EUROCON '11*, pp. 1–4, Lisbon, Portugal, April 2011. doi: 10.1109/EUROCON.2011.5929292.

[42] ABRAHÃO, G. C. B. *Codificação de Voz Utilizando Recorrência de Padrões Multiescala*. Tese de Doutorado, COPPE - Universidade Federal do Rio de Janeiro, November 2005.

[43] PINAGÉ, F. S. *Codificação de Voz Usando Recorrência de Padrões Multiescalas*. Tese de Doutorado, COPPE - Univ. Fed. do Rio de Janeiro, September 2011.

[44] ORTEGA, A., RAMCHANDRAN, K. "Rate-distortion methods for image and video compression", *IEEE Signal Processing Magazine*, v. 15, n. 6, pp. 23–50, November 1998. ISSN: 1053-5888. doi: 10.1109/79.733495.

[45] ITU-T, ISO/IEC JTC 1. *Advanced video coding for generic audio-visual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 2: Jan. 2004, Version 3: Sept 2004, Version 4: July 2005.*

[46] GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Improving multiscale recurrent pattern image coding with least-squares prediction". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, pp. 2813–2816, Cairo, Egypt, November 2009. doi: 10.1109/ICIP.2009.5414219.

[47] LI, X. "Least-square prediction for backward adaptive video coding", *EURASIP Journal on Applied Signal Processing*, v. 2006, n. 1, pp. 126–126, January 2006. ISSN: 1110-8657. doi: 10.1155/ASP/2006/90542.

[48] WITTEN, I. H., NEAL, R. M., CLEARY, J. G. "Arithmetic Coding for Data Compression", *Communications of the ACM*, v. 30, n. 6, pp. 520–540, June 1987. ISSN: 0001-0782. doi: 10.1145/214762.214771.

[49] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "On dictionary adaptation for recurrent pattern image coding", *Image Processing, IEEE Transactions on*, v. 17, n. 9, pp. 1640–1653, September 2008. ISSN: 1057-7149. doi: 10.1109/TIP.2008.2001392.

[50] TAUBMAN, D. S., MARCELIN, M. W. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. 2nd ed. Norwell, Massachusetts, Kluwer Academic Publishers, 2001. ISBN: 1-9933-9639-X.

[51] WIEGAND, T., SULLIVAN, G., BJØNTEGAARD, G., et al. "Overview of the H.264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, pp. 560–576, July 2003. ISSN: 1051-8215. doi: 10.1109/TCSVT.2003.815165.

[52] SAID, A., PEARLMAN, W. A. "A new fast and efficient image codec based on set partitioning in hierarchical trees", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 6, pp. 243–250, June 1996. ISSN: 1051-8215. doi: 10.1109/76.499834.

[53] PENNEBAKER, W., MITCHEL, J. *JPEG: Still Image Data Compression Standard*. 1st ed. Norwell, MA, USA, Van Nostrand Reinhold, 1992. ISBN: 0-4420-1272-1.

[54] MARPE, D., WIEGAND, T., GORDON, S. "H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas". In: *Proceedings IEEE International Conference on Image Processing, ICIP '05*, v. 1, pp. 593–596, Genoa, Italy, September 2005. doi: 10.1109/ICIP.2005.1529820.

[55] KOU, W. *Digital Image Compression Algorithms and standards*. Kluwer Academic Publishers, 1995. ISBN: 0-7923-9626-X.

[56] HUTTENLOCHER, D., FELZENSZWALB, P., RUCKLIDGE, W. "DigiPaper: A versatile color document image representation". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '99*, v. 1, pp. 219–223, Kobe, Japan, October 1999. doi: 10.1109/ICIP.1999.821601.

[57] HAFFNER, P., BOTTOU, L., HOWARD, P., et al. "DjVu : Analyzing and compressing scanned documents for internet distribution". In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR'99*,

pp. 625–628, Bangalore, India, September 1999. doi: 10.1109/ICDAR.1999.791865.

[58] BOTTOU, L., HAFFNER, P., HOWARD, P., et al. "High quality document image compression using DjVu", *Journal of Electronic Imaging*, v. 7, n. 3, pp. 410–425, July 1998. doi: 10.1117/1.482609.

[59] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8). "JPEG 2000 Part I Final Committee Draft Version 1.0", 2001.

[60] ZAGHETTO, A., DE QUEIROZ, R. L. "Segmentation-driven compound document coding based on H.264/AVC-Intra", *Image Processing, IEEE Transactions on*, v. 16, n. 7, pp. 1755–1760, July 2007. ISSN: 1057-7149. doi: 10.1109/TIP.2007.899036.

[61] ZAGHETTO, A., DE QUEIROZ, R. L. "Iterative pre- and post-processing for MRC layers of scanned documents". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '08*, pp. 1009–1012, S.Diego, CA, USA, October 2008. doi: 10.1109/ICIP.2008.4711928.

[62] ITU-T RECOMMENDATION T.44. "Mixed Raster Content (MRC)", *Study Group-8 Contribution*, 1998.

[63] SAID, A., DRUKAREV, A. "Simplified segmentation for compound image compression". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '99*, v. 1, pp. 229–233, Kobe, Japan, October 1999. doi: 10.1109/ICIP.1999.821603.

[64] CHENG, D., BOUMAN, C. "Document compression using rate-distortion optimized segmentation", *Journal of Electronic Imaging*, v. 10, n. 2, pp. 460–474, April 1999. doi: 10.1117/1.1344590.

[65] KONSTANTINIDES, K., TRETTER, D. "A JPEG variable quantization method for compound documents", *Image Processing, IEEE Transactions on*, v. 9, n. 7, pp. 1282–1287, July 2000. ISSN: 1057-7149. doi: 10.1109/83.847840.

[66] DING, W., LU, Y., WU, F. "Enable Efficient Compound Image Compression in H.264/AVC Intra Coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '07*, v. 2, pp. 337–340, San Antonio, Texas, October 2007. doi: 10.1109/ICIP.2007.4379161.

[67] LIN, T., HAO, P. "Compound image compression for real-time computer screen image transmission", *Image Processing, IEEE Transactions on*, v. 14, n. 8,

pp. 993–1005, August 2005. ISSN: 1057-7149. doi: 10.1109/TIP.2005.849776.

[68] PAN, Z., SHEN, H., LU, Y., et al. "Browser-friendly hybrid codec for compound image compression". In: *Proceedings IEEE International Symposium on Circuits and Systems, ISCAS '11*, pp. 101–104, Rio de Janeiro, Brazil, May 2011. doi: 10.1109/ISCAS.2011.5937511.

[69] DE QUEIROZ, R. L., FAN, Z., TRAN, T. "Optimizing block-thresholding segmentation for multi-layer compression of compound images", *Image Processing, IEEE Transactions on*, v. 9, n. 9, pp. 1461–1471, September 2000. ISSN: 1057-7149. doi: 10.1109/83.862619.

[70] DE QUEIROZ, R. L. "On data-filling algorithms for MRC layers". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '00*, v. 2, pp. 586–589, Vancouver, Canada, September 2000. doi: 10.1109/ICIP.2000.899503.

[71] BOTTOU, L., PIGEON, S. "Lossy compression of partially masked still images". In: *Proceedings of the Data Compression Conference, DCC'98*, p. 528, Snowbird, Utah, USA, March 1998. doi: 10.1109/DCC.1998.672238.

[72] SOILLE, P. *Morphological Image Analysis*. Springer, 2007. ISBN: 3-5406-5671-5.

[73] DING, W., LIU, D., HE, Y., et al. "Block-based fast compression for compound images", *Proceedings of the Internacional Conference in Multimedia & Expo*, pp. 809–812, Toronto, Ontario, Canada, July 2006. doi: 10.1109/ICME.2006.262624.

[74] OTSU, N. "A threshold selection method from gray-level histograms", *Systems, Man, and Cybernetics, IEEE Transactions on*, v. 9, n. 1, pp. 62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

[75] LAN, C., SHI, G., WU, F. "Compress Compound Images in H.264/MPGE-4 AVC by Exploiting Spatial Correlation", *Image Processing, IEEE Transactions on*, v. 19, n. 4, pp. 946–957, April 2010. ISSN: 1057-7149. doi: 10.1109/TIP.2009.2038636.

[76] ZAGHETTO, A. *Compressão de Documentos compostos utilizando o H.264/AVC-Intra*. Tese de Doutorado, Faculdade de Tecnologia - Universidade de Brasília, May 2009.

[77] HTTP://WWW.LIZADTECH.COM. *Document Express with DjVu, Enterprise Edition - LizardTech, a Celartem Company*.

[78] LIST, P., JOCH, A., LAINEMA, J., et al. “Adaptive deblocking filter”, *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, pp. 614–619, July 2003. ISSN: 1051-8215. doi: 10.1109/TCSVT.2003.815175.

[79] 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: 3-540-33578-1, S.-V. I. (Ed.), *Proceedings of the 9th International Workshop on Visual Content Processing and Representation, VLBV ’05*, v. 3893, pp. 107–114, Sardinia, Italy, September 2006.

[80] HTTP://IPHOME.HHI.DE/SUEHRING/TML/DOWNLOAD/.

[81] WIEGAND, T., SCHWARZ, H., JOCH, A., et al. “Rate-constrained coder control and comparison of video coding standards”, *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, pp. 688–703, July 2003. ISSN: 1051-8215. doi: 10.1109/TCSVT.2003.815168.

[82] MARPE, D., SCHWARZ, H., WIEGAND, T. “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard”, *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, pp. 620–636, July 2003. ISSN: 1051-8215. doi: 10.1109/TCSVT.2003.815173.

[83] FLIERL, M., GIROD, B. “Generalized B pictures and the draft H.264/AVC video-compression standard”, *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, pp. 587–597, July 2003. ISSN: 1051-8215. doi: 10.1109/TCSVT.2003.814963.

[84] BJØNTEGAARD, G. “Calculation of Average PSNR Differences Between RD-curves”, *ITU-T SG 16 Q.6 VCEG, Doc. VCEG-M33*, 2001.

[85] 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 - ConfTele2009*, Santa Maria da Feira, Portugal, May 2009.

[86] JARSKE, T., HAAVISTO, P., DEFE’E, I. “Post-filtering methods for reducing blocking effects from coded images”. In: *Proceedings of the IEEE International Conference on Consumer Electronics. Digest of Technical Papers.*, pp. 218–219, June 1994. doi: 10.1109/ICCE.1994.582234.

[87] NATH, V., HAZARIKA, D., MAHANTA, A. “Blocking artifacts reduction using adaptive bilateral filtering”. In: *Proceedings of the International Conference on Signal Processing and Communications, SPCOM 2010*, pp. 1–5, Bangalore, India, july 2010. doi: 10.1109/SPCOM.2010.5560517.

[88] XIONG, Z., ORCHARD, M., ZHANG, Y. "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 7, n. 2, pp. 433–437, April 1997. ISSN: 1051-8215. doi: 10.1109/76.564123.

[89] CHEN, T., WU, H. R., QIU, B. "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 11, n. 5, pp. 594–602, May 2001. ISSN: 1051-8215. doi: 10.1109/76.920189.

[90] XU, J., ZHENG, S., YANG, X. "Adaptive video-blocking artifact removal in discrete Hadamard transform domain", *Optical Engineering*, v. 45, n. 8, August 2006. doi: 10.1117/1.2280609.

[91] ZAKHOR, A. "Iterative procedures for reduction of blocking effects in transform image coding", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 2, n. 1, pp. 91–95, March 1992. ISSN: 1051-8215. doi: 10.1109/76.134377.

[92] HUANG, Y.-M., LEOU, J.-J., CHENG, M.-H. "A post deblocking filter for H.264 video". In: *Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN 2007*, pp. 1137–1142, Honolulu, Hawaii USA, August 2007. doi: 10.1109/ICCCN.2007.4317972.

[93] KONG, H.-S., VETRO, A., SUN, H. "Edge map guided adaptive post-filter for blocking and ringing artifacts removal". In: *Proceedings of the International Symposium on Circuits and Systems, ISCAS '04*, v. 3, pp. 929–932, Vancouver, Canada, May 2004. doi: 10.1109/ISCAS.2004.1328900.

[94] 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 International Conference on Signal Processing and Multimedia Applications, SIGMAP '06*, pp. 118–125, Setúbal, Portugal, August 2006.

[95] FRAUCHE, A. L. V. "Compressão de Sinais de Radares Meteorológicos Usando o Algoritmo MMP (Multidimensional Multiscale Parser)". Dissertação de Mestrado, Universidade Federal Fluminense, March 2008.

[96] CHABARCHINE, A., CREUTZBURG, R. "3D fractal compression for real-time video". In: *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, ISPA '01*, pp. 570–573, Pula, Croatia, June 2001. doi: 10.1109/ISPA.2001.938693.

[97] SERVAIS, M., DE JAGER, G. "Video compression using the three dimensional discrete cosine transform (3D-DCT)". In: *Proceedings of the 1997 South African Symposium on Communications and Signal Processing, COMSIG '97*, pp. 27–32, Grahamstown, South Africa, September 1997. doi: 10.1109/COMSIG.1997.629976.

[98] FRYZA, T. *Compression of Video Signals by 3D-DCT Transform*. Tese de Doutorado, Institute of Radio Electronics, FEKT Brno, University of Technology, Czech Republic, 2002.

[99] CHAN, R. K. W., LEE, M. C. "3D-DCT quantization as a compression technique for video sequences". In: *Proceedings of the International Conference on Virtual Systems and MultiMedia, VSMM '97*, pp. 188–196, Geneva, Switzerland, September 1997. doi: 10.1109/VSMM.1997.622346.

[100] BOZINOVIC, N., KONRAD, J. "Motion analysis in 3D DCT domain and its application to video coding", *Signal Processing: Image Communication*, v. 20, n. 6, pp. 510–528, July 2005. ISSN: 0923-5965. doi: 10.1016/j.image.2005.03.007.

[101] KARLSSON, G., VETTERLI, M. "Three dimensional sub-band coding of video". In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP '88*, v. 2, pp. 1100–1103, New York City, USA, April 1988. doi: 10.1109/ICASSP.1988.196787.

[102] CHOI, S.-J., WOODS, J. "Motion-compensated 3-D subband coding of video", *Image Processing, IEEE Transactions on*, v. 8, n. 2, pp. 155–167, February 1999. ISSN: 1057-7149. doi: 10.1109/83.743851.

[103] KIM, B.-J., XIONG, Z., PEARLMAN, W. "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 10, n. 8, pp. 1374–1387, December 2000. ISSN: 1051-8215. doi: 10.1109/76.889025.

[104] WANG, A., XIONG, Z., CHOU, P., et al. "Three-dimensional wavelet coding of video with global motion compensation". In: *Proceedings of the Data Compression Conference, DCC '99*, pp. 404–413, Snowbird, Utah, USA, March 1999. doi: 10.1109/DCC.1999.755690.

[105] LIN, T., WANG, S. "Cloudlet-screen computing: A multi-core-based, cloud-computing-oriented, traditional-computing-compatible parallel computing Paradigm for the masses". In: *Proceedings of the IEEE International Conference*

on Multimedia and Expo, ICME '09, pp. 1805–1808, New York City, USA, July 2009. doi: 10.1109/ICME.2009.5202873.

[106] LU, Y., LI, S., SHEN, H. "Virtualized Screen: A third element for cloud-mobile convergence", *Multimedia, IEEE*, v. 18, n. 2, pp. 4–11, February 2011. ISSN: 1070-986X. doi: 10.1109/MMUL.2011.33.

[107] CHANG, T., LI, Y. "Deep shot: A framework for migrating tasks across devices using mobile phone cameras". In: *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'11*, pp. 2163–2172, Vancouver, BC, Canada, May 2011. ISBN: 978-1-4503-0228-9. doi: 10.1145/1978942.1979257.

[108] DE CARVALHO, M. B., DA SILVA, E. A. B., FINAMORE, W. A., et al. "Universal multi-scale matching pursuits algorithm with reduced blocking effect". In: *Proceedings of the International Conference on Image Processing, ICIP '00*, v. 3, pp. 853–856, Vancouver, BC, Canada, September 2000. doi: 10.1109/ICIP.2000.899590.

[109] FINAMORE, W. A., DE CARVALHO, M. B. "Lossy Lempel-Ziv on subband coding of images". In: *Proceedings of the IEEE International Symposium on Information Theory, ISIT '94*, p. 415, Throndheim, Norway, June 1994. doi: 10.1109/ISIT.1994.395030.

[110] DE CARVALHO, M. B., DA SILVA, E. A. B. "A universal multi-dimensional lossy compression algorithm". In: *Proceedings of the International Conference on Image Processing, ICIP '99*, v. 3, pp. 767–771, Kobe, Japan, October 1999. doi: 10.1109/ICIP.1999.817220.

[111] RICHARDSON, I. A. *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd., 2003. ISBN: 0-4708-4837-5.

[112] CAMPBELL, S. L., MEYER, C. D. *Generalized Inverse of Linear Transformations*. Dover Publications, 1991. ISBN: 0-4866-6693-X.

[113] DINIZ, P. S. R., DA SILVA, E. A. B., NETTO, S. L. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2002. ISBN: 0-5217-8175-2.

[114] 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*, Island of Kos, Greece, May 2006. doi: 10.1109/ISCAS.2006.1693739.

[115] RISSANEN, J., LANGDON, G. "Aritmetic Coding", *IBM Journal of Research and Development*, v. 23, n. 2, pp. 149–162, March 1979. ISSN: 0018-8646. doi: 10.1147/rd.232.0149.

[116] FILHO, E. B. L., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Electrocardiographic signal compression using multiscale recurrent patterns", *Circuits and Systems I, IEEE Transactions on*, v. 52, n. 12, pp. 2739–2753, December 2005. ISSN: 1549-8328. doi: 10.1109/TCSI.2005.857873.

[117] HTTP://WWW.KAKADUSOFTWARE.COM.

[118] DE QUEIROZ, R. L., ORTIS, R. S., ZAGHETTO, A., et al. "Fringe benefits of the H.264/AVC". In: *Proceedings of the International Telecomunication Symposium, ITS '06*, pp. 166–170, Fortaleza, Brazil, September 2006. doi: 10.1109/ITS.2006.4433263.

[119] MRAK, M., GRGIC, S., GRGIC, M. "Picture Quality Measures in Image Compression Systems". In: *Proceeding of the IEEE International Conference on Computer as a Tool, EUROCON '03*, v. 1, pp. 233–236, Ljubljana, Slovenia, September 2003. doi: 10.1109/EURCON.2003.1248017.

[120] DE QUEIROZ, R. L. *Compressing Compound Documents,* in *The Document and Image Compression Handbook*. Edited by M. Barni, Marcel-Dekker, 2005. ISBN: 0-8493-3556-6.

[121] ZAGHETTO, A., DE QUEIROZ, R. L. "High quality scanned book compression using pattern matching". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '10*, pp. 2165–2168, Hong Kong, September 2010. doi: 10.1109/ICIP.2010.5653094.

[122] ZAGHETTO, A., MACCHIAVELLO, B., DE QUEIROZ, R. L. "HEVC-based scanned document compression". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '12*, pp. 1–4, Orlando, Florida, USA, September 2012.

[123] LUCAS, L. F. R., RODRIGUES, N. M. M., DE FARIA, S. M. M., et al. "Intra-prediction for color image coding using YUV correlation". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '10*, pp. 1329 –1332, Hong Kong, September 2010. doi: 10.1109/ICIP.2010.5653834.

[124] LEE, S. H., CHO, N. I. "Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, pp. 1037–1040, Cairo, Egypt, November 2009. doi: 10.1109/ICIP.2009.5413727.

[125] JARSKE, T., HAAVISTO, P., DEFEE, I. "Post filtering methods for reducing blocking effects from coded images", *Consumer Electronics, IEEE Transactions on*, v. 40, n. 3, pp. 521–526, August 1994. ISSN: 0098-3063. doi: 10.1109/30.320837.

[126] LIU, Y. "Unified Loop Filter for Video Compression", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 20, n. 10, pp. 1378–1382, October 2010. ISSN: 1051-8215. doi: 10.1109/TCSVT.2010.2077570.

[127] DAI, W., LIU, L., TRAN, T. "Adaptive block-based image coding with pre-/post-filtering". In: *Proceedings of the Data Compression Conference, DCC '05*, pp. 73–82, Snowbird, Utah, USA, March 2005. doi: 10.1109/DCC.2005.11.

[128] GIROD, B. "Motion-compensating prediction with fractional-pel accuracy", *Communications, IEEE Transactions on*, v. 41, n. 4, pp. 604–612, April 1993. ISSN: 0090-6778. doi: 10.1109/26.223785.

[129] LUCAS, L. F. R., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Adaptive least squares prediction for stereo image coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '11*, pp. 2013–2016, Brussels, Belgium, September 2011. doi: 10.1109/ICIP.2011.6115872.

[130] BRUNELLO, D., CALVAGNO, G., MIAN, G., et al. "Lossless compression of video using temporal information", *Image Processing, IEEE Transactions on*, v. 12, n. 2, pp. 132–139, February 2003. ISSN: 1057-7149. doi: 10.1109/TIP.2002.807354.

[131] TIWARI, A., KUMAR, R. "Least-Squares Based Switched Adaptive Predictors for Lossless Video Coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '07*, v. 6, pp. 69–72, San Antonio, Texas, USA, September 2007. doi: 10.1109/ICIP.2007.4379523.

[132] LI, X., ORCHARD, M. T. "Edge-directed prediction for lossless compression of natural images", *Image Processing, IEEE Transactions on*, v. 10, n. 6, pp. 813–817, June 2001. ISSN: 1057-7149. doi: 10.1109/83.923277.