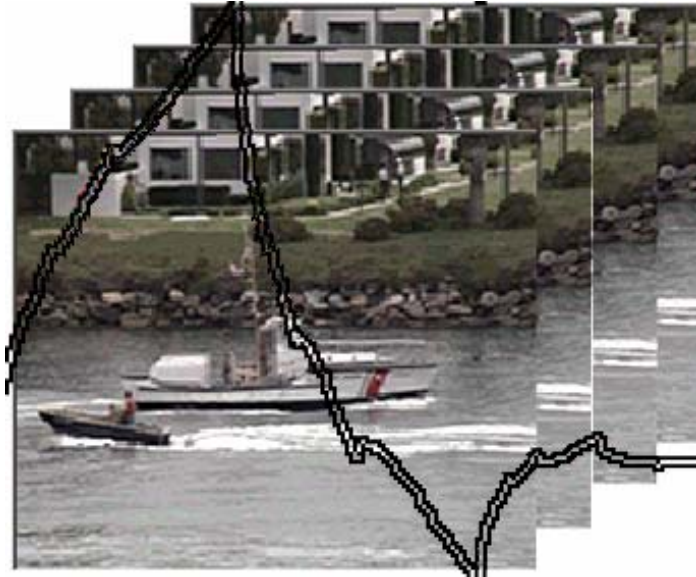




UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



CODIFICAÇÃO DE VÍDEO POR DECOMPOSIÇÃO 3D
BASEADA NA TRANSFORMADA DE ÔNDULAS

José António da Costa Salvado

(Licenciado)

Dissertação para obtenção do grau de Mestre em Engenharia

Electrotécnica e de Computadores

Orientador Científico:

Doutor Leonel Augusto Pires Seabra de Sousa, Prof. Auxiliar, DEEC IST/UTL

Júri:

Presidente: Doutor Leonel Augusto Pires Seabra de Sousa, Prof. Auxiliar, IST/UTL

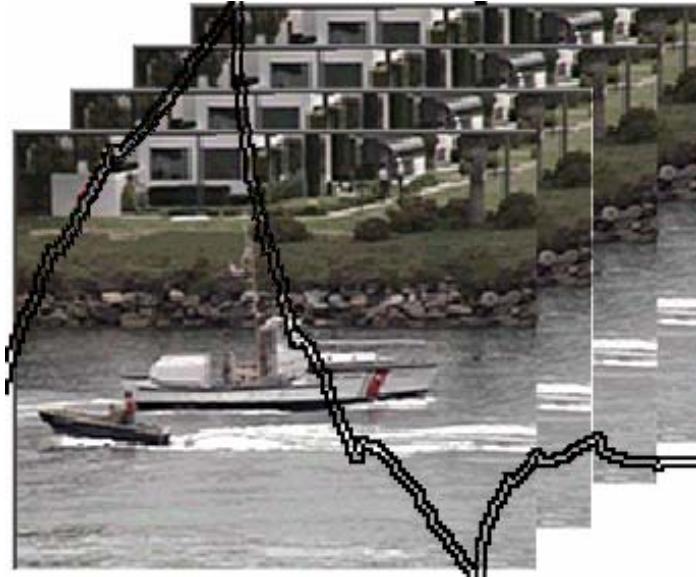
Vogais: Doutor Luís António de Menezes Côrte-Real, Prof. Auxiliar, Fac. Eng./UP

Doutora Maria Paula dos Santos Queluz Rodrigues, Prof. Auxiliar, IST/UTL

Lisboa, Abril de 2002



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



CODIFICAÇÃO DE VÍDEO POR DECOMPOSIÇÃO 3D
BASEADA NA TRANSFORMADA DE ÔNDULAS

José António da Costa Salvado

(Licenciado)

Dissertação para obtenção do grau de Mestre em Engenharia

Electrotécnica e de Computadores

Orientador Científico:

Doutor Leonel Augusto Pires Seabra de Sousa, Prof. Auxiliar, DEEC IST/UTL

Júri:

Presidente: Doutor Leonel Augusto Pires Seabra de Sousa, Prof. Auxiliar, IST/UTL

Vogais: Doutor Luís António de Menezes Côrte-Real, Prof. Auxiliar, Fac. Eng./UP

Doutora Maria Paula dos Santos Queluz Rodrigues, Prof. Auxiliar, IST/UTL

Lisboa, Abril de 2002

Tese realizada sob orientação do
Doutor Leonel Augusto Pires Seabra de Sousa
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores do Instituto Superior Técnico

Dedico esta Tese:

à Ana Catarina;

à Maria de Lurdes.

“On fait la science avec des faits, comme une maison avec des pierres; mais une accumulation des faits n’est pas plus une science qu’un tas des pierres n’est une maison.”

(A ciência constrói-se com factos, tal como uma casa com pedras; mas um conjunto de factos não é uma ciência, nem um amontoado de pedras é uma casa.)

Henri Poincaré, La Science et l’Hypothèse.

Resumo

Neste trabalho foi desenvolvido um codificador/descodificador (CoDec) de vídeo de ritmo binário variável, baseado na decomposição 3D por transformada de ôndulas, sendo o cálculo das transformadas de ôndulas, e das transformadas inversas, realizado através do esquema progressivo (*lifting scheme*).

Analisa-se a aplicação à codificação de vídeo da decomposição de sinais 1D e 2D por bancos de filtros de ôndulas, a sua relação com a análise multiresolução, e as relações de dependência hierárquica dos coeficientes da estrutura resultante da decomposição. Apresenta-se, também, a associação do esquema progressivo aos bancos de filtros e a obtenção dos algoritmos das transformadas pela factorização segundo o esquema progressivo.

Implementaram-se as transformadas (pelo esquema progressivo), para a ôndula Daubechies 2 e para as ôndulas biortogonais (4,4), para explorar a redundância espacial e temporal, aplicando-as à codificação de vídeo.

Apresentam-se resultados experimentais obtidos com o codificador desenvolvido, com base em sequências de vídeo de teste, nomeadamente o factor de compressão, a relação sinal/ruído entre as imagens recuperadas e as imagens originais e a qualidade subjectiva das imagens recuperadas. Discutem-se os resultados obtidos e o tempo necessário para processar as sequências. Propõem-se soluções para melhorar a eficiência do codificador, no que diz respeito ao tempo de processamento e ao factor de compressão. Implementaram-se, também, as transformadas de ôndula num processador digital de sinal com arquitectura VLIW, analisando-se a aceleração de processamento obtida, tendo em vista a codificação de vídeo em tempo real.

Abstract

In this work a scalable bit rate video CoDec was developed, based on the 3D wavelet transform decomposition, using the lifting scheme to calculate the wavelet transform and the inverse wavelet transform.

The application of 1D and 2D signal decomposition using wavelet filter banks to video coding is analysed, and also its relation with multiresolution analysis and the hierarchical dependencies of the coefficients within the structure obtained by decomposition. The lifting scheme relations with filter banks, and the process to obtain the transform algorithms, by factorisation into lifting steps is also shown.

The lifting wavelet transforms were implemented with Daubechies 2 orthogonal wavelet, and biorthogonal (4.4) wavelets, in order to explore the spatial and temporal redundancies, by applying these transforms to video coding.

Experimental results obtained, such as compression ratio, peak signal to noise ratio between recovered and original images and the subjective quality of the recovered images are shown, with respect to the video coder developed, based on video test sequences. The results obtained are discussed, as the time needed to process the sequences. One also proposes solutions to improve performance of the video coder, in what it concerns to processing time and compression ratio. The wavelet transforms were also implemented in a digital signal processor with VLIW architecture, and the obtained speedup in processing time is analysed, in order to meet real time video coding requirements.

Palavras-Chave

Codificação de Vídeo

Transformada de Ôndulas

Esquema Progressivo

Codificação EZW 3D

Ritmo Binário Escalável

Processadores Digitais de Sinal

Keywords

Video Coding

Wavelet Transform

Lifting Scheme

3D EZW Coding

Scalable Bit Rate

Digital Signal Processors

Agradecimentos

Desejo agradecer às pessoas cujas contribuições permitiram enriquecer o meu conhecimento técnico e científico, e às entidades que contribuíram cedendo recursos e/ou meios materiais.

Em particular agradeço:

ao Professor Leonel Augusto de Sousa, meu orientador científico, pelos conhecimentos que me transmitiu, pelo estímulo dos desafios que me colocou e pela disponibilidade na orientação deste trabalho;

à Escola Superior de Tecnologia de Castelo Branco, pela dispensa de serviço docente;

ao Sérgio Barbosa, pela disponibilidade e pelo apoio no uso do *software* Borland C++ Builder;

ao João Gonçalves, pelas tabelas de Huffman e pelo apoio na actividade docente na fase final de escrita desta tese;

à Prof. Arminda Guerra, pelas facilidades concedidas na utilização de recursos informáticos e gráficos.

Por último, uma palavra muito especial de agradecimento à minha família, em particular à Maria de Lurdes e à Ana Catarina, pelo carinho que me dispensaram durante a realização deste trabalho (que subtraiu grande parte da atenção que lhes é devida) e por terem suportado pacientemente as minhas (muitas) “ausências”.

Simbologia

NOTAÇÃO	DEFINIÇÃO
$L^2(\mathbb{R})$	Espaço das funções quadráticas integráveis em \mathbb{R}
$\langle f_x, \phi_y \rangle$	Produto interno das funções f e ϕ
$\psi_{j,k}(\cdot)$	Ôndula (escala j e translação k)
$\phi_{j,k}(\cdot)$	Função de escala (escala j e translação k)
$\hat{\psi}(\cdot)$	Transformada de Fourier da ôndula $\psi(\cdot)$
$\hat{\phi}(\cdot)$	Transformada de Fourier da função de escala $\phi(\cdot)$
$\ \phi\ _{L^2}$	Norma da função ϕ , em que $\phi \in L^2(\mathbb{R})$
$\bigcup_j V_j$	Reunião dos espaços V_j
$\bigcap_j V_j$	Intersecção dos espaços V_j
$V_j \oplus W_j$	Soma directa dos espaços V_j e W_j
$\overline{f(\cdot)}$	Conjugado complexo de $f(\cdot)$
$f(2\cdot)$	Função $f(\cdot)$ afectada de um factor de escala de 2.
$(P_{V_j} f)(x)$	Operador das projecções da função f no espaço V_j
\tilde{h}, h	Filtros de ôndulas do tipo passa-baixo, de análise (\tilde{h}) e de síntese (h)
\tilde{g}, g	Filtros de ôndulas do tipo passa-alto, de análise (\tilde{g}) e de síntese (g)
$\downarrow 2$	Processo de decimação com factor 2
$\uparrow 2$	Processo de interpolação com factor 2

Lista de Acrónimos

CIE	<i>Comission Internationale de L'Eclairage</i>
CIF	<i>Common Intermediate Format</i>
CDF	<i>Cohen-Daubechies-Féveau</i>
CoDec	Codificador e Decodificador
CQF	<i>Conjugate Quadrature Filters</i>
CWT	<i>Continuous Wavelet Transform</i>
DCT	<i>Discrete Cosine Transform</i>
DFT	<i>Discrete Fourier Transform</i>
DMA	<i>Direct Memory Access</i>
DSP	<i>Digital Signal Processor</i>
DST	<i>Discrete Sine Transform</i>
DVD	<i>Digital Versatile Disk</i>
DWT	<i>Discrete Wavelet Transform</i>
EZW	<i>Embedded Zerotree Coding of Wavelet Coefficientes</i>
FIFO	<i>First In – First Out</i>
FIR	<i>Finite Impulse Response Filter</i>
FWT	<i>Fast Wavelet Transform</i>
GOF	<i>Group of Frames</i>
ISO	<i>International Standards Organization</i>
ITU-T	<i>International Telecommunications Union – Telecommunications Standardazation Sector</i>
JPEG	<i>Joint Pictures Experts Group</i>

KLT	<i>Karhunen-Loève Transform.</i>
MBPS	<i>Millions of Bytes per Second</i>
MIPS	<i>Millions of Instructions per Second</i>
MOPS	<i>Millions of Operations per Second</i>
MPEG	<i>Motion Pictures Experts Group</i>
MRA	<i>Multiresolution Analysis</i>
MSE	<i>Mean Square Error</i>
NTSC	<i>National Television System Committee</i>
PAL	<i>Phase Alternate Line</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
QMF	<i>Quadrature Mirror Filters</i>
QCIF	<i>Quarter-CIF</i>
SAQ	<i>Successive-Aproximation Quantization</i>
SPHIT	<i>Set Partitioning in Hierarchical Trees</i>
STFT	<i>Short Time Fourier Transform.</i>
VLC	<i>Variable Length Code</i>
VLIW	<i>Very Large Instruction Word.</i>
VLSI	<i>Very Large Scale of Integration</i>

Conteúdo

Resumo	i
Abstract	ii
Palavras-Chave	iii
Agradecimentos	iv
Simbologia	v
Lista de Acrónimos	vi
Conteúdo	viii
Lista de Figuras	xi
Lista de Tabelas	xv
1. Introdução	1
1.1 Enquadramento	1
1.2 Principais Objectivos do Trabalho	6
1.3 Organização da Tese	7
2. Codificação de Imagem e Vídeo	9
2.1 Introdução	9
2.2 Representação Digital de Imagem e Vídeo	10
2.2.1 Formatos de Vídeo Normalizados	13
2.3 Técnicas de Codificação sem Perda de Informação	14
2.3.1 Codificação de Huffman	15
2.3.2 Codificação Aritmética	17
2.4 Técnicas de Codificação com Perda de Informação	19
2.4.1 Quantificação Escalar e Vectorial	20
2.4.2 Codificação com Base em Transformadas	22
2.4.3 Codificação Hierárquica com Base na Transformada de Óndulas	24
2.4.4 Estimacão e Compensacão de Movimento	25
2.5 Medidas de Eficiência da Codificação de Vídeo	27
2.6 Conclusões	29

3. Transformada de Ôndulas e Esquema Progressivo	31
3.1 Introdução	31
3.2 Ôndulas e Transformada de Ôndulas	33
3.2.1 Ôndulas de Bases Ortonormais	34
3.2.2 Transformada de Ôndulas Contínua e Discreta.....	37
3.3 Análise Multiresolução e Decomposição em Sub-Bandas	38
3.3.1 Análise com Resolução Múltipla.....	38
3.3.2 Decomposição Ortogonal e Biortogonal.....	42
3.4 Transformada de Ôndulas Discreta e Bancos de Filtros.....	43
3.4.1 Filtros de Decomposição e de Reconstrução.....	45
3.4.2 Filtros de Ôndulas de Daubechies.....	48
3.4.3 Filtros de Ôndulas Biortogonais	48
3.5 Transformada de Ôndulas pelo Esquema Progressivo.....	51
3.5.1 Filtros e Polinómios de Laurent.....	53
3.5.2 Representação Polifásica.....	55
3.5.3 Factorização Segundo o Esquema Progressivo	57
3.6 Transformada de Ôndulas Bidimensional.....	59
3.7 Conclusões.....	61
4. Implementação da DWT e DWT^{-1} pelo Esquema Progressivo.	63
4.1 Introdução	63
4.2 DWT e DWT^{-1} para a Ôndula Ortogonal Daubechies 2	64
4.3 DWT e DWT^{-1} para Ôndulas Biortogonais (4,4).....	67
4.4 Implementação das Transformadas de Ôndulas num Processador de Uso Geral....	70
4.5 Principais Características do DSP TMS320C6201.....	76
4.5.1 Execução das Instruções	79
4.5.2 Funcionamento em <i>Cascata</i>	82
4.5.3 Limitações à Eficiência da <i>Cascata</i>	85
4.6 Ambiente de Desenvolvimento para o DSP TMS320C6201	87
4.7 Implementação das Transformadas de Ôndulas no DSP TMS320C6201.....	88
4.8 Conclusões.....	93
5. CoDec de Vídeo Baseado na Transformada de Ôndulas.....	95
5.1 Introdução	95
5.2 Estrutura do CoDec de Vídeo Baseado na DWT 3D	96
5.3 Descrição do Codificador e do Descodificador	98
5.3.1 Formação da Estrutura 3D de Coeficientes em “Árvore”	102

5.3.2	Codificação e Descodificação EZW 3D	105
5.4	Aplicação Desenvolvida para a Codificação e Descodificação de Vídeo	111
5.5	Avaliação do Desempenho do CoDec de Vídeo	114
5.5.1	Factor de Compressão	115
5.5.2	Avaliação da Qualidade das Imagens Recuperadas.....	124
5.5.3	Tempo de Processamento para a EZW 3D e para a EZW ⁻¹ 3D.....	133
5.6	Conclusões.....	139
6.	Conclusões.....	141
6.1	Perspectivas de Trabalho Futuro	143
	Referências Bibliográficas	144
A –	Coeficientes dos Filtros de Ôndulas Ortogonais e Biortogonais	148
B –	Exemplos do Cálculo da DWT e da DWT⁻¹ pelo Esquema Progressivo: Ôndulas DB2 e 9/7.....	153
C –	Tabelas Usadas na Codificação de Huffman.....	164

Lista de Figuras

Número	Pág.
Figura 2.1: Varrimento no modo <i>raster</i> , para amostragem espacial em sinais de vídeo.....	11
Figura 2.2: Exemplo de um esquema de codificação de Huffman.....	16
Figura 2.3: Varrimento em ziguezague para codificação dos coeficientes quantificados da DCT-2D.	24
Figura 3.1: Esquema de decomposição dos sub espaços na MRA.....	39
Figura 3.2: Decomposição em sub-bandas de frequência (oitavas), e sua relação com os espaços ortonormais.....	41
Figura 3.3: Decomposição de um sinal discreto por bancos de filtros.	44
Figura 3.4: Bancos de filtros com decomposição e reconstrução em 2 sub-bandas.	46
Figura 3.5: Esquema de bancos de filtros causais, para decomposição e reconstrução.....	47
Figura 3.6: Diagrama de blocos do esquema progressivo: predição e actualização.....	53
Figura 3.7: Representação polifásica da transformada de ôndulas.....	56
Figura 3.8: Cálculo da transformada de ôndulas directa pelo esquema progressivo.	59
Figura 3.9: Cálculo da transformada de ôndulas inversa pelo esquema progressivo.....	59
Figura 3.10: Esquema de decomposição em sub-bandas e reconstrução, para sinais bidimensionais, por aplicação da DWT e DWT^{-1} 2-D.....	61
Figura 3.11: Transformada de ôndulas de uma imagem com decomposição em 3 níveis.	61
Figura 4.1: Extensão simétrica da sequência nas regiões limite, para o cálculo da DWT^{-1} com os filtros Daubechies 9/7.....	72
Figura 4.2: Imagens originais: tramas 5 a 8 (2º GOF) da sequência "Coastguard".....	73
Figura 4.3: Imagens transformadas no tempo: tramas 5 a 8 da sequência "Coastguard".....	73
Figura 4.4: Imagens transformadas no tempo e no espaço: tramas 5 a 8 da sequência "Coastguard".....	73
Figura 4.5: Diagrama de blocos do DSP TMS320C6201	77

Figura 4.6: Pacote de instruções parcialmente sequencial e totalmente em paralelo: a) codificação; b) paralelismo () na execução.....	79
Figura 4.7: Blocos de processamento de instruções no DSP TMS320C6201.....	81
Figura 4.8: Fases e andares que compõem a <i>cascata</i>	82
Figura 4.9: Exemplo do funcionamento da <i>cascata</i> , na situação de máxima eficiência.....	84
Figura 4.10: Exemplo de perda de eficiência no processamento por paragem da <i>cascata</i>	86
Figura 4.11: Comparação entre o desempenho no TMS320C6201 EVM e no PC equipado com o processador PentiumIII (550 MHz).....	91
Figura 4.12: Comparação entre o desempenho no DSP TMS320C6203 (300 MHz) e no PC equipado com o processador PentiumIII (550 MHz).....	91
Figura 4.13: Código em <i>assembly</i> para o núcleo (<i>kernel</i>) de processamento no cálculo da DWT no tempo (ôndula DB2), e início do epílogo.....	92
Figura 5.1: Diagrama de blocos da constituição do Codificador/Descodificador de vídeo baseado na transformada de ôndulas.....	96
Figura 5.2: Decomposição do GOF em 2 níveis pela DWT no tempo (ôndula DB2).....	99
Figura 5.3: Organização dos coeficientes da sequência resultante da DWT no tempo.....	99
Figura 5.4: Decomposição hierárquica em 3 níveis pela DWT no espaço: a) organização em sub-bandas; b) dependências entre os coeficientes em cada nível.....	100
Figura 5.5: Decomposição de uma imagem pela DWT 2D, em 3 níveis: a) relação de dependência de coeficientes, b) ordem de pesquisa dos coeficientes.....	104
Figura 5.6: Estrutura hierárquica da DWT 3D e relações de dependência.....	105
Figura 5.7: Aspecto da interface gráfica da aplicação desenvolvida para codificação e descodificação de vídeo baseada nas transformadas de ôndulas.....	111
Figura 5.8: Aspecto da janela de diálogo para selecção dos ficheiros a codificar ou a descodificar.....	112
Figura 5.9: Aspecto da aplicação: selecção dos parâmetros a considerar na codificação.....	113
Figura 5.10: Variação do factor de compressão por codificação EZW 3D em função do nível de decisão, com a DWT 9/7 no espaço.....	120
Figura 5.11: Variação do factor de compressão por codificação EZW 3D em função do nível de decisão, com a DWT DB2 no espaço.....	120
Figura 5.12: Variação do factor de compressão por codificação EZW 3D e codificação de Huffman, em função do nível de decisão, com a DWT 9/7 no espaço.....	121

Figura 5.13: Variação do factor de compressão por codificação EZW 3D e codificação de Huffman, em função do nível de decisão, com a DWT DB2 no espaço.	121
Figura 5.14: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Akiyo" (classe A).....	122
Figura 5.15: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Foreman" (classe B).	122
Figura 5.16: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Mobile & Calendar" (classe C).	123
Figura 5.17: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Children" (classe E).....	123
Figura 5.18: Valores de PSNR da luminância, para a sequência "Akiyo" com a DWT 9/7, para vários níveis de decisão.	126
Figura 5.19: Valores de PSNR da luminância, para a sequência "Akiyo" com a DWT DB2, para vários níveis de decisão.	126
Figura 5.20: Valores de PSNR da luminância, para a sequência "News" com a DWT 9/7, para vários níveis de decisão.	127
Figura 5.21: Valores de PSNR da luminância, para a sequência "News" com a DWT DB2, para vários níveis de decisão.	127
Figura 5.22: Valores de PSNR da luminância, para a sequência "Mobile & Calendar" com a DWT 9/7, para vários níveis de decisão.	128
Figura 5.23: Valores de PSNR da luminância, para a sequência "Mobile & Calendar" com DWT DB2, para vários níveis de decisão.	128
Figura 5.24: Valores de PSNR da luminância, para a sequência "Children" com DWT 9/7, para vários níveis de decisão.	129
Figura 5.25: Valores de PSNR da luminância, para a sequência "Children" com DWT DB2, para vários níveis de decisão.	129
Figura 5.26: Imagens 153 a 156 (um GOF) para a sequência "Foreman": a) imagens originais; b) imagens resultantes de codificação com DWT_9/7 e nível 16; c) imagens resultantes da codificação com DWT_9/7 e nível 64.....	130
Figura 5.27: Imagens 12, 14, 16 e 18 para a sequência "Children": a) imagens originais; b) imagens resultantes da codificação com DWT_9/7 e nível 32; c) imagens resultantes da codificação com DWT_9/7 e nível 128.	131
Figura 5.28: Sequência "Akiyo": a) Imagem 26 original; b) imagem 26 recuperada da codificação com DWT 9/7 e nível 32; c) imagem 26 recuperada da codificação com DWT DB2 e nível 32. Sequência "Mobile & Calendar": d) imagem 47	

original; e) imagem 47 recuperada da codificação com DWT 9/7 e nível 64 e, f) imagem 47 recuperada da codificação com DWT 9/7 e nível 128.	132
Figura 5.29: Tempo de processamento: EZW^{-1} 3D <i>vs.</i> EZW 3D (9/7), no caso mais favorável.....	136
Figura 5.30: Tempo de processamento: EZW^{-1} 3D <i>vs.</i> EZW 3D (9/7), no caso intermédio..	137
Figura 5.31: Tempo de processamento: EZW^{-1} 3D <i>vs.</i> EZW 3D (9/7), no caso mais desfavorável.....	137
Figura 5.32: Tempo de processamento da EZW 3D: (9/7) <i>vs.</i> (DB2), no caso mais favorável.....	138
Figura 5.33: Tempo de processamento da EZW 3D: (9/7) <i>vs.</i> (DB2), no caso intermédio. ...	138
Figura 5.34: Tempo de processamento da EZW 3D: (9/7) <i>vs.</i> (DB2), no caso mais desfavorável.....	138

Lista de Tabelas

Número	Pág.
Tabela 2.1: Resolução espacial para os formatos de vídeo digital normalizados, de acordo com as normas H.261 e H.263.	14
Tabela 2.2: Exemplo da organização de sub intervalos na codificação aritmética.	17
Tabela 2.3: Exemplo de esquema de codificação aritmética.	19
Tabela 4.1: Tempos aproximados de processamento necessários ao cálculo da DWT e da DWT^{-1} , para grupos de 4 imagens.	74
Tabela 4.2: Tipos e formatos dos dados suportados pelo DSP TMS320C6201.	78
Tabela 4.3: Operações realizadas nas diversas unidades funcionais.	82
Tabela 4.4: Descrição das operações executadas nas fases que compõem a <i>cascata</i>	83
Tabela 4.5: Andares da fase de execução e latência associada, por tipo de instrução.	85
Tabela 4.6: Número de ciclos de relógio necessários ao processamento da DWT e da DWT^{-1} , para grupos de 4 imagens.	89
Tabela 4.7: Tempos aproximados de processamento para grupos de 4 imagens, considerando as versões do DSP TMS320C62x a 167 e 300 MHz, e uma possível versão a 550 MHz.	90
Tabela 5.1: Sequências de vídeo para teste: classificação, características, e sequências consideradas.	114
Tabela 5.2: Factor de compressão para a sequência "Akiyo" (200 tramas).	116
Tabela 5.3: Factor de compressão para a sequência "Container Ship" (200 tramas).	116
Tabela 5.4: Factor de compressão para a sequência "Foreman" (200 tramas).	116
Tabela 5.5: Factor de compressão para a sequência "News" (200 tramas).	117
Tabela 5.6: Factor de compressão para a sequência "Table Tennis" (200 tramas).	117
Tabela 5.7: Factor de compressão para a sequência "Mobile & Calendar" (200 tramas).	117
Tabela 5.8: Factor de compressão para a sequência "Children" (200 tramas).	118

Tabela 5.9: Tempo médio, em segundos, de codificação e descodificação EZW 3D, para grupos de 4 tramas (GOF), e para a DWT no espaço com os filtros 9/7.....	135
Tabela 5.10: Tempo médio, em segundos, de codificação e descodificação EZW 3D, para grupos de 4 tramas (GOF), e para a DWT no espaço com a ôndula DB2.	135
Tabela A.1: Coeficientes dos filtros passa-baixo das ôndulas ortogonais de suporte compacto de Daubechies.	151
Tabela A.2: Coeficientes dos filtros passa-baixo de análise e de síntese, para ôndulas biortogonais do tipo <i>spline</i>	152

Capítulo 1

INTRODUÇÃO

Neste capítulo faz-se o enquadramento geral desta dissertação na área da codificação de vídeo, introduz-se as técnicas de codificação em geral e, em particular, as técnicas de codificação baseadas em ondas. Referem-se também os objectivos do trabalho e a organização da tese.

1.1 Enquadramento

Em quase todos os seres animais a visão e a audição são capacidades sensoriais privilegiadas na recolha de informação sobre o meio envolvente, indispensáveis à sua sobrevivência. Ao Homem são também importantes na percepção e conhecimento do mundo. Porém, o sistema visual humano é muito complexo. Pode-se afirmar que actualmente i) o sistema visual humano não é ainda completamente conhecido, ii) não existem critérios ou medidas que permitam avaliar a qualidade das imagens, segundo os critérios de qualidade de um observador-padrão, porque iii) não existe um modelo que permita estabelecer um padrão e definir o observador “típico”. Existe, pois, um elevado grau de subjectividade na avaliação da informação visual (imagens).

A recolha, a análise e a divulgação da informação, tem assumido ao longo dos tempos uma importância determinante na evolução das sociedades. Na sociedade actual, os avanços tecnológicos da última década, a par da adopção generalizada das tecnologias digitais, permitiram melhoramentos significativos dos sistemas de transmissão e armazenamento de dados, tornando possível a existência de novos serviços e o desenvolvimento de novos sistemas. Exemplos são os sistemas de videoconferência, de difusão de imagens e vídeo

(sequências de N imagens com referência temporal), via Internet, em tempo real, o armazenamento de vídeo em CD-ROM, e em DVD (*Digital Versatile Disk*), etc.

Por um lado, devido à grande quantidade de dados a processar, por outro lado, devido à ainda reduzida largura de banda dos canais de comunicação actualmente disponíveis, estes sistemas devem possuir elevada capacidade de processamento. A importância da elevada capacidade de processamento é reforçada nos sistemas para operação em tempo real, como é o caso dos sistemas de videoconferência, ou videodifusão. Por exemplo, uma sequência de vídeo com uma resolução 4CIF (576 linhas e 704 *pixels*¹ por linha) e 8 *bits* por *pixel*, quando se pretende obter um ritmo de 25 imagens/s corresponde a um débito binário de 81,1 *Mbits/s*. Pensando num CD-ROM, com uma capacidade de armazenamento de 650 *Mbyte*, apenas se poderia armazenar cerca de 8 segundos de vídeo.

A transmissão de imagem ou vídeo digital, sem compressão, implica ritmos binários elevados, apenas suportados por canais de transmissão de banda larga. Para transmissão em canais de banda estreita adoptam-se normalmente formatos de imagem reduzidos tais como CIF, QCIF e Sub-QCIF. Porém, como as imagens da sequência são fortemente correlacionadas, tanto no espaço como no tempo, é comum adoptarem-se técnicas de sub amostragem, espacial e temporal, e técnicas de compressão que exploram a correlação espacial e temporal de uma forma mais elaborada, mas que têm exigências computacionais mais elevadas. Estas técnicas de codificação admitem perdas, isto é, são processos não invertíveis, que exploram ainda a característica do sistema de visão humano. O desenvolvimento recente de técnicas e algoritmos eficientes para codificação² (compressão) de dados levou ao desenvolvimento de processadores de elevado desempenho para a sua implementação.

A correlação espacial, também designada por redundância espacial na perspectiva da compressão, é em geral explorada transformando a representação da imagem. A transformada de Fourier discreta (DFT), a transformada de Karhunen-Loève (KLT), a transformada de seno discreta (DST) e a transformada de co-seno discreta (DCT) são exemplos de transformações utilizadas.

Grande parte das normas de compressão de imagem, como sejam as normas JPEG [Bhask+96], e de compressão de vídeo, como sejam as normas MPEG [Gall_91,Bhask+96],

¹ *Pixel* resulta da contração, em Inglês, das palavras *Picture Element*, e significa ponto ou elemento de imagem.

² Embora em geral o termo codificação possa não estar associado a compressão, nesta tese utiliza-se o termo codificação como sinónimo de compressão, podendo implicar perda de informação.

H.261 e H.263 [ITU.263] calculam a DCT de blocos de imagem de dimensão fixa, normalmente 8×8 *pixels*. A codificação é do tipo com perdas, o que significa que o processo de codificação não é reversível, perdas que são fundamentalmente devidas à operação de quantificação.

Na codificação de vídeo utiliza-se também codificação do tipo diferencial, associada a técnicas de estimação/compensação de movimento, para reduzir a redundância temporal [Bhask+96]. Neste tipo de codificação, a partição da imagem em blocos cria fronteiras artificiais na imagem, que se reflectem nas componentes de alta frequência, e que produzem um efeito indesejável, designado por “efeito de bloco”.

A transformada de óndulas (*wavelets* em Inglês)³ (DWT) é outro tipo de transformada usada para explorar a redundância espacial. Apesar de não existir consenso quanto à designação Portuguesa para *wavelet*, a tendência parece ser a da adopção do termo óndula, havendo mesmo propostas nesse sentido, como é o caso em [Caer_97].

As primeiras construções matemáticas com características de óndula são conhecidas desde 1909, com o trabalho de A. Haar. No entanto foi em 1980 que os investigadores Grossmann e Morlet (físico e engenheiro, respectivamente), no domínio do processamento de sinais geofísicos, apresentaram a primeira definição de óndula e de transformada de óndula, como alternativa à utilização da análise pela transformada de Fourier local (STFT) [Graps_95].

As bases para a transformada de óndulas discreta (DWT) surgiram em 1976, através do trabalho quase simultâneo, mas independente, de Croisier, Esteban e Garland, (em processamento de voz) e Crochiere, Webber e Flanagan (em processamento de imagem), com decomposição de sinais discretos usando filtros espelhados em quadratura (QMF), a qual se designou por codificação em sub-bandas. Em 1983, Burt estabeleceu uma técnica semelhante à codificação em sub-bandas a qual designou, devido à sua organização hierárquica, por codificação piramidal.

Em 1985 registou-se um novo contributo na aplicação das óndulas ao processamento digital de sinais, com o trabalho conjunto de S. Mallat e Y. Meyer, ao estabelecer a relação entre os filtros QMF, a decomposição piramidal e as bases ortonormais, introduzindo o conceito de análise multiresolução [Malat_89,Gosw+99]. Mais tarde, Vetterli generalizou o conceito de análise multiresolução com aplicação das óndulas a sinais multi-dimensionais [Vette+92]. Entretanto I.

³ Óndulas são construções matemáticas que permitem efectuar a análise localizada de sinais, numa “janela” tempo-escala.

Daubechies propôs as ôndulas ortogonais de suporte compacto e, mais tarde, conjuntamente com Cohen e Feveau, as bases gerais das ôndulas de suporte compacto⁴, designadamente das ôndulas ortogonais e biortogonais, e a sua associação à análise multiresolução [Daub_92]. A principal contribuição da teoria das ôndulas, e da teoria da análise multiresolução é o facto de permitirem analisar os sinais a várias escalas. Em termos de compressão, consegue-se estabelecer um compromisso entre o valor aproximado da sequência de *pixels* e os detalhes (diferenças) associados, numa determinada escala.

Em 1994 foi proposto um método alternativo para obtenção de ôndulas ortogonais e biortogonais: o esquema progressivo (ou *lifting scheme*, em Inglês), que se descreve no capítulo 3. O esquema progressivo foi proposto por W. Sweldens, como alternativa à análise de Fourier na construção de ôndulas biortogonais [Sweld_95, Sweld_96]. Mais tarde, generalizou-se este método à construção de ôndulas de segunda geração, isto é, ôndulas cujo domínio não permite o uso da transformada de Fourier [Sweld_97]. Além da construção de ôndulas, o esquema progressivo também pode ser aplicado para cálculo eficiente da DWT e DWT^{-1} [Daub+98].

Nos últimos anos, o uso da transformada de ôndulas tem merecido um crescente interesse para a análise de sinal em geral e, em particular, para a compressão/codificação de imagem [Pearl+98] e de vídeo. Entre os vários esquemas de codificação com base na transformadas de ôndulas podem-se destacar a codificação embebida e hierárquica com a transformada de ôndulas (EZW) [Shapr_93], por fraccionamento hierárquico de conjuntos de coeficientes da transformada de ôndulas (SPHIT) [Said1+96], por trens de ôndulas, com minimização das perdas pelo esquema progressivo [Cald+98], e outros tais como [Anton+92, Cald+97]. A norma mais recente de compressão de imagem, JPEG 2000, é suportada na transformada de ôndulas [Marc+00, Skod+01, Usevt_01]. Por outro lado, também, no seio de grupos ad-hoc do MPEG tem renascido o interesse na realização de codificadores baseados na transformada de ôndulas [MPEG_02], nomeadamente codificadores 3D e cinema digital.

Em [Hilt+94] faz-se a comparação do desempenho de técnicas de codificação de imagem baseadas em ôndulas. Os resultados apresentados em [Hilt+94] mostram que a codificação EZW é aquela que apresenta melhores resultados e que os codificadores baseados nas ôndulas biortogonais apresentam melhores resultados do que os baseados nas ôndulas ortogonais [Anton+92]. O esquema de codificação EZW proposto por Shapiro [Shapr_93] tem suportado

⁴ Também geralmente designadas por ôndulas de Cohen-Daubechies-Feveau (CDF).

vários trabalhos de codificação de imagem com base em ôndulas, [Tham_95, Said2+96, Algaz+97]. Este esquema de codificação baseia-se na representação dos coeficientes da DWT, que se organizam em bandas (escalas) de frequência, segundo uma estrutura hierárquica.

Nas técnicas de codificação baseadas na transformada de ôndulas, em vez de se efectuar o fraccionamento das imagens em blocos, aplica-se a transformada à totalidade da imagem, evitando-se assim o aparecimento do “efeito de bloco”. Devido à sua organização hierárquica, com resolução adequada à análise nas diferentes escalas (codificação em sub-bandas), as técnicas de codificação baseadas em ôndulas revelam-se bastante promissoras para ritmos binários baixos, em particular devido à qualidade subjectiva das imagens que permitem obter [Villa+95,Tham1+98]. Por outro lado, as técnicas de codificação baseadas na transformada de ôndulas são robustas (do ponto de vista da transmissão e descodificação de erros) e facilitam a implementação de esquemas de transmissão progressiva da informação (controlo do ritmo binário).

A aplicação da transformada de ôndulas para codificar vídeo foi proposta a partir do início da década de 90. Os codificadores baseados nas transformadas de ôndulas inicialmente propostos têm uma estrutura semelhante à dos codificadores baseados na DCT, procedendo-se à substituição do cálculo da transformada local de co-seno pelo cálculo da transformada (global) de ôndulas [Ngan+96,Mart+97]. A decomposição da trama de vídeo a várias escalas efectuada pela DWT fornece, também, uma representação da estrutura global de movimento no sinal de vídeo a várias escalas. Dado que a actividade de movimento de uma trama é diferente mas correlacionada para as diferentes escalas, provou-se ser mais eficiente realizar a estimação de movimento no domínio da transformada [Mand+96].

Mais recentemente, propôs-se a extensão do esquema de codificação EZW a 3 dimensões, para a codificação de vídeo [Chen+96]. Este esquema de codificação, que permite obter factores de compressão elevados com uma reduzida degradação da qualidade das imagens, é o adoptado nesta tese para o desenvolvimento de um codificador de vídeo. Nesta dissertação utiliza-se o esquema progressivo para o cálculo rápido da transformada de ôndulas, e da transformada inversa [Chao+96, Cald+98, Chrys+00].

Na realização de codificadores e descodificadores, utilizam-se por vezes processadores de uso geral. Porém, os requisitos de processamento cada vez mais exigentes têm levado à adopção de processadores com arquitecturas especialmente vocacionadas para o processamento digital, podendo estes ser programáveis (DSPs), ou especialmente dedicados para uma classe particular

de algoritmos. Apesar dos sistemas com circuitos dedicados serem, regra geral, os mais eficientes, os sistemas com DSPs apresentam a vantagem de serem mais flexíveis, permitindo modificar o tipo de processamento por programação.

Os DSPs, em geral, adoptam a arquitectura de Harvard [Ifch+93, Pirsh_99, DSP_Hbk], dispondo de barramentos separados para dados e programa, dispondo de pelo menos um multiplicador e um acumulador realizado em *hardware*, possibilitando a realização de uma multiplicação e uma adição, em simultâneo, num único ciclo de relógio. As unidades aritméticas e lógicas (ALU) dos DSPs são especiais e na sua realização adoptam-se técnicas de multiplicação rápida [Sousa_98, Kung_88, Pirsh_99] e instruções especiais. Recentemente, o desempenho dos DSPs foi melhorado com o aumento dos recursos, permitindo a realização de operações de uma forma concorrente, através da adopção de arquitecturas VLIW⁵ [Pirsh_99]. Nestas arquitecturas, os diferentes campos da palavra de instrução contém as instruções individuais para cada uma das unidades funcionais, a ALU, o(s) multiplicador(es), o(s) registo(s) de deslocamento, etc. As várias unidades funcionais partilham um banco de registos, comum (ou mais), cujo acesso é suportado numa rede de barramentos cruzados. Apesar da complexidade e dos custos de *hardware*, estas redes têm a vantagem de não introduzir conflitos no acesso aos registos.

Os DSPs podem ser usados em processamento digital de imagem e vídeo, quer como processadores principais, quer como co-processadores, permitindo obter ganhos significativos nos tempos de processamento. Neste trabalho usa-se um DSP com arquitectura VLIW (TMS320C6201) que pode funcionar como co-processador, e permite obter ganhos significativos nos tempos de processamento, tendo-se implementado as rotinas de cálculo das transformadas de ôndulas, directas e inversas, no DSP TMS320C6201.

1.2 Principais Objectivos do Trabalho

O principal objectivo desta dissertação é o desenvolvimento de um codificador/descodificador (CoDec) de vídeo de ritmo binário variável, baseado na decomposição 3D por transformada de ôndulas. Trata-se de uma extensão da codificação EZW para imagem, a 3D, que foi

⁵ Do Inglês, *Very Large Instruction Word*. Arquitectura de processadores, com palavra de instrução de grande dimensão.

inicialmente proposta por [Tham+96, Tham1+98]. Outro dos objectivos deste trabalho é avaliar os requisitos computacionais e o tempo de processamento associados à codificação de vídeo com base na transformada de ôndulas. Este é um aspecto importante para o desenvolvimento de CoDecs de vídeo a funcionar em tempo real, e pouco tratado na literatura técnica. Para isso implementou-se o codificador num computador pessoal equipado com um processador Intel Pentium III a 550 MHz, com o sistema operativo Windows® 98, e com o ambiente de desenvolvimento Borland C++ Builder™ 5.0 para programação em “C/C++”.

Pela primeira vez foi utilizado o esquema progressivo para o cálculo da transformada de ôndulas, directa e inversa, em aplicações de codificação de vídeo. Outro dos objectivos do trabalho é a avaliação do desempenho dos diferentes tipos de ôndulas para a codificação de vídeo, tendo-se considerado neste trabalho as ôndulas ortogonais Daubechies 2 e biortogonais (4,4). Mais ainda, implementaram-se as rotinas de cálculo das transformadas (pelo esquema progressivo) num DSP com arquitectura VLIW, que pode funcionar como co-processador (TMS3206201 a 167 MHz). O ambiente de trabalho usado neste caso foi um computador pessoal equipado com as ferramentas de desenvolvimento para o DSP: um módulo de *hardware* (TMS320C6201 EVM) e as respectivas ferramentas de desenvolvimento e programação em “C/C++”. Para avaliar o desempenho do CoDec utilizaram-se seqüências de vídeo para teste, em formato QCIF, 8 *bits* por *pixel*, definidas no espaço YUV.

1.3 Organização da Tese

Esta tese está organizada em seis capítulos e três apêndices. No segundo capítulo são referidos os fundamentos associados à representação digital de imagens e de vídeo, e os formatos de vídeo normalizados. Apresentam-se também os aspectos mais relevantes das técnicas de codificação, com e sem perda de informação, em particular as técnicas de codificação com base na transformada de ôndulas e referem-se as medidas utilizadas para avaliar o desempenho de codificadores. No terceiro capítulo faz-se a introdução à teoria das ôndulas e à transformada de ôndulas, e a sua relação com os esquemas de decomposição hierárquica em sub-bandas. Faz-se também a apresentação do esquema progressivo (*lifting scheme*), como método eficiente para cálculo dos coeficientes da transformada de ôndulas. No quarto capítulo refere-se, em particular, a implementação das transformadas de ôndulas, directa e inversa, num processador de uso geral (Intel Pentium™ III a 550 MHz) e num processador digital de sinal

(TMS320C6201 da Texas Instruments), podendo, este último, funcionar como co-processador num sistema de codificação de vídeo. Neste capítulo apresentam-se as características do DSP TMS320C62x, indicando como se pode aproveitar as características da sua arquitectura, do tipo VLIW, para o cálculo da transformada de ondas pelo esquema progressivo. No quinto capítulo descreve-se o CoDec de vídeo baseado na transformada de ondas desenvolvido neste trabalho e faz-se a análise do seu desempenho, nomeadamente no que diz respeito ao factor de compressão e à qualidade das imagens das sequências codificadas. Para isso são usadas sequências de vídeo de teste com diferentes características. No sexto capítulo apresentam-se as conclusões finais desta tese e as perspectivas de trabalho futuro.

No apêndice A, apresentam-se os coeficientes e as propriedades dos filtros de ondas ortogonais de Daubechies e os coeficientes das ondas biortogonais. No apêndice B apresentam-se alguns exemplos simples, ilustrativos do cálculo dos coeficientes da DWT e da DWT^{-1} pelo esquema progressivo para sequências finitas, usando a onda ortogonal DB2 e o par de ondas biortogonais (4,4). No apêndice C apresentam-se as tabelas relativas à codificação de Huffman, usadas no CoDec desenvolvido neste trabalho.

Capítulo 2

CODIFICAÇÃO DE IMAGEM E VÍDEO

Neste capítulo referem-se os aspectos fundamentais associados à representação de imagens e vídeo digitais, e os formatos de vídeo normalizados. Referem-se ainda vários aspectos relativos às técnicas de codificação, com e sem perdas, aplicáveis à codificação de vídeo, em particular as técnicas de codificação com base na transformada de ondas.

2.1 Introdução

A compressão de imagem ou vídeo é um processo de compactação de informação, com o objectivo de a representar com o menor número possível de *bits*, mantendo determinados graus de qualidade e inteligibilidade, podendo conduzir ou não a perda de informação. Este processo visa a redução dos requisitos para armazenamento da informação, ou para a sua transmissão.

Em termos clássicos, uma imagem é um sinal bidimensional, obtido por reflexão do espectro da luz⁶ visível, permitindo identificar objectos e a sua localização no espaço. Actualmente, porém, este conceito generalizou-se à representação bidimensional de sinais obtidos noutros espectros de radiação, desde que lhe seja associada luz visível para que possa ser observada (por exemplo, Raios X, SONAR, Ecografia).

⁶ A luz é parte de um espectro mais vasto, contínuo, de radiação. O olho humano é sensível a uma gama muito estreita de comprimentos de onda – **espectro visível da luz** – $\lambda=350$ nm (**violeta**) a $\lambda=750$ nm (**vermelho**), aproximadamente. O espectro visível da luz compreende seis componentes monocromáticas fundamentais: violeta, azul, verde, amarelo, laranja, vermelho [Alon+99]. As restantes componentes de cor da luz visível correspondem a componentes espectrais múltiplas das fundamentais.

Se por um lado o sistema visual humano é fortemente dependente da luz, por outro lado, o espectro associado a uma imagem não é geralmente uniforme; varia de região para região, e pode apresentar componentes de frequência fundamentais e múltiplas. A informação contida numa imagem é “interpretada” pelo sistema visual humano como luz irradiada com diferentes intensidades e comprimentos de onda. A identificação de objectos numa imagem é possível i) devido à não uniformidade do espectro e ii) à intensidade relativa de cada componente. No entanto, a sensibilidade espectral da retina ao espectro de luz visível não é linear [Jlim_90, DSP_Hbk]. A intensidade luminosa detectada depende dos valores da intensidade real de uma forma logarítmica e, além disso, a sensibilidade do olho humano a um estímulo é adaptativa. A sensibilidade espectral global do olho humano varia com a frequência, atingindo o valor máximo na vizinhança do comprimento de onda $\lambda=555$ nm (verde-amarelo).

2.2 Representação Digital de Imagem e Vídeo

Para possibilitar a representação numérica de uma imagem, foram elaborados modelos matemáticos capazes de traduzir adequadamente o modo como o sistema visual detecta e interpreta a luz. Assim, foram associadas diferentes sensibilidades em três regiões diferentes do espectro visível – azul (*Blue*, ou B), verde (*Green*, ou G) e vermelho (*Red*, ou R) – correspondendo aos comprimentos de onda $\lambda_R = 700$ nm, $\lambda_G = 546,1$ nm e $\lambda_B = 435,8$ nm, respectivamente. Assim, a Comissão Internacional de Iluminação (CIE), definiu o sistema RGB de cores primárias, e uma grandeza que traduz a sensibilidade espectral da visão humana, designada por luminância –*Y*– tendo por unidade o lúmen [lm]. A luminância está relacionada com as componentes de cor RGB da seguinte forma [Jlim_90, Bhask+96]:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (2.1)$$

O sistema RGB de cores primárias é um sistema aditivo, isto é, qualquer cor (incluindo o cinzento) obtém-se por combinação das três componentes primárias (RGB) em proporções adequadas.

Uma sequência de imagens digitais, representando cenas diferentes, em instantes diferentes mas temporalmente próximos, transmitindo a ideia de movimento contínuo⁷, forma uma sequência de vídeo. A digitalização de sinais de vídeo envolve os processos de amostragem no tempo, no espaço e quantificação.

A amostragem espacial consiste na medição da intensidade luminosa num conjunto finito de pontos, numa área rectangular finita de dimensão $M \times N$. O modo de pesquisa (varrimento) mais comum é o modo *raster*, progressivo (não entrelaçado) ou entrelaçado, como se ilustra na figura 2.1. No modo não entrelaçado, as amostras são pesquisadas da esquerda para a direita, de cima para baixo, de forma progressiva, formando a imagem. No modo entrelaçado as amostras são divididas em dois subconjuntos: linhas pares e linhas ímpares. A pesquisa realiza-se da esquerda para a direita e de cima para baixo, inicialmente para todas as linhas ímpares e de seguida para todas as linhas pares. Cada subconjunto de amostras (pares e ímpares) forma um campo e os dois subconjuntos formam a imagem.

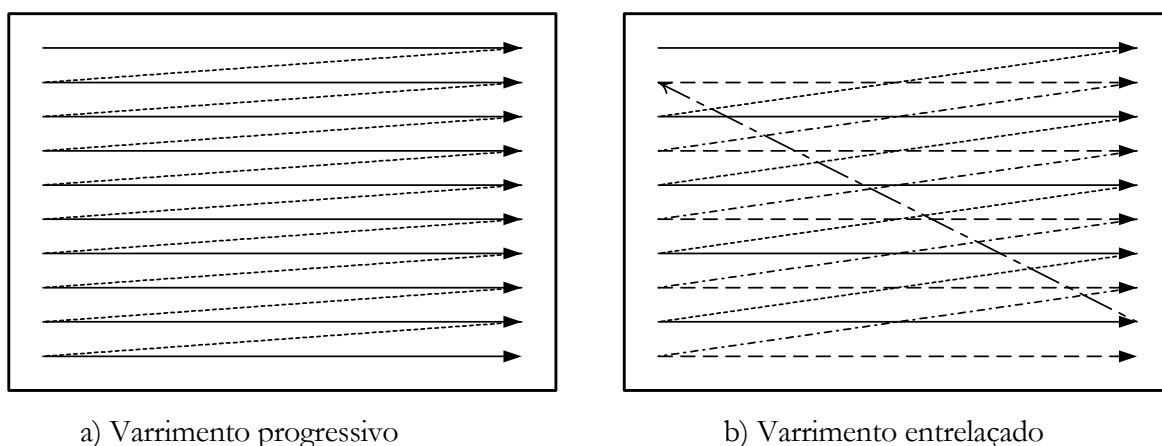


Figura 2.1: Varrimento no modo *raster*, para amostragem espacial em sinais de vídeo.

Como o sistema visual humano apresenta uma resposta relativamente lenta às variações das imagens no tempo, é possível transmitir uma ideia razoável de movimento com quinze ou vinte imagens por segundo. A qualidade é tanto melhor quanto maior for o número de imagens por segundo. Em televisão é comum usar-se o modo entrelaçado com taxas de amostragem no tempo de 25 ou 30 imagens por segundo e a frequência de campos de 50 ou 60 tramas por segundo.

⁷ Apesar da informação ser discreta (em representação) deve transmitir-se a ideia de movimento contínuo.

Finalmente faz-se a quantificação das amostras, que normalmente é uniforme, com um número de níveis que é uma potência de 2 ($Q = 2^n$, n - bits), permitindo representar cada valor por um número fixo de bits. Após os processos de amostragem espacial e temporal e de quantificação, têm-se $M \times N$ pontos ou elementos de imagem, designados por *pixels*, representados por um número fixo de *bits*.

Normalmente, no sistema RGB, uma imagem é representada por três matrizes de dimensão $M \times N$, que representam os valores de intensidade das componentes cromáticas dos *pixels*. Como o sistema visual humano é mais sensível à intensidade luminosa do que à cor, é comum representar os sinais de vídeo digital através de um sinal de luminância (Y) e dois sinais de crominância (C_b e C_r). Habitualmente usam-se os seguintes formatos de vídeo com sub amostragem, $Y C_b C_r$, respectivamente:

4:2:2 – por cada linha, a resolução das crominâncias é metade da resolução da luminância;

4:1:1 – por cada linha, a resolução das crominâncias é um quarto da resolução da luminância;

4:2:0 – a resolução das crominâncias é metade da resolução da luminância tanto na direcção horizontal como na vertical.

O formato de vídeo sem sub amostragem é 4:4:4. Por exemplo, a norma H.263 usa o formato 4:2:0.

As componentes do sinal de vídeo, por sua vez, estão relacionadas com o sistema de cores primárias da seguinte forma:

$$Y = 0,299 \cdot (R - G) + G + 0,114 \cdot (B - G) \quad (2.2 \text{ a})$$

$$C_b = 0,564 \cdot (B - Y) \quad (2.2 \text{ b})$$

$$C_r = 0,713 \cdot (R - Y) \quad (2.2 \text{ c})$$

Ou, na forma matricial,

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,500 \\ 0,500 & -0,419 & -0,081 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.3)$$

2.2.1 Formatos de Vídeo Normalizados

A norma CCIR-601 da ITU-T permite diferentes formatos de vídeo: o sistema NTSC⁸ (Estados Unidos da América do Norte e Japão), com resolução 720×480 ($Y: 720 \times 480$, $C_r C_b: 360 \times 480$), 30 imagens por segundo; e o formato PAL⁹ (Europa), com resolução 720×576 ($Y: 720 \times 576$, $C_r C_b: 360 \times 576$) e 25 imagens por segundo. No sistema PAL usa-se o formato de sub amostragem 4:2:2 (em cada linha, a resolução das crominâncias é metade da resolução da luminância). Contudo, em vez do sistema de coordenadas $Y C_b C_r$, usa-se, também, YUV, igualmente relacionado com o sistema RGB do seguinte modo:

$$Y = 0,299 \cdot (R - G) + G + 0,114 \cdot (B - G) \quad (2.4 \text{ a})$$

$$U = 0,493 \cdot (B - Y) \quad (2.4 \text{ b})$$

$$V = 0,877 \cdot (R - Y) \quad (2.4 \text{ c})$$

Ou na forma matricial,

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,148 & -0,289 & 0,437 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

Porém, como a norma CCIR-601 especifica valores diferentes para a amostragem temporal e resolução espacial nos sistemas PAL e NTSC, foi proposto um formato de vídeo comum intermédio (CIF). O formato CIF utiliza imagens com resolução espacial 352×288 *pixels*, e 30 imagens por segundo (no sistema PAL). As normas ITU-T H.261 e H.263, suportam formatos de imagem organizados em múltiplos ou submúltiplos da resolução padrão¹⁰ de 720×576 (PAL), estabelecida na norma CCIR-601. O formato de sub amostragem utilizado é 4:2:0, isto é, a resolução das crominâncias é metade da resolução da luminância, tanto na horizontal como na vertical. Na tabela 2.1 apresentam-se os valores da resolução espacial suportados pelas normas H.261 e H.263.

⁸ Do Inglês, *National Television Systems Committee*. Comissão criada nos EUA, em 1950, para normalização dos sistemas TV.

⁹ Do Inglês, *Phase Alternate Line*.

¹⁰ As normas H.261 e H.263, fraccionam a imagem em Macroblocos de 16×16 *pixels*. Como 360 não é múltiplo de 16, usa-se o múltiplo mais próximo, isto é, 352×288 *pixels*.

	$Y(\text{linhas} \times \text{colunas})$	$C_b, C_r(\text{linhas} \times \text{colunas})$	H.261	H.263
Sub-QCIF	96 × 128	48 × 64	<input type="checkbox"/>	<input checked="" type="checkbox"/>
QCIF	144 × 176	72 × 88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CIF	288 × 352	144 × 176	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4CIF	576 × 704	288 × 352	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16CIF	1152 × 1408	576 × 704	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Tabela 2.1: Resolução espacial para os formatos de vídeo digital normalizados, de acordo com as normas H.261 e H.263.

Na avaliação do desempenho do CoDec realizado neste trabalho (capítulo 5), usaram-se sequências de vídeo em formato QCIF, armazenadas em ficheiros em formato binário, com extensão **.YUV**. A representação de cada imagem corresponde a três matrizes (Y, U e V), correspondentes a cada uma das componentes (uma para a luminância e as restantes duas para as crominâncias), 8 *bits* por *pixel*, a que correspondem valores inteiros entre 0 e 255.

2.3 Técnicas de Codificação sem Perda de Informação

As técnicas de codificação que não implicam perda de informação, isto é, que são perfeitamente reversíveis, exploram essencialmente a redundância do código associada à representação dos *pixels* ou dos coeficientes das transformadas. Na representação digital de imagem, por exemplo, o valor dos *pixels* tem normalmente uma dimensão fixa em *bits*. Uma forma simples de representação consiste em codificar os valores de cada *pixel* com base num esquema de codificação com dimensão fixa, de complexidade $O(\log_2 N)$, em que N representa o número de símbolos a codificar. Porém, apesar de simples, este método não é eficiente.

Os métodos que exploram a redundância do código baseiam-se na probabilidade de ocorrência do valor, gerando códigos ou símbolos a partir do histograma da distribuição de probabilidade (codificação do tipo entrópico), associado a códigos de comprimento variável (VLC). Os valores com maior probabilidade de ocorrência são representados com menor número de símbolos e vice-versa. A entropia de uma fonte, $H(S)$, (segundo Shannon) é definida por:

$$H(S) = \eta = \sum_i p_i \cdot \log_2 \frac{1}{p_i} \quad (2.6)$$

em que p_i representa a probabilidade de ocorrência do símbolo s_i , na fonte S .

A dimensão média dos símbolos do código é definida por:

$$l_{av} = \sum_i l_i \cdot p_i \quad (2.7)$$

onde l_i é a dimensão em *bits* do código correspondente ao símbolo s_i , com probabilidade p_i .

A eficiência da codificação (em percentagem) é determinada pelo número médio de *bits* necessários para codificar a informação contida numa fonte, e a entropia associada a essa fonte, sendo definida através do cociente entre as equações 2.6 e 2.7.

2.3.1 Codificação de Huffman

O código de Huffman é um dos códigos VLC mais simples e mais frequentemente usados [Bhask+96, Jlim_90]. O comprimento do código, em *bits*, varia inversamente com a probabilidade de ocorrência dos símbolos, isto é, os símbolos com maior probabilidade de ocorrência têm dimensão menor, e vice-versa. Assim, quanto menor for o número de símbolos a codificar, menor será a entropia. Normalmente, a codificação de Huffman é usada em fontes com uma distribuição estatística de símbolos (valores) significativa, portanto, onde se torna efectivo explorar a redundância associada ao código.

O processo de construção do código de Huffman para uma sequência de símbolos, consiste na determinação, *a priori*, da probabilidade de ocorrência de cada símbolo, e na atribuição de um código binário com comprimento variável e decrescente com a probabilidade de ocorrência. O novo código garante a inexistência de ambiguidades na descodificação. Por razões práticas de implementação, as probabilidades de ocorrência dos símbolos são normalmente expressas por múltiplos de fracções de potências de 2.

A construção do código de Huffman envolve os seguintes passos:

1. Ordenação da tabela de símbolos por ordem decrescente de probabilidades;
2. Agrupamento dos dois símbolos com menor probabilidade (dois últimos símbolos), dando origem a um novo símbolo, cuja probabilidade de ocorrência corresponde à

soma das probabilidades iniciais; ao último símbolo da tabela atribui-se o *bit* 1 e ao penúltimo, o *bit* 0;

3. O passo anterior repete-se até que o conjunto final contenha apenas um símbolo com probabilidade 1,0.

O procedimento descrito pode ser visto como um processo recursivo de construção de uma árvore binária. Os códigos assim construídos ficam organizados por ordem crescente de significado: primeiro os menos significativos e depois os mais significativos. Para que não exista ambiguidade na descodificação, torna-se necessário inverter esta ordem.

Considere-se, por exemplo, uma sequência com 5 símbolos diferentes com as seguintes probabilidades: $p(a_1) = \frac{5}{8}$, $p(a_2) = \frac{3}{32}$, $p(a_3) = \frac{3}{32}$, $p(a_4) = \frac{2}{32}$, $p(a_5) = \frac{1}{8}$. A codificação destes símbolos é apresentada na figura 2.2.

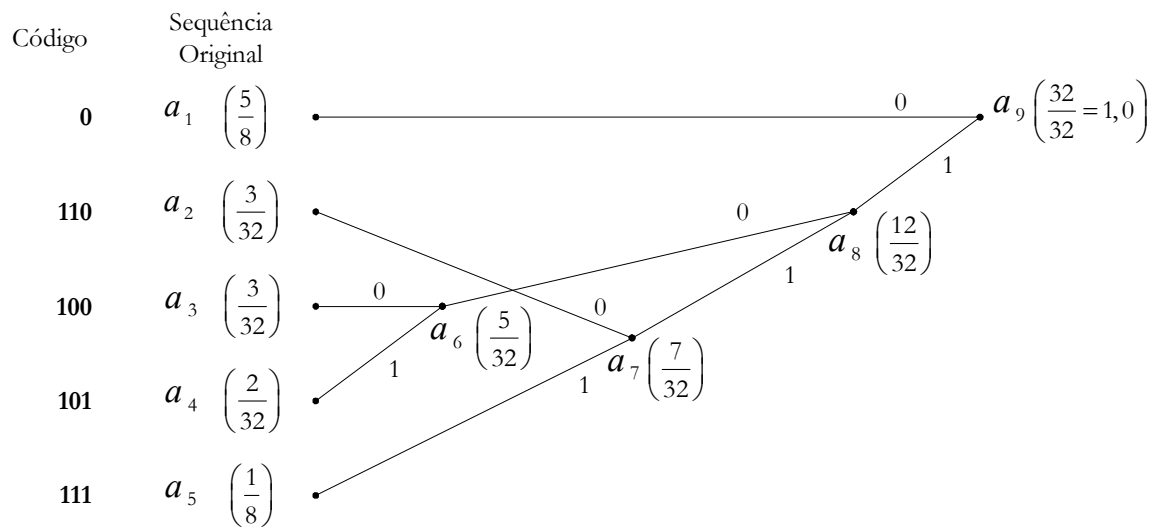


Figura 2.2: Exemplo de um esquema de codificação de Huffman.

A representação de uma sequência de oito símbolos, $a_4 a_1 a_1 a_1 a_3 a_5 a_3 a_2$, através do código de comprimento fixo, requer 3 *bits* por símbolo, isto é, 24 *bits*. Com o código de Huffman apenas são necessários 18 *bits* ($3+1+1+1+3+3+3+3$), sendo a dimensão média do código e a entropia de:

Código de Huffman

$$\sum_i l_i \cdot p_i = \left(3 \cdot \frac{3}{32} + 1 \cdot \frac{5}{8} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{3}{32} + 3 \cdot \frac{2}{32} \right) = \frac{25}{16} = 1,75 \text{ bits/símbolo};$$

Entropia

$$\sum_i p_i \cdot \log_2 \frac{1}{p_i} = \left(\frac{3}{32} \log_2 \frac{32}{3} + \frac{5}{8} \log_2 \frac{8}{5} + \frac{3}{32} \log_2 \frac{32}{3} + \frac{2}{32} \log_2 16 + \frac{1}{8} \log_2 8 \right) \approx 1,69 \text{ bits/símbolo}$$

2.3.2 Codificação Aritmética

Na codificação de Huffman os símbolos têm um número inteiro de *bits*, isto é, o limite inferior de compressão no código de Huffman é de um *bit* por símbolo. Em esquemas de codificação de sequências com uma grande sucessão de símbolos iguais e alternância esporádica de símbolos diferentes a eficiência não é muito grande. Através da combinação de vários símbolos num só é possível obter factores de compressão mais elevados. Porém, verifica-se também um aumento na complexidade de construção do código.

A codificação aritmética é um esquema de codificação sem perdas, alternativo à codificação de Huffman, em que os símbolos são combinados. A codificação aritmética, porém, não associa um número inteiro de *bits* a cada símbolo. Isto permite obter factores de compressão superiores (ou pelos menos iguais) aos que se obtém com a codificação de Huffman e, tal como nesta, também é necessário conhecer a probabilidade de ocorrência de cada símbolo. O processo de codificação inicia-se com a criação de uma tabela inicial de probabilidade de ocorrência de símbolos, com valores reais entre 0 e 1, e na subdivisão desse intervalo em sub intervalos, tantos quantos os símbolos a codificar. Por exemplo, nas condições do exemplo da figura 2.2 obtém-se a seguinte subdivisão de intervalos:

s_i	p_i	Sub intervalo
a_1	$5/8$	$[0; 20/32 [$
a_2	$3/32$	$[20/32; 23/32 [$
a_3	$3/32$	$[23/32; 26/32 [$
a_4	$2/32$	$[26/32; 28/32 [$
a_5	$1/8$	$[28/32; 1 [$

Tabela 2.2: Exemplo da organização de sub intervalos na codificação aritmética.

O processo de codificação envolve os seguintes passos:

1. No início do processo de codificação, assume-se que os símbolos da sequência estão contidos num dos sub intervalos do alfabeto proposto. O intervalo actual é, portanto, definido entre 0 e 1;
2. Com a recepção do primeiro símbolo é seleccionado o sub intervalo correspondente, que é assumido como “intervalo actual”. O intervalo anterior é definido pelos seus limites superior e inferior, isto é, Ant_{Sup} e Ant_{Inf} , respectivamente. A gama de valores desse intervalo é definida por $Interv = Ant_{Sup} - Ant_{Inf}$. Os limites inferior e superior do intervalo actual são definidos por: $Act_{Inf} = Ant_{Inf} + Interv \times SubInterv_{Inf}$ e $Act_{Sup} = Ant_{Sup} + Interv \times SubInterv_{Sup}$.
3. O processo repete-se até que seja encontrado o último símbolo, determinando-se assim o intervalo final;

Por exemplo, para codificação da sequência $a_4 a_1 a_1 a_5 a_3 a_2$ (usada anteriormente como exemplo) em que o primeiro símbolo é a_4 , seguindo os passos 1 a 3 da codificação aritmética obtém-se no esquema de codificação da tabela 2.3 conforme se explica a seguir.

Com a recepção do primeiro símbolo é seleccionado o sub intervalo $[26/32; 28/32[$ considerando o intervalo anterior definido entre 0 e 1 ($Interv = 1$). O intervalo actual é dado por $Act_{Inf} = 0 + 1 \times \frac{26}{32} \approx 0,8125$ e $Act_{Sup} = 0 + 1 \times \frac{28}{32} \approx 0,8750$. Em seguida é recebido o segundo símbolo, a_1 , a que corresponde o sub intervalo $[0; 20/32[$. Os limites do novo intervalo passam a ser definidos por $Act_{Inf} = 0,8125 + (0,8750 - 0,8125) \times 0 = 0,8125$ e $Act_{Sup} = 0,8125 + (0,8750 - 0,8125) \times \frac{20}{32} \approx 0,851562$. Ao terceiro símbolo (a_1) corresponde o mesmo sub intervalo, isto é, $[0; 20/32[$, e o novo intervalo passa a ser definido por $Act_{Inf} = 0,8125$ e $Act_{Sup} = 0,8125 + (0,851562 - 0,8125) \times \frac{20}{32} \approx 0,836914$. Por cada novo símbolo o processo desenvolve-se do mesmo modo, terminando com o símbolo a_2 a que correspondem valores no intervalo $[0,827704; 0,827706[$. Não há necessidade de transmitir os valores limite do intervalo; é mais adequado transmitir um valor contido no intervalo.

Símbolo	Sub intervalo	Intervalo (anterior)	Intervalo (actual)
a_4	$[26/32; 28/32 [$	$[0; 1[$	$[0,812500; 0,875000 [$
a_1	$[0; 20/32 [$	$[0,812500; 0,875000 [$	$[0,812500; 0,851562 [$
a_1	$[0; 20/32 [$	$[0,812500; 0,851562 [$	$[0,812500; 0,836914 [$
a_1	$[0; 20/32 [$	$[0,812500; 0,836914 [$	$[0,812500; 0,827758 [$
a_5	$[28/32 ; 1 [$	$[0,812500; 0,827758 [$	$[0,825850; 0,827758 [$
a_5	$[28/32 ; 1 [$	$[0,825850; 0,827758 [$	$[0,827519; 0,827758 [$
a_3	$[23/32; 26/32 [$	$[0,827519; 0,827758 [$	$[0,827690; 0,827713 [$
a_2	$[20/32; 23/32 [$	$[0,827690; 0,827713 [$	$[0,827704; 0,827706 [$

Tabela 2.3: Exemplo de esquema de codificação aritmética.

Para a sequência referida podia-se usar, por exemplo, o valor 0,827705, que é representado por $2^{-1} + 2^{-2} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-14} + 2^{-17}$, sendo portanto necessários 17 *bits* para a sua representação. Como sequências diferentes originam intervalos diferentes não existe ambiguidade associada ao código. Comparando-se a dimensão da palavra de código para a sequência $a_4 a_1 a_1 a_3 a_5 a_3 a_2$ ela é de 17 *bits* com um codificador aritmético, 18 *bits* para o codificador de Huffman, e 24 *bits* para um codificador de dimensão fixa.

2.4 Técnicas de Codificação com Perda de Informação

A transmissão de vídeo digital, sem compressão, normalmente implica ritmos binários elevados. Por exemplo, para transmitir a informação de uma fonte de vídeo policromática, em formato CIF (288×352), 8 *bits* por *pixel*, 25 imagens por segundo, sem sub amostragem espacial, é necessário um débito binário de 60,825 *Mbits/s*. Como as imagens da sequência de vídeo são fortemente correlacionadas, tanto no espaço como no tempo, é comum adoptarem-se técnicas de sub amostragem. Com sub amostragem espacial no formato 4:2:0 e sub amostragem temporal de 4, o ritmo binário diminuiria de 60,825 para 7,6 *Mbits/s*. Para

transmitir essa informação, por exemplo, através de um canal com largura de banda $n \times 64 \text{ kbits/s}$ (linha RDIS – Rede Digital com Integração de Serviços), torna-se necessário obter um factor de compressão superior a $(1/n) \times 118$, o que se obtém apenas se se recorrer a técnicas de codificação com perda de informação.

A estrutura típica de um codificador de imagem ou vídeo inclui blocos de transformação, de quantificação e de codificação. Como o codificador produz uma sequência de *bits* que se pretende transmitir, é comum usar-se codificadores do tipo entrópico, de Huffman ou aritméticos, de modo a explorar a redundância do código e aumentar o factor de compressão.

2.4.1 Quantificação Escalar e Vectorial

Os métodos de codificação que usam quantificação têm implicitamente perda de informação. A quantificação usa-se normalmente após aplicação de transformadas às imagens, dado que transformam o conjunto discreto de valores dos *pixels* num conjunto contínuo de valores reais, ou complexos.

A quantificação consiste essencialmente na limitação dos valores possíveis para cada coeficiente, estabelecendo uma correspondência entre os valores contínuos e discretos dos coeficientes. Deste modo, a quantificação resulta num processo de discretização, linear ou não linear, nos limites definidos pelos valores dos coeficientes da transformada, para um conjunto finito de sub intervalos, ou níveis de discretização pretendidos. A quantificação faz-se por identificação do sub intervalo mais próximo, dentro de determinados níveis de decisão, mediante o cálculo do erro de quantificação associado, substituindo o valor de cada coeficiente pelo índice ou código correspondente.

Seja f um escalar relativo ao valor de um coeficiente resultante da transformada. O seu valor quantificado, \hat{f} , para um número finito de níveis, L , pode ser expresso pela equação:

$$\hat{f} = Q(f) = r_i \quad d_{i-1} < f \leq d_i \quad (2.8)$$

em que Q designa a operação de quantificação entre os níveis de decisão d_i e d_{i-1} , e r_i o nível de reconstrução. O método de quantificação mais simples consiste na aplicação de um quantificador uniforme, em que os níveis de decisão e de reconstrução se encontram uniformemente espaçados.

O valor quantificado pode também ser expresso em função do erro de quantificação

$$\hat{f} = Q(f) = f + e_Q \quad (2.9)$$

em que, $e_Q = \hat{f} - f$

O problema central associado à realização de um quantificador é a escolha dos intervalos de quantificação, que minimizam a distorção devido aos erros de quantificação. Normalmente usa-se o critério da minimização do erro quadrático médio (MSE). Para uma variável aleatória f , com uma função de densidade de probabilidade $p_f(f_0)$, o cálculo do MSE é dado pela equação:

$$D = \int_{-\infty}^{\infty} p_f(f_0) \cdot (\hat{f} - f_0)^2 df_0 \quad (2.10)$$

Como normalmente a variável f é limitada em L sub intervalos (níveis) d_i , com $i = 0, \dots, L-1$, tal que $d_{i-1} < f \leq d_i$, e o quantificador tiver um nível de reconstrução $r_i = Q(f) = \hat{f}_i$, a equação 2.10 fica,

$$D = \sum_{i=1}^L \int_{d_{i-1}}^{d_i} p_f(f_0) \cdot (\hat{f}_i - f_0)^2 df_0 \quad (2.11)$$

A quantificação escalar restringe o valor que cada coeficiente pode tomar face à gama de valores que poderia ter originalmente, sendo cada valor quantificado de forma independente.

A quantificação vectorial, ou quantificação em bloco é um método alternativo à quantificação escalar, que consiste em segmentar a imagem, agrupando os coeficientes em blocos e quantificando os vários escalares desse bloco de uma só vez.

Extraindo um vector $\mathbf{v}_k = [v_1, v_2, \dots, v_N]$ de dimensão N de uma imagem, correspondendo a um bloco ou uma célula, C_k , e estabelecendo uma correspondência entre este vector e um vector $\mathbf{r}_j = [r_1, r_2, \dots, r_N]$, ($i = 1, 2, \dots, L$), igualmente com dimensão N , procede-se a uma quantificação vectorial. Assim, tal como na codificação escalar, cada vector de segmentação de imagem é quantificado por um quantificador vectorial, de forma que,

$$\hat{\mathbf{v}}_k = VQ(\mathbf{v}_k) = \mathbf{r}_j, \quad 1 \leq j \leq L \quad (2.12)$$

As medidas de distorção, ou o erro de quantificação, resultam das diferenças entre os valores iniciais e os valores quantificados,

$$d(\mathbf{v}_k, \hat{\mathbf{v}}_k) = e_{vQ} = \hat{\mathbf{v}}_k - \mathbf{v}_k \quad (2.13)$$

A distorção média global é o erro quadrático médio generalizado [Jlim_90], e define-se através da equação:

$$D = E[d(\mathbf{v}_k, \hat{\mathbf{v}}_k)] = E[\hat{e}_{vQ} e_{vQ}^T] = E[(\hat{\mathbf{f}} - \mathbf{f})^T (\hat{\mathbf{f}} - \mathbf{f})] \quad (2.14)$$

As técnicas de codificação com base em transformadas, normalmente recorrem a um dos métodos de quantificação referidos.

2.4.2 Codificação com Base em Transformadas

As técnicas de codificação que usam transformadas, excepto a DWIT, decompõem normalmente as imagens em blocos de dimensões mais reduzidas (8×8 ou 16×16), aos quais se aplica a transformada. Cada bloco sofre uma transformação de representação para o domínio da transformada, sendo posteriormente quantificada e codificada.

Para que exista uma boa concentração de energia, usam-se transformadas tais como a DFT, a KLT e a DCT. Estas transformadas são lineares e aplicadas sobre bases ortogonais, realizando de forma efectiva a decorrelação dos *pixels*, concentrando energia e permitindo representar as imagens num número de coeficientes bastante mais reduzido que a representação original.

A transformada KLT decorrelaciona bem os coeficientes e minimiza a degradação (distorção) das imagens. Porém, a sua implementação prática é difícil e computacionalmente exigente, sendo, por isso, normalmente substituída pela DCT que apresenta resultados aproximados e permite a realização de algoritmos eficientes.

Para factores de compressão elevados verifica-se a existência de efeito de bloco [Bhask+96], devido ao processo de codificação independente de blocos adjacentes. Apesar desta limitação, a DCT e a DCT^{-1} têm sido bastante usadas na codificação de imagens e de vídeo, quer no suporte da norma JPEG (imagem), quer nas normas H.261, H.263 e MPEG-1/2 (vídeo).

Dada uma sequência bidimensional $x_{m,n}$ com $N \times N$ elementos ($0 \leq m, n \leq N-1$), os coeficientes da sua transformada DCT-2D $X_{u,v}$ são definidos pela equação:

$$X_{u,v} = \frac{2}{N} C(u) \cdot C(v) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \cdot \cos\left(\frac{(2m+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2n+1)v\pi}{2N}\right) \quad (2.15)$$

em que,

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & x = 0 \\ 1 & x \neq 0 \end{cases}$$

Nas normas de codificação anteriormente referidas e que usam a DCT, normalmente utiliza-se $N = 8$.

A transformada DCT-2D inversa é definida pela equação:

$$x_{m,n} = \frac{2}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X_{u,v} \cdot C(u) \cdot C(v) \cdot \cos\left(\frac{(2m+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2n+1)v\pi}{2N}\right) \quad (2.16)$$

A DCT e a DCT^{-1} formam um par de transformadas complementares, para a decorrelação de sinais. A sua simples aplicação não resulta, portanto, em qualquer compressão; esta apenas se obtém por quantificação dos coeficientes da transformada. Como o sistema visual humano é mais sensível às frequências espaciais baixas do que às frequências altas, os coeficientes da transformada relativos às frequências altas podem ser quantificados de forma mais “grosseira” do que os coeficientes relativos às frequências mais baixas.

Os coeficientes relativos às altas-frequências são normalmente nulos, ou pelo menos negligenciáveis face a um dado nível de decisão. A codificação dos coeficientes faz-se segundo um esquema de varrimento em ziguezague, como se ilustra na figura 2.4. Segundo este esquema, a ordem do varrimento reflecte a importância dos coeficientes, codificando os coeficientes de frequência espacial mais baixa antes dos coeficientes de frequência mais elevada. O coeficiente de mais baixa frequência (DC) $F(0,0)$ é codificado separadamente dos restantes coeficientes, não sendo incluído no varrimento em ziguezague.

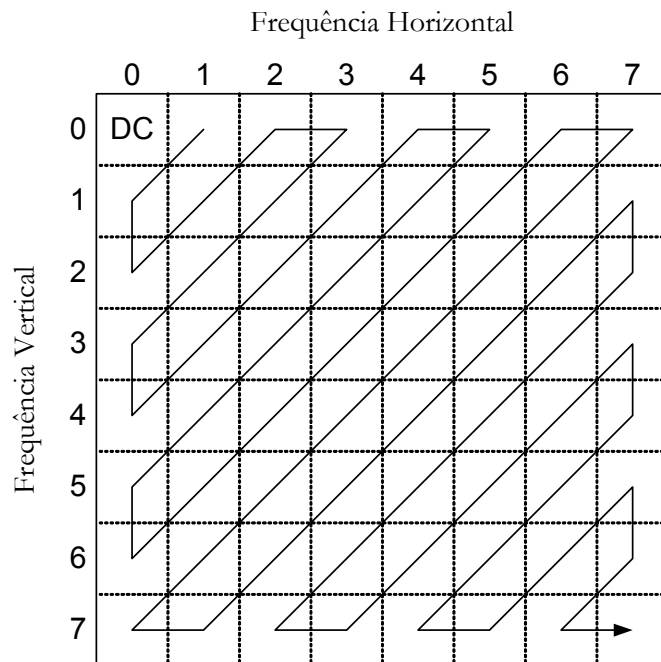


Figura 2.3: Varrimento em zigzag para codificação dos coeficientes quantificados da DCT-2D.

2.4.3 Codificação Hierárquica com Base na Transformada de Ôndulas

Ao contrário da DCT, nos esquemas de codificação com base na DWT, a imagem não é fraccionada em blocos, aplicando-se a transformada à totalidade da imagem. A DWT realiza a decorrelação espacial entre os *pixels* de uma imagem, concentrando a energia nas regiões referentes às frequências espaciais mais baixas, fazendo uma transposição da informação contida em toda a imagem para uma região com dimensões mais reduzidas, normalmente

$\left(\frac{M}{2^p}, \frac{N}{2^p}\right)$, em que p representa o número de níveis (escalas) da transformada.

A aplicação da DWT a uma imagem realiza uma decomposição piramidal, com uma organização hierárquica em árvore. Os vários níveis, ou escalas, resultantes da aplicação da DWT, correspondem a aproximações progressivas à imagem original, com maior detalhe.

A estrutura piramidal de decomposição da DWT, associada ao conceito de análise multiresolução proposto por Y. Meyer e S.Mallat [Malat_89] (que serão referidos em detalhe no capítulo seguinte) permitem a realização de esquemas efectivos de codificação de imagem de vídeo, organizando os coeficientes da transformada numa estrutura hierárquica.

A análise multiresolução consiste na projecção ortogonal de uma função num conjunto de sub-espacos de $L^2(\mathbb{R})$, fechados, organizados num esquema de aproximação sucessiva [Malat_89, Daub_92, Gosw+99]. Estabelecendo a relação entre a análise multiresolução e o esquema de decomposição em sub-bandas, I. Daubechies apresentou um esquema de análise e síntese, através de filtros FIR ortogonais (igualmente referidos no capítulo seguinte) associados às respectivas ôndulas ortogonais.

As técnicas de codificação baseadas na transformada de ôndulas envolvem, em geral três fases: a aplicação da transformada (DWT), a quantificação dos coeficientes da transformada e a codificação dos coeficientes quantificados. Têm sido propostos vários esquemas de codificação, com quantificadores de natureza escalar e natureza vectorial, usando filtros QMF e filtros de diferentes tipos de ôndulas.

Dos esquemas de codificação baseados na transformada de ôndulas importa destacar, pela sua popularidade, a codificação EZW (para imagens) proposta por J. Shapiro [Shapr_93]. O esquema EZW baseia-se na existência de uma estrutura hierárquica em árvore, organizada em bandas de frequência, resultante da aplicação da DWT, e que explora as relações hierárquicas de dependência segundo as escalas (níveis), e segundo dadas orientações. O esquema de codificação (e descodificação) exploram a elevada probabilidade de que os coeficientes mais significativos se situam nas regiões correspondentes às frequências mais baixas (níveis mais altos), enquanto os menos significativos se situam nas regiões de frequência mais elevada (níveis mais baixos). Este esquema permite atingir factores de compressão aceitáveis sem contudo degradar significativamente a qualidade das imagens. A codificação EZW será referida com maior detalhe no capítulo 5, na descrição do CoDec de vídeo, em particular no que respeita à sua extensão a sinais de vídeo (3D) [Chen+96].

2.4.4 Estimação e Compensação de Movimento

As sucessivas imagens de uma sequência de vídeo encontram-se, geralmente, fortemente correlacionadas, em particular em cenas com pouco movimento. As técnicas de decorrelação espacial, anteriormente indicadas, referem-se apenas a uma imagem e, portanto, não exploram a correlação existente entre imagens.

Uma técnica bastante simples de explorar a redundância no tempo consiste em obter a representação de uma imagem no instante t pelas diferenças em relação à imagem anterior, no

instante $t-1$, codificando-a posteriormente. O processo de cálculo das diferenças entre imagens, estabelecendo uma correspondência entre elas, pode ser visto como uma técnica de codificação predictiva no tempo. Porém, se a sequência de imagens apresentar uma quantidade razoável de movimento esta técnica revela-se pouco eficiente. Além disso, é difícil obter uma qualidade razoável na codificação de sequências de vídeo, em especial para ritmos binários baixos ($< 300 \text{ kbits} / \text{s}$), sem o recurso a técnicas de estimação e compensação de movimento [Bhask+96].

Assim, nos esquemas de codificação com base nas normas H.261 e H.263, que usam fraccionamento da imagem em blocos, a redução da redundância no tempo faz-se pelo cálculo das diferenças entre blocos de imagens consecutivas, cuja localização resulta da estimação de movimento da imagem actual face à imagem anterior. A compensação de movimento pode ser realizada pela translação dos blocos que constituem uma imagem, segundo uma direcção e um sentido definidos por um vector de movimento resultante do processo de estimação.

As normas H.261 e H.263, e também as normas MPEG-1/2, utilizam métodos de estimação de movimento por emparelhamento de blocos (macroblocos de $16 \times 16 \text{ pixels}$). Existem várias técnicas para estimação de movimento por emparelhamento de blocos, de entre os quais o método de pesquisa exaustiva por permitir obter resultados óptimos. Porém, apresenta um peso computacional muito elevado, sendo apenas possível a sua implementação em processadores dedicados.

Com o objectivo de reduzir o peso computacional, foram desenvolvidos outros métodos de pesquisa que conduzem a resultados sub óptimos, como o método de pesquisa em 3 passos (TSS) proposto por Koga *et. al.* em 1981. Além deste, têm sido propostos outros métodos de estimação por emparelhamento de blocos, tais como: melhoramentos ao método TSS [ReLi+94, JiLu+97], o método de pesquisa em 4 passos [LaiPo+96], métodos com decimação e estimação de movimento em sub-blocos [Bliu+93] e o método de pesquisa centrado numa área em forma de losango [Tham2+98].

Os métodos de pesquisa consistem em fazer deslizar sobre uma região de pesquisa de dimensão $(a + \Delta_a) \times (b + \Delta_b)$, na imagem do instante $t-1$, um bloco de dimensão $a \times b$, referente à imagem no instante t . O objectivo é encontrar um bloco candidato, com a mesma dimensão do bloco de referência, que minimiza uma medida de distância (ou distorção). A diferença entre a posição do bloco de referência em relação ao bloco candidato corresponde ao vector de movimento.

Geralmente, utilizam-se as seguintes medidas de distância, por ordem decrescente de complexidade:

Função de Correlação Normalizada

$$FCN(i, j) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_t(m, n) \cdot x_{t-1}(m+i, n+j)}{\left[\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_t^2(m, n) \right]^{\frac{1}{2}} \left[\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{t-1}^2(m+i, n+j) \right]^{\frac{1}{2}}} \quad (2.17)$$

Erro Quadrático Médio

$$EQM(i, j) = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x_t(m, n) - x_{t-1}(m+i, n+j)]^2 \quad (2.18)$$

Erro Absoluto Médio

$$EAM(i, j) = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x_t(m, n) - x_{t-1}(m+i, n+j)| \quad (2.19)$$

Em que x_t e x_{t-1} representam as imagens nos instantes t e $t-1$, respectivamente, e i, j definem as dimensões da área de pesquisa, na horizontal e na vertical, respectivamente.

2.5 Medidas de Eficiência da Codificação de Vídeo

Para comparar o desempenho das técnicas de codificação, são geralmente usadas medidas de avaliação com base em parâmetros que relacionam a sequência original com a sequência de imagens codificadas. Estas medidas relacionam diversas grandezas mensuráveis, tais como as dimensões das imagens, a quantidade de dados para as representar e o nível de ruído associado, e traduzem o grau de aproximação obtida pelo processo de codificação.

Por outro lado, tal avaliação não invalida a comparação das imagens originais e codificadas por simples observação, pese embora o grau de subjectividade associado. Esta avaliação, apesar de subjectiva, pode revelar-se um método bastante efectivo. De facto, uma técnica que possibilite um bom factor de compressão mas provoque uma degradação visível significativa das imagens, pode não ter grande utilidade.

Geralmente usam-se os seguintes parâmetros de medida do desempenho na codificação.

Factor de Compressão

O factor de compressão é definido pelo quociente entre as dimensões da sequência original e da sequência codificada, em *bits*,

$$\gamma = \frac{S_o}{S_c} \quad (2.20)$$

em que S_o é a dimensão da sequência original e S_c é a dimensão da sequência codificada.

Erro Quadrático Médio

As técnicas de codificação que implicam perda de informação, devem ser ainda comparadas através de parâmetros que permitam avaliar a degradação da imagem. A degradação espacial de imagens é geralmente avaliada com base no erro quadrático médio (MSE). Para uma sequência de L imagens de dimensão $M \times N$,

$$MSE = \frac{\sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I_l(m,n) - I'_l(m,n)]^2}{L \cdot M \cdot N} \quad (2.21)$$

Na equação 2.21, $I_l(m,n)$ representa o *pixel* da posição (m,n) da imagem l na sequência original, e $I'_l(m,n)$ representa o *pixel* da posição (m,n) da imagem l na sequência reconstruída.

Relação Sinal Ruído

Outro parâmetro que também permite avaliar a degradação das imagens, considerando-se a degradação resultante da adição de ruído com distribuição uniforme, é a relação sinal ruído (SNR), definida pela equação 2.22.

$$SNR = 10 \cdot \log_{10} \left\{ \frac{\sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I_l(m,n)]^2}{\sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I_l(m,n) - I'_l(m,n)]^2} \right\} \quad (\text{dB}) \quad (2.22)$$

Relação Sinal Ruído de Pico

É também comum usar-se outro parâmetro, derivado do MSE, a relação sinal ruído de pico (PSNR), definido pela equação 2.23. Nesta equação A representa a amplitude máxima do *pixel*, em que $A = 255$ em sistemas onde o *pixel* seja representado por 8 *bits*.

$$PSNR = 10 \cdot \log_{10} \frac{A^2}{MSE} \quad (\text{dB}) \quad (2.23)$$

Os parâmetros mais utilizados são o erro quadrático médio e a relação sinal ruído de pico. Faz-se notar que a valores diferentes de MSE ou PSNR podem corresponder imagens com degradação aparentemente igual. Noutros casos, para valores idênticos de MSE ou PSNR, as imagens podem ter uma qualidade aparente muito diferente. Isto depende do observador, e deve-se à não uniformidade da sensibilidade da visão humana. O uso destes parâmetros surge assim frequentemente associado a critérios classificadores, da qualidade ou degradação das imagens, de carácter subjectivo.

2.6 Conclusões

Neste capítulo apresentaram-se, sucintamente, as principais técnicas para a codificação de imagem e vídeo.

A transmissão de vídeo digital envolve, regra geral, ritmos binários elevados. Por outro lado, as imagens consecutivas das sequências de vídeo são fortemente correlacionadas, tanto no tempo como no espaço. Para reduzir o ritmo binário, usam-se técnicas que exploram as redundâncias espaciais e temporais, recorrendo a técnicas de sub amostragem, ao uso de transformadas com quantificação dos coeficientes, e de esquemas de codificação do tipo entrópico.

A transformada mais utilizada tem sido a DCT, apesar do aparecimento de “efeito de bloco”, nomeadamente para ritmos de transmissão reduzidos. A transformada de ôndulas discreta, tal como outras transformadas, permite a decorrelação efectiva dos sinais, concentrando a energia numa região de dimensões mais reduzidas, apresentando como vantagem o não aparecimento do “efeito de bloco”. Além disso, a transformada de ôndulas gera um esquema de decomposição em sub-bandas, com vantagens na realização de esquemas efectivos de codificação hierárquica, com controlo do factor de compressão.

No caso de transmissão de vídeo em canais de banda estreita, de modo a minimizar o efeito de degradação das imagens devido aos elevados factores de compressão requeridos, usam-se técnicas de estimação e compensação de movimento. No processo de estimação e compensação de movimento, a obtenção de resultados óptimos tem um elevado peso computacional, pelo que se recorre geralmente a algoritmos de cálculo rápido mas que produzem resultados sub-óptimos.

As medidas para avaliar o desempenho dos esquemas de codificação, compreendem parâmetros como o factor de compressão e a relação Sinal/Ruído, para medir a degradação associada ao esquema de codificação. Além destas, é também comum estabelecerem-se critérios de natureza subjectiva na avaliação da degradação das imagens codificadas.

Capítulo 3

TRANSFORMADA DE ÔNDULAS E ESQUEMA PROGRESSIVO

Neste capítulo apresentam-se os aspectos teóricos fundamentais relativamente às ôndulas e à transformada de ôndulas, destacando em particular a obtenção da transformada de ôndulas pelo esquema progressivo (*lifting scheme*). Estes aspectos são relevantes para o enquadramento do trabalho desenvolvido, considerando-se importantes para a compreensão dos capítulos seguintes.

3.1 Introdução

O método clássico para analisar o espectro de um sinal é a análise de Fourier, sendo comum a representação do sinal através da combinação linear de funções de base. Um exemplo típico é o desenvolvimento de uma função em série de Fourier em que o conjunto, $\{1, \sin(nx), \cos(nx) : n \in \mathbb{N}\}$, forma a base do espaço discreto de Fourier. Nesta dissertação, os termos **função** e **sinal** são usados indistintamente para designar variáveis dependentes do tempo ou do espaço, considerando uma variável genérica, x , sem referir especificamente qual o seu domínio. Como na análise de Fourier as funções de base são ortonogonais, os coeficientes da série podem ser expressas através do produto interno da função e da base. Considere-se, por exemplo, o desenvolvimento em série de uma função (f) de período 2π , através do produto interno entre a função e a função de base (v).

$$\langle f, v \rangle = \int_0^{2\pi} f(x) \cdot v(x) \cdot dx \quad (3.1)$$

O desenvolvimento da função em série de Fourier (as bases são sinusoides) é dado por,

$$f(x) = a_0 + \sum_{k=1}^{\infty} [a_k \cdot \cos(kx) + b_k \cdot \sin(kx)] \quad (3.2)$$

sendo os coeficientes da série obtidos por,

$$a_0 = \langle f, 1 \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x) \cdot dx \quad (3.3 \text{ a})$$

$$a_k = \langle f, \cos(kx) \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \cdot \cos(kx) \cdot dx \quad (3.3 \text{ b})$$

$$b_k = \langle f, \sin(kx) \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \cdot \sin(kx) \cdot dx \quad (3.3 \text{ c})$$

Porém, a análise de Fourier não é adequada nos casos em que a função não seja periódica ou tenha suporte compacto, isto é, apenas tenha valores não nulos num dado intervalo, e nulos fora desse intervalo. Isto é justificado pelo facto das funções de base da análise de Fourier não serem temporalmente localizadas (não se cancelam fora de um dado intervalo) e, portanto, a série converge lentamente.

Um método clássico para analisar o espectro local de um sinal, consiste no uso da transformada de Fourier local (STFT) em que a função apenas tem que ser definida numa determinada escala temporal (“janela”). A função “janela” deve ser bem localizada e ter suporte compacto. A compressão no tempo (redução na dimensão da janela) conduz a uma dilatação na frequência, e vice-versa. A figura de mérito da “janela”, definida pelo produto tempo-frequência, é limitada pelo princípio de incerteza de Heisenberg¹¹. À medida que a “janela” é mais reduzida no tempo, diminui-se a resolução na frequência, e vice-versa. Com a alteração das dimensões da “janela” é necessário calcular novamente a STFT para as novas condições, com custos evidentes no tempo computacional.

¹¹ O princípio de incerteza de Heisenberg, definido pela inequação $\Delta x \Delta \omega \geq \frac{1}{2}$, estabelece uma limitação inferior na resolução no tempo e na frequência. As regiões do espaço identificadas pelas variações na escala do tempo e da frequência, em que se verifique $\Delta t \cdot \Delta \omega = 2\pi$, designam-se por rectângulos de Heisenberg.

Por outro lado, na STFT, as funções sinusoidais são moduladas pela função envolvente. Como a função de análise é a mesma para cada frequência, a escala no plano tempo-frequência é constante, independentemente da sua localização temporal.

Tal como na STFT, a ideia fundamental associada à transformada de ondas é a análise e representação de funções (sinais) numa escala adequada, dado que a transformada de Fourier local, e a transformada de ondas, permitem a análise e a representação no plano tempo-frequência.

3.2 Ondas e Transformada de Ondas

As ondas são funções de análise e representação escaláveis, isto é, podem ser dilatadas ou comprimidas no domínio do tempo, sendo bem localizadas nos domínios do tempo e da frequência. A transformada de ondas usa ondas como funções de análise, às quais estão associados parâmetros de deslocamento (translação) e de escala (dilatação/compressão). Uma onda é uma função $\psi \in L^2(\mathbb{R})$ e a sua versão escalada e deslocada é definida por,

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \cdot \psi\left(\frac{x-b}{a}\right), \quad a > 0, b \in \mathbb{R} \quad (3.4)$$

em que os parâmetros a e b são os parâmetros de escala e de translação, respectivamente.

A equação 3.4 define a forma como são geradas as translações e dilatações da onda de base, também designada por onda mãe (*mother wavelet*). O parâmetro escala está inversamente associado à frequência, isto é, a factores de escala baixos correspondem altas frequências, e vice-versa. Em aplicações em que é frequente a análise de sinais de carácter transitório, com componentes de alta e baixa frequência (como por exemplo em processamento de imagem e de vídeo), esta transformada permite adaptar a resolução da análise, consoante a frequência e a ocorrência no tempo. Face à STFT, a transformada de ondas apresenta vantagens, por permitir obter uma resolução ajustável no plano tempo-escala (e não fixa como na STFT). Com os recentes avanços registados no estudo da ondas, e na transformada de ondas pelo esquema progressivo em particular, o tempo computacional requerido para o seu cálculo é também mais reduzido.

3.2.1 Ôndulas de Bases Ortonormais

Os vários tipos de ôndulas podem ser classificadas em: ôndulas elementares (*crude wavelets*), ôndulas infinitamente regulares, ôndulas ortogonais de suporte compacto, ôndulas biortogonais de suporte compacto e ôndulas complexas. No grupo das ôndulas elementares encontram-se a ôndula Gaussiana, a ôndula de Morlet e a ôndula “chapéu mexicano” (*mexican hat*); no grupo das ôndulas infinitamente regulares inclui-se a ôndula de Meyer; e no grupo das ôndulas complexas, as ôndulas complexas Gaussianas, complexas de Morlet, complexas de Shannon e B-Splines complexas. As ôndulas destes três grupos apresentam simetria, mas não têm suporte compacto.

Para o processamento digital de sinal são fundamentais as seguintes características: capacidade de análise e de reconstrução, e a existência de algoritmos eficientes para o cálculo. O suporte compacto é de especial importância pois, como será referido mais adiante, na análise multiresolução, correspondem à aplicação de filtros com resposta impulsiva finita (FIR).

As ôndulas ortogonais e biortogonais com suporte compacto verificam estas características, e, por esse motivo, são as mais usadas em processamento de sinal. Nesta tese apenas se consideram os casos particulares das ôndulas ortogonais de Daubechies (DB_n) e as ôndulas biortogonais de Daubechies, ambas com suporte compacto, nomeadamente: a ôndula Daubechies 2^{12} , ou DB_2 , e as ôndulas biortogonais (4,4), ou ôndulas Daubechies $9/7^{13}$.

A definição de ôndulas ortogonais e biortogonais é feita mais adiante. Porém, pode-se desde já referir uma das principais diferenças entre este tipo de ôndulas: as ôndulas biortogonais permitem o uso de duas funções diferentes para análise e para síntese. Obtêm-se assim ôndulas simétricas de suporte compacto, para análise e para síntese, e funções de escala, o que não é possível com bases ortonormais contínuas, como no caso da decomposição ortogonal, em que a mesma ôndula é usada tanto para análise como para síntese. Uma característica importante das ôndulas é a possibilidade de se gerarem bases ortonormais em $L^2(\mathbb{R})$ [Gosw+99, Daub_92].

A base mais simples que se conhece (e também a mais antiga) é a ôndula de Haar definida por:

¹² Ôndula ortogonal Daubechies 2 tem 2 momentos nulos e os filtros associados têm 4 coeficientes. Por este facto, esta ôndula é também por vezes referenciada por ôndula D4.

¹³ As ôndulas biortogonais (4,4) são também designadas por par de filtros 9/7, ou Daubechies 9/7, porque o número efectivo de coeficientes (coeficientes não nulos) dos filtros é 9 para o filtro de análise, e 7 para o filtro de síntese.

$$\psi(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & x \notin [0, 1] \end{cases} \quad (3.5)$$

Contudo, a base de Haar definida pela equação 3.5 não é completa, isto é, existem determinadas sequências que não podem ser representadas por esta base. Complementando o espaço obtém-se a base de Haar da equação 3.6, que já permite representar qualquer sequência.

$$\psi(x) = \begin{cases} 1, & 0 \leq x < \frac{1}{2} \\ -1, & \frac{1}{2} \leq x < 1 \\ 0, & x \notin [0, 1[\end{cases} \quad (3.6)$$

Porém, esta função analiticamente muito simples não tem boa localização no plano tempo-frequência, a sua transformada de Fourier, $\hat{\psi}(\xi)$, apresenta um decaimento na frequência, ξ , próximo de $|\xi|^{-1}$, quando $\xi \rightarrow \infty$. Além disso, não tem derivadas contínuas em alguns pontos e por isso diz-se que tem fraca regularidade.

Por escolha adequada da função ψ e dos parâmetros a e b , [Daub_92] obtém-se,

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \cdot \psi(2^{-m}x - n) \quad m, n \in \mathbb{Z} \quad (3.7)$$

As funções obtidas pela equação 3.7 constituem uma base ortonormal no espaço $L^2(\mathbb{R})$, permitindo, portanto, que qualquer função nesse espaço seja aproximada por combinação linear de $\psi_{m,n}$. O índice m é o parâmetro de escala (resolução) e o índice n é o parâmetro de translação. Com base na definição de bases ortonormais podem-se definir as bases ortogonais e biortogonais.

DEFINIÇÃO 1: O conjunto $\{f_i \in L^2(\mathbb{R}) : i \in \mathbb{Z}\}$ é uma base ortonormal, se para quaisquer funções $f_i, f_j \in L^2(\mathbb{R})$, com $i, j \in \mathbb{Z}$, se verificar,

$$\langle f_i, f_j \rangle = \delta_{i,j}$$

em que $\delta_{i,j}$ é a função delta de Kronecker definida por,

$$\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Se qualquer função g pertencente ao mesmo conjunto, puder ser expressa por um conjunto ortonormal $\{f_i \in L^2(\mathbb{R}) : i \in \mathbb{Z}\}$, é possível desenvolver a função na forma de uma série,

$$g = \sum_{k \in \mathbb{Z}} a_k \cdot f_k$$

em que os coeficientes a_k são obtidos pelo produto interno da função com a função de base,

$$a_k = \langle g, f_k \rangle = \sum_l a_l \cdot \delta_{l,k} \quad \square$$

DEFINIÇÃO 2: Seja o conjunto de funções $\{f_i \in L^2(\mathbb{R}) : i \in \mathbb{Z}\}$ uma base biortogonal. Se existir um outro conjunto de funções $\{\tilde{f}_j \in L^2(\mathbb{R}) : j \in \mathbb{Z}\}$ tal que,

$$\langle f_i, \tilde{f}_j \rangle = \delta_{i,j}$$

então o conjunto $\{\tilde{f}_j \in L^2(\mathbb{R}) : j \in \mathbb{Z}\}$ é uma base dual de $\{f_i \in L^2(\mathbb{R}) : i \in \mathbb{Z}\}$.

É possível desenvolver qualquer função h em série, usando a base biortogonal,

$$h = \sum_{k \in \mathbb{Z}} b_k \cdot f_k$$

em que,

$$b_k = \langle g, f_k \rangle = \sum_l b_l \cdot \delta_{l,k}$$

É igualmente possível desenvolver a função h em série, usando a base dual,

$$h = \sum_{k \in \mathbb{Z}} \tilde{b}_k \cdot \tilde{f}_k$$

obtendo-se, neste caso, os coeficientes duais,

$$\tilde{b}_l = \langle h, \tilde{f}_l \rangle = \sum_k b_k \cdot \delta_{k,l} \quad \square$$

Generalizando a base de Haar, I. Daubechies propôs um conjunto de bases ortogonais de suporte compacto: a família de ôndulas de Daubechies, DBn , em que n indica o número de momentos nulos [Daub_92]. Enquanto a ôndula de Haar não tem derivadas contínuas, as ôndulas de Daubechies têm, pelo menos, uma derivada contínua.

3.2.2 Transformada de Ôndulas Contínua e Discreta

A transformada de ôndulas contínua (CWT) de uma função, $f(x)$, [Gosw+99], referente a uma ôndula de análise, ψ , é, por definição,

$$W_{a,b}f(x) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x) \overline{\psi_{a,b}(x)} \cdot dx \quad (3.8)$$

em que $\overline{\psi(\cdot)}$ significa o complexo conjugado de $\psi(\cdot)$, e

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \cdot \psi\left(\frac{x-b}{a}\right) \quad a, b \in \mathbb{R} \wedge a > 0 \quad (3.9)$$

O factor $\frac{1}{\sqrt{a}}$ é o factor de normalização de energia ao longo das várias escalas.

Tal como a transformada de Fourier discreta (DFT), foi introduzida a transformada de ôndulas discreta (DWT) que, quando associada ao conceito de análise multiresolução, permite obter algoritmos eficientes para cálculo dos coeficientes da transformada de ôndulas. Desenvolvendo a função em série de ôndulas (*wavelet series*), verifica-se que os coeficientes da série correspondem aos coeficientes da transformada de ondulas contínua, obtidos em determinados pontos, espaçados de um factor 2, isto é $\left(\frac{1}{2^j}, \frac{k}{2^j}\right)$ [Gosw+99].

Partindo das expressões 3.8 e 3.9, restringindo a a valores positivos, assumindo que ψ verifica a condição de admissibilidade [Gosw+99], considerando as bases ortogonais em $L^2(\mathbb{R})$, e tomando os valores discretos dos parâmetros de escala e translação, $a = 2^{-j}$ e b da forma $b = k2^{-j}$, em que $k, j \in \mathbb{Z}$, obtém-se:

$$w_{k,j} = W_{\psi}f\left(\frac{1}{2^j}, \frac{k}{2^j}\right) = \int_{-\infty}^{\infty} f(x) \overline{\psi\left(\frac{x - k/2^j}{1/2^j}\right)} \cdot dx \quad (3.10)$$

A equação 3.10 corresponde ao desenvolvimento de uma função em série de ôndulas; gerando um conjunto discreto de valores da correlação entre a ôndula ψ e a função $f(x)$, com representação conjunta no plano tempo-escala, isto é, a transformada de ôndulas.

3.3 Análise Multiresolução e Decomposição em Sub-Bandas

O método de análise multiresolução, ou análise com resolução múltipla (MRA), introduzido por S. Mallat e Y. Meyer, constitui uma das aplicações mais importantes da transformada de ondas em processamento de sinal.

Considere-se, por exemplo, um sinal constituído por variações lentas e rápidas no tempo, isto é, componentes de baixa e alta-frequência, respectivamente. A representação discreta deste sinal implica um número elevado de amostras (teorema de Shannon). Uma forma mais eficiente de representar o sinal no domínio discreto, consiste na utilização de várias taxas de amostragem (resoluções). A aproximação obtida para a maior escala (baixa frequência), conjuntamente com os detalhes obtidos nas restantes escalas, permitem a representação eficiente da função original. A ideia fundamental associada à MRA é a decomposição de uma função arbitrária, $f \in L^2(\mathbb{R})$, em várias funções mais simples, por aproximação em escalas sucessivas, isto é, a representação da função f com vários níveis de detalhe. Como os parâmetros de escala são potências de base 2 (eq. 3.10), as várias escalas de frequência organizam-se em oitavas.

3.3.1 Análise com Resolução Múltipla

Considere-se o conjunto de sub espaços fechados, $V_j, (j \in \mathbb{Z})$, tal que:

$$\{0\} \subset \dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \subset L^2(\mathbb{R}) \quad (3.11)$$

$$\overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbb{R}), \quad \bigcap_{j=-\infty}^{\infty} V_j = \{0\} \quad (3.12)$$

e uma função de escala $\phi \in V_0$, tal que as suas versões deslocadas $(\phi_{0,x})$

$$\{\phi_{0,k} = \phi(x-k) : k \in \mathbb{Z}\} \text{ base ortonormal de } V_0 \quad (3.13)$$

constituem uma base ortonormal de V_0 . Pelas relações entre os espaços na MRA, uma função f move-se nos espaços quando escalada de um factor 2,

$$f \in V_j \Leftrightarrow f(2 \cdot) \in V_{j+1} \Leftrightarrow f(2^{-j} \cdot) \in V_0 \quad (3.14)$$

Os axiomas 3.11 a 3.14 caracterizam completamente a MRA. Através de 3.14 verifica-se que todos os sub espaços correspondem a versões escaladas do espaço original (com factor 2).

A partir destes axiomas verifica-se o encadeamento¹⁴ dos sub espaços, $V_j \in L^2(\mathbb{R})$, tal que a sua soma directa é $L^2(\mathbb{R})$; os vários sub espaços correspondem a versões escaladas do espaço central, V_0 , com o factor de escala 2. É ainda requerido que a função de escala, ϕ , tenha área unitária [Daub_92 (pp. 175)],

$$\int_{-\infty}^{\infty} \phi(x) \cdot dx = 1 \quad (3.15)$$

Definindo o operador P_j , das projecções ortogonais da função $f \in L^2(\mathbb{R})$ no sub espaço V_j , verifica-se que as projecções correspondem a aproximações à função f , tal que $\lim_{j \rightarrow \infty} P_j f = f$, i. e., as sucessivas projecções permitem obter aproximações mais detalhadas à função f . Com base no encadeamento dos espaços definido anteriormente definem-se os espaços complementares, W_j , ortogonais a V_j , $V_j \perp W_j$, representados na figura 3.1.

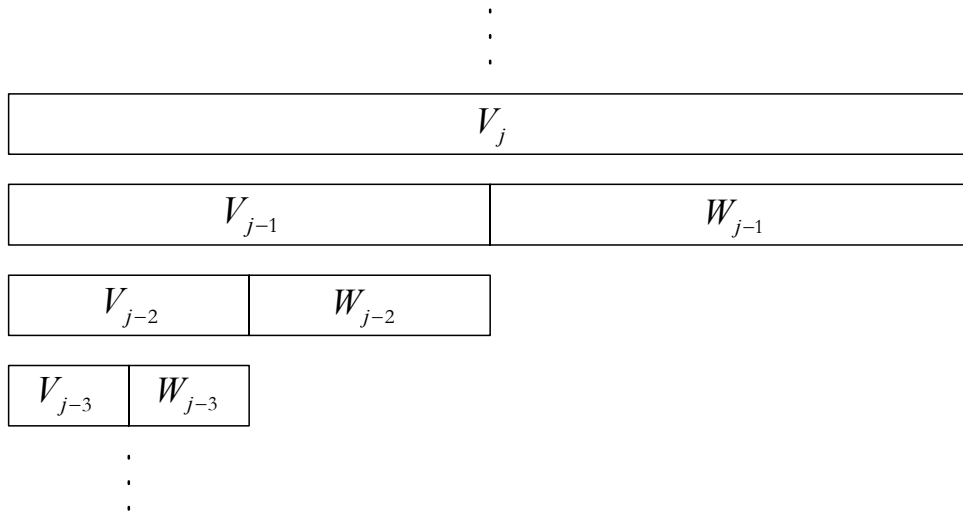


Figura 3.1: Esquema de decomposição dos sub espaços na MRA.

O sub espaço V_j designa-se por espaço de aproximações, e o sub espaço W_j designa-se por espaço de ôndula ou de detalhe, sendo complementar a V_j em V_{j+1} . Como a decomposição é ortogonal,

$$V_j \cap W_j = \{0\} \quad j \in \mathbb{Z} \quad (3.16)$$

$$V_j \oplus W_j = V_{j+1} \quad (3.17)$$

A partir das expressão (3.17) e de acordo com a caracterização da MRA obtém-se,

$$f \in W_j \Leftrightarrow f(2^j \cdot) \in W_0 \quad (3.18)$$

$$V_{j+1} = V_j \oplus W_j = W_j \oplus W_{j-1} \oplus V_{j-1} = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \dots$$

$$V_j = \bigoplus_{l=-\infty}^{j-1} W_l \quad (3.19)$$

De igual modo, como os espaços W_j são ortogonais entre si, isto é, $W_j \perp W_{j'}$, $j \neq j'$

$$\bigoplus_{j \in \mathbb{Z}} W_j = L^2(\mathbb{R}) \quad (3.20)$$

Pela figura 3.1 e pelas expressões anteriores, verifica-se que os sub espaços V_j são gerados por bases $\left\{ \phi_{k,j} : 2^{j/2} \phi(2^j x - k) : k \in \mathbb{Z} \right\}$ e os sub espaços W_j por $\left\{ \psi_{k,j} : 2^{j/2} \psi(2^j x - k) : k \in \mathbb{Z} \right\}$. A função ϕ designa-se função base de escala, e a função ψ é a ôndula de base, ou ôndula mãe (*mother wavelet*, em Inglês) e j representa as diferentes escalas. Ou seja, quaisquer funções $f_j(x) \in V_j$ e $g_j(x) \in W_j$ podem ser representadas por combinações lineares de $\phi_{k,j}(x)$ e $\psi_{k,j}(x)$, respectivamente.

$$f_j(x) = \sum_k a_{k,j} \cdot \phi(2^j x - k) \quad (3.21)$$

$$g_j(x) = \sum_k w_{k,j} \cdot \psi(2^j x - k) \quad (3.22)$$

com $\left\{ a_{k,j} \right\}_{k \in \mathbb{Z}}$, $\left\{ w_{k,j} \right\}_{k \in \mathbb{Z}} \in l^2$, em que o espaço l^2 corresponde a $L^2(\mathbb{R})$ para sequências discretas.

¹⁴ A notação no encadeamento dos sub-espacos não é consensual. Alguns autores usam uma notação idêntica à que é usada nesta tese, por exemplo [Gosw+99], outros, como [Daub_92], usam uma notação na ordem inversa de encadeamento.

As funções $f_{j-1}(x) \in V_{j-1}$ e $g_{j-1}(x) \in W_{j-1}$, podem ser ambas geradas a partir de $f_j(x) \in V_j$, pois as bases $\phi_{k,j-1}(x)$ de V_{j-1} e $\psi_{k,j-1}(x)$ de W_{j-1} , estão relacionadas com $\phi_{k,j}(x)$ de V_j . A decomposição dos sub espaços, V_j e W_j , efectuada nas condições anteriores, é uma decomposição ortonormal, cuja relação com a decomposição em sub-bandas é exemplificada na figura 3.2.

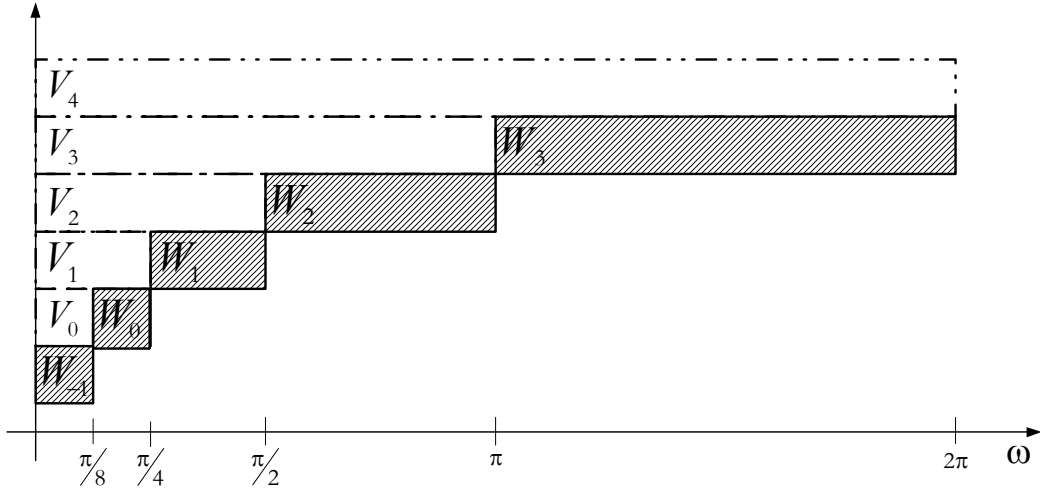


Figura 3.2: Decomposição em sub-bandas de frequência (oitavas), e sua relação com os espaços ortonormais.

Como $V_j \subset V_{j+1}$, qualquer função em V_j pode ser desenvolvida em série usando as funções de base. Para o caso em que $V_0 \subset V_1$, em particular para $\phi(x) = \phi_{0,0}(x) \in V_0$, tem-se

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \cdot \phi_{1,k}(x) = \sqrt{2} \cdot \sum_{k=-\infty}^{\infty} a_k \cdot \phi(2x - k) \quad (3.23)$$

em que a_k são os coeficientes dos filtros, e

$$a_k = \int_{-\infty}^{\infty} \phi(x) \cdot \phi_{1,k}(x) \cdot dx \quad (3.24)$$

No caso das funções de suporte compacto, existem apenas alguns valores de a_k diferentes de zero. Porém, para se obter uma representação única da função ϕ , pelos coeficientes dos filtros (até uma dada escala), ϕ deve ter uma área unitária (eq 3.15). Assim tem-se,

$$\phi(x) = \sqrt{2} \cdot \sum_{k=0}^{D-1} a_k \cdot \phi(2x - k) \quad (3.25)$$

De modo análogo, mas para o espaço de ondas, como $W_0 \subset W_1$:

$$\psi(x) = \sqrt{2} \cdot \sum_{k=0}^{D-1} b_k \cdot \phi(2x - k) \quad (3.26)$$

$$b_k = \int_{-\infty}^{\infty} \psi(x) \cdot \phi_{1,k}(x) \cdot dx \quad (3.27)$$

As expressões 3.25 e 3.26 são conhecidas por equação de dilatação e equação da ôndula, respectivamente. Das expressões 3.24 e 3.27, obtém-se uma equação que permite determinar os coeficientes b_k a partir dos coeficientes a_k [Niels_98]:

$$b_k = (-1)^k a_{D-1-k}, \quad k = 0, 1, \dots, D-1 \quad (3.28)$$

D é um inteiro, positivo e par, e os coeficientes a_k e b_k são os coeficientes dos filtros. O valor $\sqrt{2}$ é um factor de normalização dos coeficientes dos filtros das ôndulas ortogonais e biortogonais [Daub_92], designado por factor de conservação da área (ver apêndice A).

3.3.2 Decomposição Ortogonal e Biortogonal

Considerando que $V_j \oplus W_j = V_{j+1}$ (eq. 3.17) e atendendo a que $V_j \perp W_j$, este processo de decomposição é ortogonal. Como $V_j \perp W_j$, a ôndula $\psi_{j,k} \in W_j$ é ortogonal à função de escala, $\phi_{j,k} \in V_j$. No caso particular do espaço central, $\{\phi(x-k) : k \in \mathbb{Z}\}$ gera V_0 e $\{\psi(x-k) : k \in \mathbb{Z}\}$ gera W_0 . Como $V_0 \perp W_0$ tem-se:

$$\int_{-\infty}^{\infty} \phi(x) \cdot \psi(x-l) \cdot dx = 0, \quad \forall l \in \mathbb{Z} \quad (3.29)$$

Porém, as funções de base $\{\phi(x-k) : k \in \mathbb{Z}\}$ e $\{\psi(x-k) : k \in \mathbb{Z}\}$, não têm que ser necessariamente ortogonais entre si (as funções ϕ e ψ , individualmente), isto é, pode-se ter:

$$\int_{-\infty}^{\infty} \phi(x) \cdot \phi(x-l) \cdot dx \neq 0 \quad (3.30)$$

$$\int_{-\infty}^{\infty} \psi(x) \cdot \psi(x-l) \cdot dx \neq 0 \quad (3.31)$$

Como os sub espaços W_j são ortogonais entre si, as ôndulas são ortogonais ao longo das várias escalas. Considere-se a ôndula $\psi_{j,k} \in W_j$ e a sua função dual $\hat{\psi}_{j,k} \in \tilde{W}_j$, e também a função escala, $\phi_{k,j} \in V_j$ e a sua dual, $\tilde{\phi}_{k,j} \in \tilde{V}_j$. Embora a ortogonalidade dos espaços V_j e W_j

não seja um requisito, é obrigatório que se verifique, $V_j \perp \tilde{W}_j$. Assim, enquanto $\phi_{k,j}$ e $\psi_{k,j}$ estão associadas a V_j , $\tilde{\phi}_{k,j}$ e $\hat{\psi}_{k,j}$ estão associadas ao sub espaço \tilde{V}_j , designando-se este processo por decomposição biortogonal. Na decomposição biortogonal, enquanto a função dual da ôndula, $\tilde{\psi}$, pode ser usada na análise de uma função numa dada localização tempo-escala, a ôndula pode ser usada para representação da função. Por este motivo a dual da ôndula designa-se por ôndula de análise, e a ôndula designa-se por ôndula de síntese.

As condições de ortogonalidade [Gosw+99, Niels_98] são de definidas por:

$$\int_{-\infty}^{\infty} \phi_{j,k}(x) \cdot \phi_{j,l}(x) \cdot dx = \delta_{k,l} \quad (3.32)$$

$$\int_{-\infty}^{\infty} \psi_{i,k}(x) \cdot \tilde{\psi}_{j,l}(x) \cdot dx = \delta_{i,j} \cdot \delta_{k,l} \quad (3.33)$$

$$\int_{-\infty}^{\infty} \phi_{i,k}(x) \cdot \tilde{\psi}_{j,l}(x) \cdot dx = 0 \quad (3.34)$$

3.4 Transformada de Ôndulas Discreta e Bancos de Filtros

Dada a possibilidade de representar funções nos espaços da MRA através dos coeficientes dos filtros (equações 3.25 e 3.26), e atendendo a que o espectro da função de escala e o espectro da ôndula equivalem a filtros passa-baixo e passa-alto, respectivamente, as diversas escalas da MRA equivalem a bancos de filtros.

Assim, a decomposição ortonormal de um sinal discreto pode ser associada a bancos de filtros, constituídos em cada nível por um par de filtros (de análise) passa-baixo, \tilde{h} , e passa-alto, \tilde{g} , como ilustra a figura 3.3.

Na figura 3.3, a sequência discreta inicial corresponde à aproximação a uma função $x(t)$, isto é, $x[n] = a_{0,n}$. O símbolo $\downarrow 2$ representa o processo de decimação por um factor de 2. Os filtros têm resposta impulsiva $\{\tilde{h}_k\}$ e $\{\tilde{g}_k\}$, respectivamente [Daub_92].

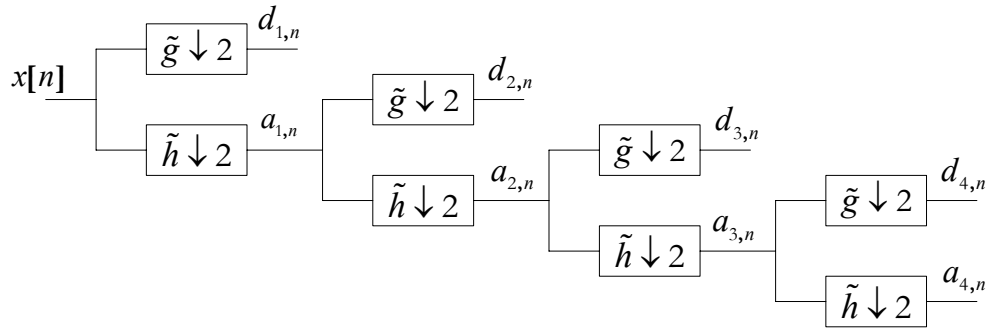


Figura 3.3: Decomposição de um sinal discreto por bancos de filtros.

Segundo este esquema, em cada andar de decomposição é feita a decimação e a convolução da sequência com os coeficientes dos filtros. Em cada estágio, obtém-se a aproximação à sequência original, $a_{k,n}$, e as diferenças (ou detalhes) entre os valores da sequência, $d_{k,n}$, isto é, a transformada de ondas discretas (DWT). Porém, convém referir que a DWT não é invariante ao deslocamento, isto é, a DWT de uma função e a DWT da sua versão deslocada no tempo, não correspondem a versões da DWT, deslocadas uma da outra.

O processo pode ser invertido para reconstrução, através da DWT^{-1} , obtendo-se assim bancos de filtros de decomposição e de reconstrução. Para garantir a reconstrução, num banco de filtros ortogonais, os filtros de síntese, $\{h_k\}$ e $\{g_k\}$, devem ser os duais dos filtros de análise, isto é, a sua resposta impulsiva é temporalmente inversa. Em [Daub_92] estabelecem-se as relações entre os filtros de análise e de síntese:

$$\tilde{g}_k = (-1)^{k+1} h_{1-k} \quad (3.35)$$

$$g_k = (-1)^{k+1} \tilde{h}_{1-k} \quad (3.36)$$

Existe reconstrução perfeita se a sequência reconstruída for uma réplica da sequência original, podendo estar afectada de um atraso ou de um factor de escala, isto é, se verificarem as condições de reconstrução perfeita, como se verá mais adiante.

Nota: É comum associar a MRA a bancos de filtros QMF que se caracterizam pelas expressões:

$$h_n = h_{-n}$$

$$g_n = 1^{-n} h_n$$

$$\tilde{h}_n = h_{-n}$$

$$\tilde{g}_n = 1^{-n} h_n$$

Porém, note-se que os filtros QMF não correspondem a filtros de ôndulas.

3.4.1 Filtros de Decomposição e de Reconstrução

Atendendo ao encadeamento e às relações entre os espaços na MRA, para qualquer $j \in \mathbb{Z}$, existem duas sequências de coeficientes, $\{h_k\}, \{g_k\} \in l^2$, tais que,

$$\phi(2^j x) = \sum_k h_k \cdot \phi(2^{j+1} x - k) \quad (3.37)$$

$$\psi(2^j x) = \sum_k g_k \cdot \phi(2^{j+1} x - k) \quad (3.38)$$

Note-se que no caso das ôndulas com suporte compacto (expressões 3.25 e 3.26), as expressões anteriores são afectadas do coeficiente $\sqrt{2}$.

Para a decomposição, atendendo a que $V_{j+1} = V_j + W_j$ e $\phi(2x), \phi(2x-1) \in V_j$, existem duas sequências $\{\tilde{h}[k]\}, \{\tilde{g}[k]\} \in l^2$ tais que,

$$\phi(2x) = \sum_k \{\tilde{h}[2k] \cdot \phi(x-k) + \tilde{g}[2k] \cdot \psi(x-k)\} \quad (3.39)$$

$$\phi(2x-1) = \sum_k \{\tilde{h}[2k-1] \cdot \phi(x-k) + \tilde{g}[2k-1] \cdot \psi(x-k)\} \quad (3.40)$$

Combinando a expressões 3.39 e 3.40 e generalizando, obtém-se:

$$\phi(2^{j+1} x - l) = \sum_k \{\tilde{h}[2k-l] \cdot \phi(2^j x - k) + \tilde{g}[2k-l] \cdot \psi(2^j x - k)\} \quad (3.41)$$

De modo análogo, é possível reconstruir a função usando um bancos de filtros duais, relativamente aos filtros de análise.

A figura 3.4 ilustra um bloco de decomposição e reconstrução com dois canais, composto por uma secção de análise (ou de decomposição) e uma secção de síntese (ou de reconstrução). A secção de análise inclui um filtro passa-baixo, \tilde{h} , e um filtro passa-alto, \tilde{g} , complementares entre si; o mesmo se aplica aos filtros h e g , da secção de síntese.

Considere-se a transformada Z das seqüências discretas a_j e d_j , obtidas por separação dos coeficientes de ordem par e ímpar de $x[n]$, no sistema da figura 3.4, em que por simplicidade se assume não existir qualquer tipo de processamento. A transformada Z da resposta dos filtros de análise, $\tilde{h}(z)$ e $\tilde{g}(z)$, por convolução com as subsequências de entrada, seguida de decimação por um de factor 2 (retêm-se as amostras pares), respectivamente, é dada por:

$$a(z) = \frac{1}{2} \left[x(z^{1/2}) \cdot \tilde{h}(z^{1/2}) + x(-z^{1/2}) \cdot \tilde{h}(-z^{1/2}) \right] \quad (3.42)$$

$$d(z) = \frac{1}{2} \left[x(z^{1/2}) \cdot \tilde{g}(z^{1/2}) + x(-z^{1/2}) \cdot \tilde{g}(-z^{1/2}) \right] \quad (3.43)$$

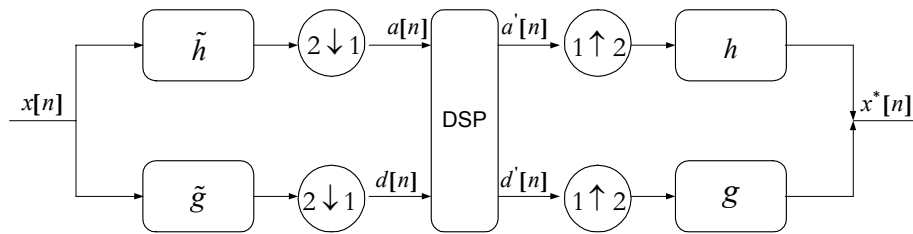


Figura 3.4: Bancos de filtros com decomposição e reconstrução em 2 sub-bandas.

Considerando que não existe processamento, tem-se,

$$a(z) = a'(z) \quad \text{e} \quad d(z) = d'(z) \quad (3.44)$$

A resposta dos filtros de síntese e interpolação, precedida de inserção de zeros, é dada por:

$$a''(z) = \frac{1}{2} \left[x(z) \cdot \tilde{h}(z) \cdot h(z) + x(-z) \cdot \tilde{h}(-z) \cdot h(z) \right] \quad (3.45)$$

$$d''(z) = \frac{1}{2} \left[x(z) \cdot \tilde{g}(z) \cdot g(z) + x(-z) \cdot \tilde{g}(-z) \cdot g(z) \right] \quad (3.46)$$

A reconstrução da seqüência original obtém-se por combinação das subsequências, tal que,

$$\begin{aligned} x^*(z) &= a''(z) + d''(z) = \\ &= \frac{1}{2} x(z) \left[\tilde{h}(z) \cdot h(z) + \tilde{g}(z) \cdot g(z) \right] + \frac{1}{2} x(-z) \left[\tilde{h}(-z) \cdot h(z) + \tilde{g}(-z) \cdot g(z) \right] \end{aligned} \quad (3.47)$$

O segundo termo desta equação contém os efeitos de sobreposição espectral da seqüência de entrada [Gosw+99]. Para os eliminar escolhem-se os filtros de tal modo que:

$$h(z) = \pm \tilde{g}(-z) \quad (3.48)$$

$$g(z) = \mp \tilde{h}(-z) \quad (3.49)$$

Com base nas expressões 3.48 e 3.49 verificam-se duas possibilidades para a equação 3.47. Escolhendo a versão imposta pelos sinais indicados superiormente, pode-se reescrever 3.47:

$$x^*(z) = \frac{1}{2} x(z) [\tilde{h}(z) \cdot \tilde{g}(-z) - \tilde{g}(z) \cdot \tilde{h}(-z)] \quad (3.50)$$

Para que exista reconstrução perfeita, a sequência reconstruída apenas pode ser uma réplica atrasada da sequência original, isto é, $x^*(z) = x(z)z^{-m}$. Assim, existe reconstrução perfeita se os filtros de análise e de síntese verificarem as condições de reconstrução perfeita e de não sobreposição espectral, respectivamente:

$$\tilde{h}(z) \cdot h(z) + g(z) \cdot \tilde{g}(z) = 2z^{-m} \quad (3.51)$$

$$h(z) \cdot \tilde{h}(-z) + g(z) \cdot \tilde{g}(-z) = 0 \quad (3.52)$$

Para tornar os filtros de análise causais é necessário introduzir-lhes um atraso, z^{-1} ; mas por outro lado, como as bases são simétricas, pode-se ignorar o atraso z^{-m} . Modificando o sistema da figura 3.4 de acordo com estes pressupostos, obtém-se o sistema da figura 3.5. A condição necessária e suficiente para que a reconstrução seja perfeita é agora definida por:

$$\tilde{h}(z^{-1}) \cdot h(z) + g(z) \cdot \tilde{g}(z^{-1}) = 2 \quad (3.53)$$

$$h(z) \cdot \tilde{h}(-z^{-1}) + g(z) \cdot \tilde{g}(-z^{-1}) = 0 \quad (3.54)$$

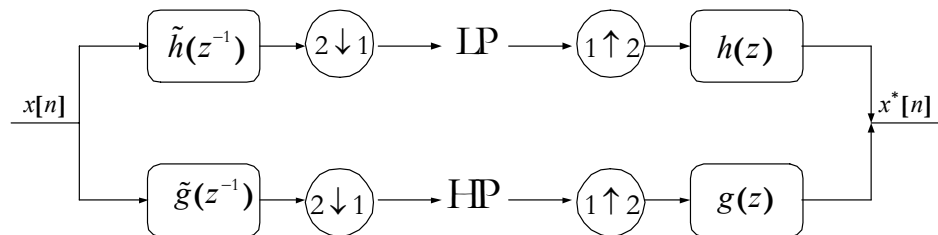


Figura 3.5: Esquema de bancos de filtros causais, para decomposição e reconstrução.

Os filtros de análise e de síntese são filtros em quadratura conjugada (CQF)¹⁵[Daub_92]. Uma vez escolhidos os filtros do bloco de análise, ou do bloco de síntese, é possível obter de imediato os filtros do outro bloco.

Uma característica importante das bases ortonormais associadas a um esquema de MRA, é o facto de originarem pares de filtros CQF, que são filtros FIR com a mesma dimensão. A realização de bancos de filtros FIR de fase linear é sempre desejável, por um lado pela sua simplicidade, e por outro lado porque minimizam os efeitos da distorção de fase.

3.4.2 Filtros de Ôndulas de Daubechies

Os filtros de ôndulas mais simples são os filtros de ôndulas de Daubechies, DB. É também comum referir os filtros de ôndulas de Daubechies pelo número de coeficientes do filtro, como por exemplo, DB1 \equiv D2, DB2 \equiv D4, etc. Os filtros de ôndulas de Daubechies são filtros FIR com fase quase linear, e caracterizam-se por ter suporte compacto e fraca regularidade. Contudo a regularidade aumenta com o aumento do número de coeficientes dos filtros associados, o que se reflecte directamente na linearidade da fase dos filtros. O filtro mais simples é o filtro de Haar, também referido por DB1, ou D2. Porém, ao contrário de todos os outros, este filtro não tem derivadas contínuas. Os coeficientes dos filtros associados às ôndulas ortogonais de Daubechies, e as suas propriedades são referidas no apêndice A.

3.4.3 Filtros de Ôndulas Biortogonais

Com ôndulas ortogonais não é possível obter reconstrução perfeita usando as mesmas funções para análise e para síntese. Além disso, os filtros associados às ortogonais não apresentam característica de fase linear, introduzindo, portanto, distorção de fase.

Na decomposição biortogonal, as funções de análise e de síntese garantem reconstrução perfeita e são simétricas, isto é, os filtros associados têm fase linear. Usando esta propriedade é possível determinar os filtros associados as ôndulas biortogonais fazendo, $\phi = \tilde{\phi}$ e $\psi = \tilde{\psi}$. No entanto, por vezes tem mais interesse o comportamento de uma função na frequência, do que

¹⁵ As condições dos coeficientes dos filtros foram obtidas por generalização dos filtros QMF.

a própria função. No caso das funções de base, em particular para a função de escala, a transformada de Fourier (contínua) é dada por:

$$\hat{\phi}(\omega) = \int_{-\infty}^{\infty} \phi(x) \cdot e^{-j\omega x} \cdot dx, \quad \omega \in \mathbb{R} \quad (3.55)$$

Atendendo ao requisito da área unitária, dado pela equação 3.15, e aplicando a transformada de Fourier a ambos os termos da equação 3.25 obtém-se:

$$\begin{aligned} \hat{\phi}(\omega) &= \sqrt{2} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} \phi(2x - k) \cdot e^{-j\omega x} \cdot dx \\ \hat{\phi}(\omega) &= \sqrt{2} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} \phi(y) \cdot e^{-j\omega \frac{(y+k)}{2}} \cdot \frac{dy}{2} \quad y = 2x - k \\ \hat{\phi}(\omega) &= \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} a_k \cdot e^{-j\frac{\omega}{2}k} \int_{-\infty}^{\infty} \phi(y) \cdot e^{-j\frac{\omega}{2}y} \cdot dy \\ \hat{\phi}(\omega) &= A\left(\frac{\omega}{2}\right) \cdot \hat{\phi}\left(\frac{\omega}{2}\right) \end{aligned} \quad (3.56)$$

em que, $A(\omega)$ é uma função de período, 2π , dada por:

$$A(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} a_k \cdot e^{-j\omega k}, \quad \omega \in \mathbb{R} \quad (3.57)$$

De modo análogo, para a equação 3.26, obtém-se:

$$\begin{aligned} \hat{\psi}(\omega) &= \sqrt{2} \sum_{k=0}^{D-1} b_k \int_{-\infty}^{\infty} \phi(2x - k) \cdot e^{-j\omega x} \cdot dx \\ \hat{\psi}(\omega) &= B\left(\frac{\omega}{2}\right) \cdot \hat{\phi}\left(\frac{\omega}{2}\right) \end{aligned} \quad (3.58)$$

em que, $B(\omega)$ é definida por:

$$B(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} b_k e^{-j\omega k}, \quad \omega \in \mathbb{R} \quad (3.59)$$

Pela organização encadeada e ortogonal dos espaços V_j e W_j , na MRA, sabe-se que o espaço de detalhe é ortogonal e complementar ao espaço de aproximação em qualquer escala.

$$V_j \cap W_j = \{0\}, \quad s \in \mathbb{Z} \quad (3.60)$$

$$V_j + W_j = V_{j+1} \quad (3.61)$$

Em termos de MRA, as funções $\{\phi_{j,k}\}: j, k \in \mathbb{Z}$ e a sua dual, $\{\tilde{\phi}_{j,k}\}: j, k \in \mathbb{Z}$, constituem bases ortonormais dos sub espaços de $L^2(\mathbb{R})$, V_j e \tilde{V}_j , respectivamente. Por sua vez, as bases $\{\psi_{j,k}\}: j, k \in \mathbb{Z}$ e a sua dual, $\{\tilde{\psi}_{j,k}\}: j, k \in \mathbb{Z}$, são bases ortonormais dos espaços de detalhe, W_j e \tilde{W}_j , complementares a V_j e \tilde{V}_j , respectivamente. Como os espaços de detalhe são ortogonais entre si ao longo das várias escalas, e pelas condições de ortogonalidade obtém-se:

$$V_j \perp \tilde{W}_j \Rightarrow \langle \phi_{j,k}(x), \tilde{\psi}_{j,m}(x) \rangle = 0 \quad (3.62)$$

$$\tilde{V}_j \perp W_j \Rightarrow \langle \tilde{\phi}_{j,k}(x), \psi_{j,m}(x) \rangle = 0 \quad (3.63)$$

$$\langle \phi_{j,k}(x), \tilde{\phi}_{j,k'}(x) \rangle = \delta_{k,k'} \quad (3.64)$$

$$\langle \psi_{j,k}(x), \tilde{\psi}_{j',k'}(x) \rangle = \delta_{j,j'} \cdot \delta_{k,k'} \quad (3.65)$$

Para estas bases, as relações entre escalas são definidas por:

$$\phi_{j,k}(x) = \sum_k h[k] \cdot \phi(2x - k) \quad (3.66)$$

$$\tilde{V}_j \perp W_j \Rightarrow \langle \tilde{\phi}_{j,k}(x), \psi_{j,m}(x) \rangle = 0 \quad (3.67)$$

$$V_j \perp \tilde{W}_j \Rightarrow \langle \phi_{j,k}(x), \tilde{\psi}_{j,m}(x) \rangle = 0 \quad (3.68)$$

$$\tilde{V}_j \perp W_j \Rightarrow \langle \tilde{\phi}_{j,k}(x), \psi_{j,m}(x) \rangle = 0 \quad (3.69)$$

A condição de reconstrução perfeita para uma função m de período 2π em $L^2([0, 2\pi])$, nestas condições, é dada por:

$$m_0(\omega) \cdot \overline{\tilde{m}_0(\omega)} + m_0(\omega + \pi) \cdot \overline{\tilde{m}_0(\omega + \pi)} = 1 \quad (3.70)$$

No caso das ôndulas ortogonais, tem-se:

$$\hat{\phi}(\omega) = m_0\left(\frac{\omega}{2}\right) \cdot \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.71)$$

$$\hat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right) \cdot \hat{\psi}\left(\frac{\omega}{2}\right) \quad (3.72)$$

$$\hat{\phi}(\omega) = \tilde{m}_0\left(\frac{\omega}{2}\right) \cdot \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.73)$$

$$\hat{\psi}(\omega) = \tilde{m}_1\left(\frac{\omega}{2}\right) \cdot \hat{\psi}\left(\frac{\omega}{2}\right) \quad (3.74)$$

em que:

$$m_0(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} h_k \cdot e^{-j\omega k} \quad (3.75)$$

$$m_1(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} g_k \cdot e^{-j\omega k} \quad (3.76)$$

$$\tilde{m}_0(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} \tilde{h}_k \cdot e^{-j\omega k} \quad (3.77)$$

$$\tilde{m}_1(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} \tilde{g}_k \cdot e^{-j\omega k} \quad (3.78)$$

As ôndulas biortogonais são também conhecidas pelo número de momentos nulos da ôndula e da ôndula dual, e apresentam as seguintes propriedades:

- A função de escala $\phi_{j,k}(x)$ é sempre simétrica;
- As funções ôndula $\psi_{j,k}(x)$ são sempre simétricas ou anti-simétricas;
- Os filtros das ôndulas biortogonais são filtros FIR de fase linear.

3.5 Transformada de Ôndulas pelo Esquema Progressivo

A ideia fundamental associada ao esquema progressivo (*lifting scheme*, em Inglês) e que motivou a sua designação, é a construção de blocos simples (triviais) de MRA, evoluindo gradualmente para blocos de MRA mais complexos com determinadas propriedades. No início é usada uma ôndula trivial, isto é, uma função que não provoca quaisquer alterações às

sequências, mas que contém as propriedades das ôndulas. Com base nesta função vai-se construindo uma nova ôndula com outras propriedades.

O esquema progressivo é um método eficiente para cálculo da transformada de ôndulas, que foi usado no trabalho desta tese (ver capítulo 4). Face aos métodos clássicos para cálculo da transformada de ôndulas, o esquema progressivo apresenta as seguintes vantagens:

- é um método genérico de implementação mais fácil;
- permite o cálculo rápido da DWT com ganhos no tempo de processamento (até duas vezes), embora mantenha uma complexidade $O(n)$;
- permite calcular a transformada inversa pela simples inversão dos passos usados no cálculo dos coeficientes da transformada directa, com a mesma complexidade da DWT;
- permite determinar a DWT e a DWT^{-1} sem necessidades adicionais de memória. Na DWT o sinal original é substituído pela sua transformada em cada nível, e vice-versa.

O uso da transformada de ôndulas visa tirar partido da estrutura de correlação presente nos sinais, obtendo-se uma boa aproximação aos sinais. A estrutura de correlação é tipicamente localizada no tempo (espaço) e na frequência.

Considere-se um sinal discreto $x = (x_k)_{k \in \mathbb{Z}}$, $x_k \in \mathbb{R}$, decomposto em dois conjuntos disjuntos: as amostras de ordem par $\{x_e = (x_{2k})_{k \in \mathbb{Z}}\}$ e as amostras de ordem ímpar $\{x_o = (x_{2k+1})_{k \in \mathbb{Z}}\}$. Como estes dois conjuntos estão fortemente correlacionados, é possível definir um operador, P , para predição dos valores do conjunto das amostras ímpares a partir do conjunto das amostras pares. O operador P não tem que ser obrigatoriamente exacto, sendo neste caso necessário guardar a informação da diferença (detalhe). Este processo (predição da amostra e registo do detalhe) constitui o primeiro passo do esquema progressivo:

$$d = x_e - P(x_o) \quad (3.79)$$

A partir do detalhe, d , e da amostra ímpar, é possível recuperar a amostra par invertendo o processo:

$$x_e = P(x_o) + d \quad (3.80)$$

Deste modo obtém-se a transformação de (x_e, x_o) em (x_e, d) . Como x_e é obtido por subamostragem, a separação de frequências é pobre, podendo ocorrer sobreposição espectral. Para que tal não se verifique é proposto um segundo passo, substituindo a amostra par por um valor aproximado e regular, usando um operador de actualização, U , aplicado ao detalhe:

$$a = x_e + U(d) \quad (3.81)$$

Este passo é, também, facilmente invertível,

$$x_e = a - U(d) \quad (3.82)$$

A figura 3.6 ilustra o diagrama de blocos dos dois passos relativos ao esquema progressivo.

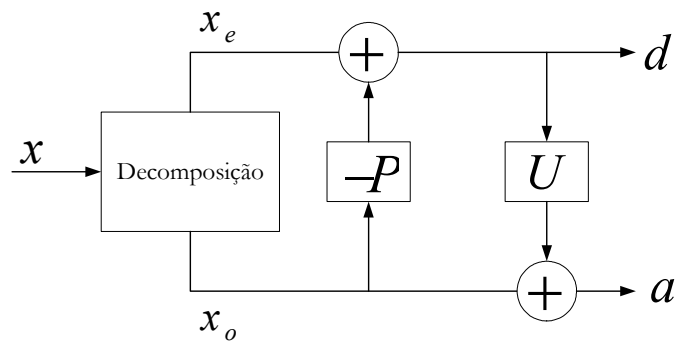


Figura 3.6: Diagrama de blocos do esquema progressivo: predição e actualização.

Qualquer banco de filtros FIR pode ser decomposto segundo o esquema progressivo, através da sua representação por uma matriz polifásica. As condições para reconstrução perfeita podem ser obtidas através de operações com matrizes, cujos elementos correspondem a polinómios de Laurent com coeficientes reais. Os passos relativos ao esquema progressivo são representados por operações sobre matrizes triangulares, superiores ou inferiores, cujos coeficientes da diagonal têm valor unitário.

3.5.1 Filtros e Polinómios de Laurent

A transformada Z de um filtro FIR, $h(z)$, corresponde a um polinómio de Laurent definido por,

$$h(z) = \sum_{k=k_e}^{k_h} h_k z^{-k} \quad (3.83)$$

em que k_e e k_b são o menor e o maior valor inteiro de k , respectivamente, para os quais $h_k \neq 0$. O grau do polinómio de Laurent é dado por $|h| = k_e - k_b$.

A soma algébrica de polinómios de Laurent é ainda um polinómio de Laurent. O produto de dois polinómios de Laurent de grau n e n' , é também um polinómio de Laurent de grau $n + n'$.

Sejam os polinómios $a(z)$ e $b(z) \neq 0$, em que $|a(z)| \geq |b(z)|$. Existe sempre um quociente, $q(z)$, dado por um polinómio de Laurent, em que $|q(z)| = |a(z)| - |b(z)|$, e um polinómio de Laurent correspondente ao resto, $r(z)$, em que $|r(z)| < |b(z)|$, tal que,

$$a(z) = b(z) \cdot q(z) + r(z)$$

Ou ainda,

$$q(z) = a(z) / b(z)$$

$$r(z) = a(z) \bmod b(z)$$

Algoritmo Euclidiano para Polinómios de Laurent

O algoritmo Euclidiano foi originalmente desenvolvido para encontrar o maior divisor comum (MDC) entre dois números naturais. Porém, o seu uso pode ser estendido para encontrar o MDC entre dois polinómios, embora neste caso a solução não seja única.

Sejam $a(z)$ e $b(z) \neq 0$ dois polinómios de Laurent, com $|a(z)| \geq |b(z)|$. Fazendo $a_0(z) = a(z)$ e $b_0(z) = b(z)$ e calculando iterativamente as seguintes equações para $j \geq 0$:

$$a_{j+1}(z) = b_j(z)$$

$$b_{j+1}(z) = a_j(z) \bmod b_j(z)$$

obtém-se, $a_n(z) = \text{MDC}(a(z), b(z))$, em que n é o menor número para o qual $b_n(z) = 0$.

O facto da divisão de polinómios de Laurent não ser única pode até ser vantajoso, sendo possível obter sistemas causais [Daub+98].

No esquema progressivo utilizam-se matrizes 2×2 de polinómios de Laurent,

$$\mathbf{M}(z) = \begin{bmatrix} a(z) & b(z) \\ c(z) & d(z) \end{bmatrix}$$

Se o determinante da matriz for um monómio, a matriz é invertível e diz-se unitária se verificar a condição,

$$\mathbf{M}(z)^{-1} = \mathbf{M}(z^{-1})^T$$

No esquema da transformada de ondas discretas por decomposição em sub-bandas, da figura 3.5, definem-se a matriz de modulação, $\mathbf{M}(z)$, e a matriz de modulação dual, $\tilde{\mathbf{M}}(z)$, tais que:

$$\mathbf{M}(z) = \begin{bmatrix} h(z) & h(-z) \\ g(z) & g(-z) \end{bmatrix} \quad (3.84)$$

$$\tilde{\mathbf{M}}(z) = \begin{bmatrix} \tilde{h}(z) & \tilde{h}(-z) \\ \tilde{g}(z) & \tilde{g}(-z) \end{bmatrix} \quad (3.85)$$

A condição de reconstrução perfeita pode então ser definida por

$$\mathbf{M}(z)\tilde{\mathbf{M}}(z^{-1})^T = 2\mathbf{I} \quad (3.86)$$

em que \mathbf{I} é a matriz identidade com dimensão 2×2 . No caso particular da transformada de ondas ortogonais, em que $h = \tilde{h}$ e $g = \tilde{g}$, $\mathbf{M}(z) = \tilde{\mathbf{M}}(z) = \sqrt{2} \cdot \mathbf{I}$.

3.5.2 Representação Polifásica

A representação polifásica de um filtro, $h(z)$, é definida pela equação:

$$h(z) = h_e(z^2) + z^{-1}h_o(z^2) \quad (3.87)$$

em que h_e e h_o correspondem aos coeficientes de ordem par e ímpar, respectivamente, definidos por:

$$h_e(z) = \sum_k h_{2k} z^{-k} \quad (3.88 \text{ a})$$

$$h_o(z) = \sum_k h_{2k+1} z^{-k} \quad (3.88 \text{ b})$$

A representação polifásica é também uma ferramenta útil na representação da transformada de ôndulas através das matrizes polifásicas. A construção da matriz polifásica de síntese faz-se agrupando os coeficientes de ordem par e ímpar dos filtros de síntese,

$$\mathbf{P}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \quad (3.89)$$

tal que,

$$\mathbf{P}(z^2)^T = \frac{1}{2} M(z) \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \quad (3.90)$$

De modo análogo, constrói-se a matriz polifásica dual, $\tilde{\mathbf{P}}(z)$ (usando os filtros de análise), e define-se a condição de reconstrução perfeita,

$$\mathbf{P}(z)\tilde{\mathbf{P}}(z^{-1})^T = \mathbf{I} \quad (3.91)$$

Com base nas matrizes polifásicas pode-se representar a transformada de ôndulas, como ilustra a figura 3.7. É conveniente que as matrizes polifásicas apenas contenham polinómios de Laurent, isto é, o seu determinante apenas pode ser um monómio em z . Sem perda de generalidade, assume-se que $\det \mathbf{P}(z) = 1$. Para um dado filtro h , caso o determinante não seja unitário, é sempre possível escalar e deslocar o filtro de modo a torná-lo unitário. Uma vez encontrada a matriz tal que $\det \mathbf{P}(z) = 1$, obtém-se a matriz $\tilde{\mathbf{P}}(z)$ e os filtros FIR associados à transformada de ôndulas.

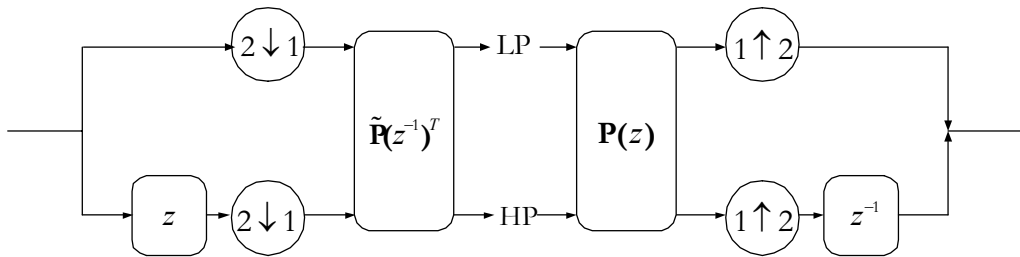


Figura 3.7: Representação polifásica da transformada de ôndulas

Assim, a partir da condição de reconstrução perfeita (eq. 3.91), e aplicando a regra de Cramer,

$$\tilde{h}_e(z) = g_o(z^{-1}), \quad \tilde{h}_o(z) = -g_e(z^{-1}), \quad \tilde{g}_e(z) = -h_o(z^{-1}), \quad \tilde{g}_o(z) = h_e(z^{-1})$$

é possível estabelecer as equações dos filtros,

$$\tilde{g}(z) = z^{-1}h(-z^{-1}) \quad (3.92)$$

$$\tilde{h}(z) = -z^{-1}g(-z^{-1}) \quad (3.93)$$

O exemplo mais simples obtém-se com $\mathbf{P}(z) = \mathbf{I}$. A transformada de ôndula resultante, neste caso, apenas realiza a separação da sequência em amostras de ordem par e ímpar, e é vulgarmente designada por ôndula passiva (*lazy wavelet*, em Inglês).

3.5.3 Factorização Segundo o Esquema Progressivo

O esquema progressivo estabelece uma relação para reconstrução perfeita, entre pares de filtros que tenham um filtro comum (passa-baixo ou passa-alto). Partindo da ôndula passiva, e aplicando gradualmente os passos do esquema progressivo, constrói-se um bloco de análise multiresolução com propriedades específicas. Sabendo que neste esquema de MRA os filtros são complementares¹⁶, usando os teoremas de *lifting* e *dual lifting* [Daub+98], é possível determinar os novos filtros em cada novo estágio da análise multiresolução.

Primal Lifting

Sejam os filtros complementares (h, g) do bloco de síntese. Usando o teorema do *lifting*, qualquer outro filtro, g^{new} , finito e complementar a h , é da forma,

$$g^{new}(z) = g(z) + h(z) \cdot s(z^2)$$

em que $s(z)$ é um polinómio de Laurent. O mesmo se pode aplicar ao bloco de análise, isto é, qualquer novo filtro, \tilde{h}^{new} , finito e complementar a \tilde{g} , é dado por,

$$\tilde{h}^{new}(z) = \tilde{h}(z) - \tilde{g}(z) \cdot s(z^{-2}) \quad \square$$

Como se pode verificar, estas operações não alteram o determinante da matriz polifásica, dado que, $\mathbf{P}^{new}(z) = \mathbf{P}(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$ e $\tilde{\mathbf{P}}^{new}(z) = \tilde{\mathbf{P}}(z) \begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix}$, respectivamente. Os novos filtros assim obtidos têm propriedades idênticas aos que os originaram.

¹⁶ Um par de filtros, (h, g) ou (\tilde{h}, \tilde{g}) , dizem-se complementares, se o determinante da sua matriz polifásica for unitário.

Dual Lifting

Sejam os filtros complementares (h, g) do bloco de síntese. Usando o teorema de *dual lifting*, qualquer outro filtro, h^{new} , finito e complementar a g , é da forma,

$$h^{new}(z) = h(z) + g(z) \cdot t(z^2)$$

em que $t(z)$ é um polinómio de Laurent.

Um novo filtro, \tilde{g}^{new} , finito e complementar a \tilde{h} , dado por,

$$\tilde{g}^{new}(z) = \tilde{g}(z) - \tilde{h}(z) \cdot t(z^{-2}) \quad \square$$

Os teoremas referidos anteriormente correspondem a passos de *primal lifting* e *dual lifting*, respectivamente, no algoritmo de factorização do esquema progressivo.

O algoritmo de factorização usado no esquema progressivo assenta fundamentalmente no algoritmo Euclidiano para factorização de polinómios de Laurent [Daub+98], e nos teoremas de *lifting* e *dual lifting*. Com base nestes, pode-se demonstrar que dado um par de filtros complementares (h, g) , existem sempre polinómios de Laurent, $s_j(z)$ e $t_j(z)$, para $1 \leq j \leq m$ e uma constante $k \neq 0$, que conduzem às seguintes factorização das matrizes polifásicas:

$$\mathbf{P}(z) = \prod_{j=1}^m \begin{bmatrix} 1 & s_j(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_j(z) & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & k^{-1} \end{bmatrix} \quad (3.94)$$

$$\tilde{\mathbf{P}}(z) = \prod_{j=1}^m \begin{bmatrix} 1 & 0 \\ -s_j(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_j(z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k^{-1} & 0 \\ 0 & k \end{bmatrix} \quad (3.95)$$

Ou seja, qualquer transformada de ôndulas associada a filtros finitos, pode ser obtida através da ôndula passiva, seguida de m passos de *primal lifting* e de *dual lifting*, afectada de uma constante de escala. No caso das ôndulas ortogonais, em que $\mathbf{P}(z) = \tilde{\mathbf{P}}(z)$, obtêm-se duas factorizações diferentes. As figuras 3.8 e 3.9 representam o esquema dos vários passos envolvidos para obtenção da transformada de ôndulas directa e inversa, respectivamente.

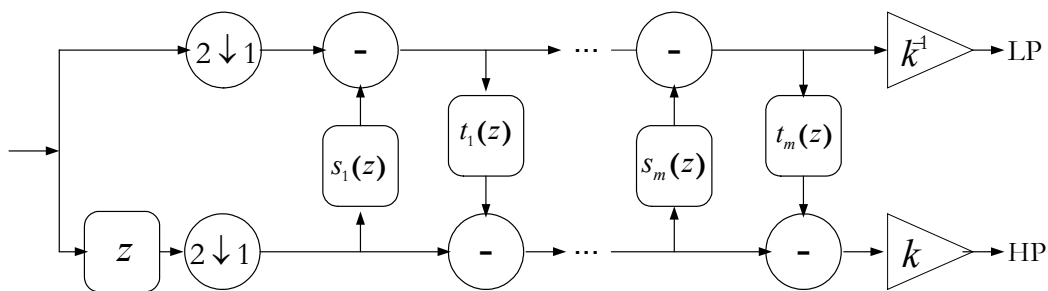


Figura 3.8: Cálculo da transformada de ondas directa pelo esquema progressivo.

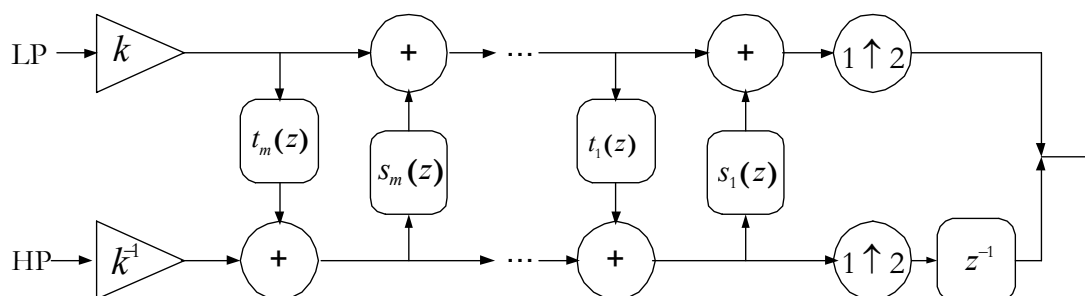


Figura 3.9: Cálculo da transformada de ondas inversa pelo esquema progressivo

3.6 Transformada de Ondulas Bidimensional

A transformada de ondas unidimensional pode ser expandida a duas dimensões, com base no produto de funções separáveis. Além de simples, este método tem ainda como vantagem o facto do produto de funções ortogonais gerar também uma função ortogonal.

No caso de um sinal bidimensional (2D), a representação das componentes do sinal é feita através de ondas e uma função de escala, todas bidimensionais. Por cada par função de escala e onda correspondente, constroem-se três ondas 2D, assim definidas:

$$\Psi^1_{j,k}(x, y) = \psi(x - j) \cdot \phi(y - k) \quad (3.96)$$

$$\Psi^2_{j,k}(x, y) = \phi(x - j) \cdot \psi(y - k) \quad (3.97)$$

$$\Psi^3_{j,k}(x, y) = \psi(x - j) \cdot \psi(y - k) \quad (3.98)$$

e uma função de escala 2D, definida por,

$$\Phi_{j,k}(x, y) = \phi(x - j) \cdot \phi(y - k) \quad (3.99)$$

As funções definidas pelas equações 3.96 a 3.99 são ôndulas, dado que verificam a condição,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi_{j,k}^i(x,y) \cdot dx \cdot dy = 0 \quad i = 1, 2, 3 \quad (3.100)$$

A análise multiresolução, anteriormente definida, pode também ser generalizada a duas dimensões, e difere da situação a 1D por usar três ôndulas e uma função de escala em cada nível de resolução. A transformada de ôndulas discreta bidimensional (DWT 2D), resulta da extensão a 2D do algoritmo 1D aplicado sequências separáveis [Dudg+84], primeiro segundo a direcção das linhas e depois segundo a direcção das colunas.

Considere-se por exemplo uma sequência bidimensional, $c^j(m,n)$, resultante de uma matriz de dimensão $M \times N$ à qual se aplica o algoritmo de decomposição, inicialmente segundo a direcção x (linhas) e posteriormente segundo a direcção y (colunas). Devido à operação de decimação, na decomposição segundo as linhas, resultam duas matrizes de dimensão $M \times \frac{N}{2}$.

Do mesmo modo, após a decomposição segundo as colunas resultam quatro matrizes de dimensão $\frac{M}{2} \times \frac{N}{2}$, e as sequências $a^{j-1}(m,n)$, $d_1^{j-1}(m,n)$, $d_2^{j-1}(m,n)$ e $d_3^{j-1}(m,n)$. A sequência $a^{j-1}(m,n)$ é referente à função de escala e contém a informação relativa às baixas frequências; as sequências d_1^{j-1} , d_2^{j-1} e d_3^{j-1} são referentes às três ôndulas consideradas e contém informação relativa às altas-frequências. O procedimento pode ser aplicado recursivamente à matriz $a^l(m,n)$ (componentes de baixa frequência), formando um bloco de MRA com várias escalas, ou níveis de detalhe. A reconstrução do sinal original faz-se pela inversão deste procedimento.

A DWT 2D gera um esquema hierárquico de decomposição e reconstrução, como se representa na figura 3.10. A forma mais comum de a representar é sob a forma de uma matriz com a mesma dimensão da matriz original, como ilustra a figura 3.11 (3 níveis de decomposição).

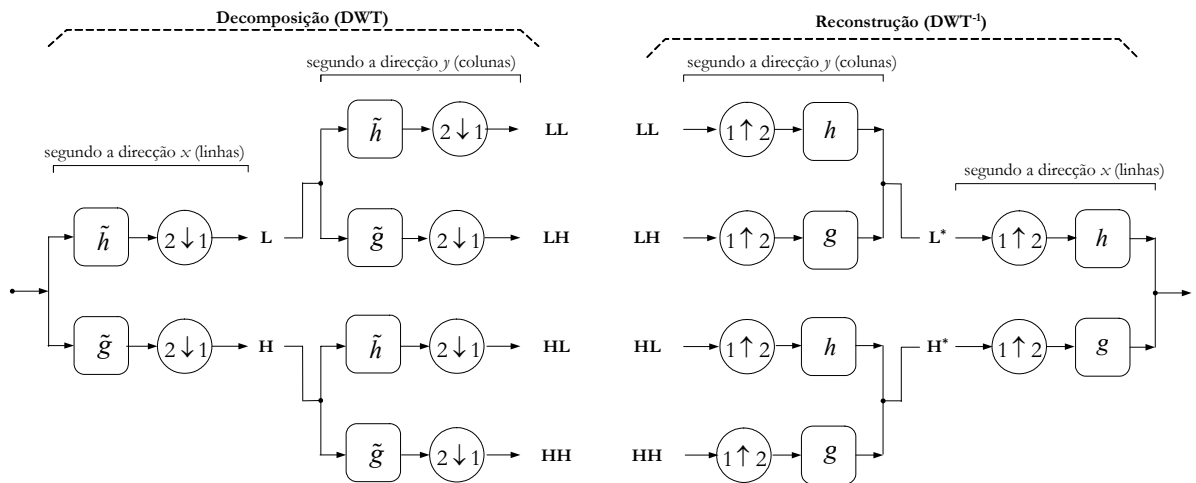


Figura 3.10: Esquema de decomposição em sub-bandas e reconstrução, para sinais bidimensionais, por aplicação da DWT e DWT^{-1} 2-D.

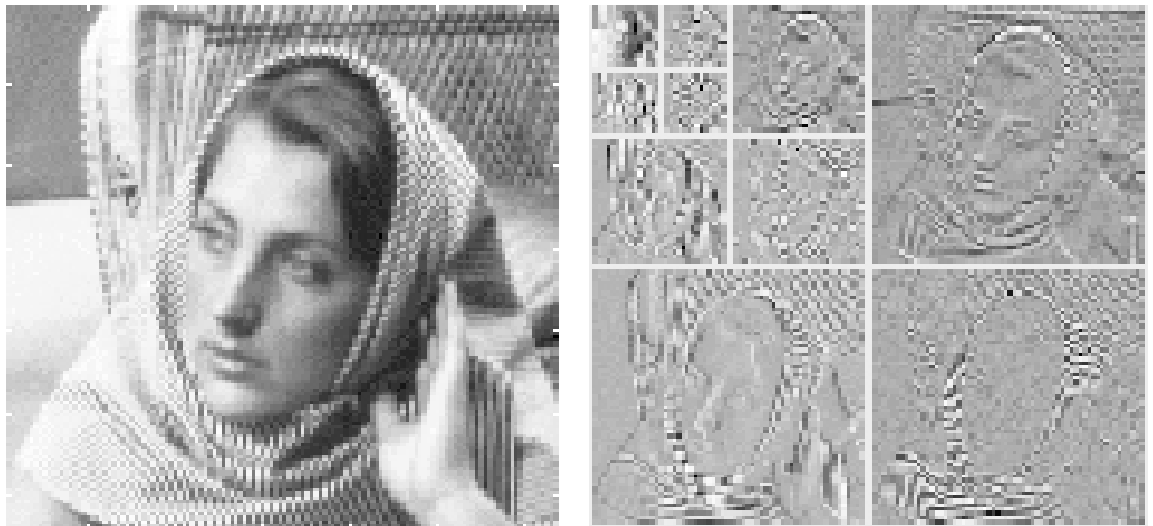


Figura 3.11: Transformada de ondas de uma imagem com decomposição em 3 níveis.

3.7 Conclusões

Neste capítulo foram referidos os aspectos fundamentais associados às ondas e às transformadas de ondas, e a sua relação com os esquemas de análise (decomposição) e de síntese (reconstrução) de sinais em bancos de filtros.

Através de esquemas de decomposição em sub-bandas, com base nos filtros associados às ondas, é possível implementar a transformada de ondas e a transformada de ondas

inversa. É ainda possível escolher os filtros de ôndulas, de acordo com critérios de regularidade e melhoria da concentração de energia. O uso de filtros associados às ôndulas ortogonais, ou biortogonais, tem vantagens face aos filtros QMF. Por um lado, as ôndulas ortogonais de Daubechies e as ôndulas biortogonais conduzem sempre à realização de filtros FIR, que podem apresentar característica de fase linear e, por outro lado, permitem reconstrução perfeita. Os filtros FIR de fase linear são desejáveis em processamento de imagem e vídeo, porque minimizam os efeitos de distorção de fase. Além disso, a transformada de ôndulas associada a este tipo filtros pode ser factorizada através do esquema progressivo.

O cálculo da transformada pelo esquema progressivo permite a redução no custo computacional por duas razões fundamentais: por um lado porque permite realizar operações recursivamente; por outro lado porque a factorização das matrizes polifásicas origina a decomposição triangular (superior e inferior) das matrizes intermédias, com os restantes elementos nulos.

A transformada de ôndulas 2D conduz a um esquema de decomposição piramidal, em sub-bandas de frequência diferentes, descorrelacionadas entre si. Esta característica permite, por exemplo, a realização de operações de filtragem local, mantendo as sub-bandas de interesse e eliminando as restantes. Por outro lado, permite realizar esquemas eficientes de codificação baseados nos coeficientes da transformada, como se apresenta no capítulo 5.

Capítulo 4

IMPLEMENTAÇÃO DA DWT E DWT⁻¹ PELO ESQUEMA PROGRESSIVO.

Neste capítulo apresentam-se os algoritmos e descreve-se a implementação da DWT e da DWT⁻¹ pelo esquema progressivo, para a ôndula ortogonal Daubechies 2 e para as ôndulas biortogonais (4.4), usadas no CoDec de vídeo. Comparam-se os tempos de cálculo das transformadas num processador de uso geral (computador pessoal com processador Intel Pentium III a 550 MHz) e num processador digital de sinal com arquitectura VLIW (TMS320C6201 da Texas Instruments).

4.1 Introdução

No capítulo anterior, em 3.5.3, foram referidos os procedimentos gerais para a factorização das matrizes polifásicas que representam a DWT pelo esquema progressivo. O esquema progressivo permite acelerar o cálculo dos coeficientes da transformada, e da transformada inversa, sem necessidade de recurso a memória adicional. No CoDec de vídeo desenvolvido nesta dissertação (descrito no capítulo 5), usam-se as transformadas de ôndulas no domínio do tempo e do espaço, ambas calculadas com base no esquema progressivo. Para a transformada no tempo usa-se a ôndula ortogonal Daubechies 2, e para a transformada no espaço usam-se as ôndulas biortogonais Daubechies 9/7, ou em alternativa a ôndula ortogonal Daubechies 2.

Para avaliar a possibilidade de acelerar o cálculo das transformadas com base em processadores digitais de sinal (DSPs), programou-se um DSP da última geração para o cálculo das transformadas de ôndulas aplicadas no CoDec de vídeo.

Os DSPs têm, em geral, uma arquitectura Harvard [Ifch+93, Pirsh_99, DSP_Hbk], dispondo de barramentos separados para dados e programa, e de circuitos para acelerar a realização das operações aritméticas. O DSP usado nesta dissertação tem uma arquitectura VLIW, executando até 8 instruções por ciclo de relógio, agendadas estaticamente pelo compilador. Os diferentes campos da palavra de instrução contêm as instruções individuais para cada unidade funcional: ALU, multiplicador, registo de deslocamento, etc. A melhoria no desempenho obtida com os processadores VLIW, depende do grau de paralelismo exibido, e do número de unidades funcionais disponíveis. Neste trabalho procedeu-se ao agendamento das instruções necessárias para o cálculo das transformadas de ôndulas, de forma a garantir uma elevada taxa de ocupação das unidades funcionais.

4.2 DWT e DWT⁻¹ para a Ôndula Ortogonal Daubechies 2

Em [Daub+98] os filtros de síntese relativos à ôndula DB2 são definidos pelos seguintes polinómios de Laurent:

$$h(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} \quad (4.1)$$

$$g(z) = -h_3 z^2 + h_2 z^1 - h_1 z + h_0 z^{-1} \quad (4.2)$$

Os coeficientes dos filtros (ver apêndice A) definidos em [Daub_92], na tabela 6.1 (pp. 195), são:

$$\begin{aligned} h_0 &= \frac{1 + \sqrt{3}}{4\sqrt{2}} \approx 0,4829 & h_1 &= \frac{3 + \sqrt{3}}{4\sqrt{2}} \approx 0,8365 \\ h_2 &= \frac{3 - \sqrt{3}}{4\sqrt{2}} \approx 0,2241 & h_3 &= \frac{1 - \sqrt{3}}{4\sqrt{2}} \approx -0,1294 \end{aligned}$$

Define-se a matriz polifásica de síntese:

$$\mathbf{P}(z) = \tilde{\mathbf{P}}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} h_0 + h_2 z^{-1} & -h_3 z^1 - h_1 \\ h_1 + h_3 z^{-1} & h_2 z^1 + h_0 \end{bmatrix} \quad (4.3)$$

e a matriz polifásica de análise, tal que:

$$\mathbf{P}(z)^{-1} = \tilde{\mathbf{P}}(z^{-1}) \quad (4.4)$$

Factorizando a matriz polifásica da expressão (4.3) obtém-se,

$$\mathbf{P}(z) = \tilde{\mathbf{P}}(z) = \begin{bmatrix} 1 & -\sqrt{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z^{-1} & 0 \\ \frac{\sqrt{3}}{4} & 1 \end{bmatrix} \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \quad (4.5)$$

Como o determinante da matriz polifásica é unitário (ver secção 3.5.2), pode-se usar (4.5) para factorizar a matriz polifásica de síntese, e também a matriz polifásica de análise. Para a matriz polifásica de análise tem-se:

$$\tilde{\mathbf{P}}(z^{-1})^T = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix} \quad (4.6)$$

Pelo esquema progressivo, os coeficientes da DWT relativa à ôndula DB2 são obtidos a partir da sequência de valores de entrada ($x = \{x_l, l \in \mathbb{Z}\}$), aplicando à sequência a ôndula passiva (*lazy wavelet*). A ôndula passiva efectua a separação da sequência original em duas sequências com índices pares e ímpares, (x_{2l} e x_{2l+1}), para cálculo dos coeficientes de aproximação (a) e de detalhe (d). Os coeficientes de aproximação e de detalhe obtêm-se por aplicação de filtros passa-baixo e passa-alto, h e g , respectivamente, a cada par de valores da sequência a transformar, através das seguintes expressões:

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix} \begin{bmatrix} x_{2l} \\ x_{2l+1} \end{bmatrix}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2l} & x_{2l+1} \\ x_{2l+1} & -\sqrt{3}x_{2l} \end{bmatrix}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} x_{2l} + \left(\frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z\right) d_l^{(1)} \\ d_l^{(1)} \end{bmatrix} \quad d_l^{(1)} = x_{2l+1} - \sqrt{3}x_{2l}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a_l^{(1)} \\ \left[x_{2l} + \left(\frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z\right) d_l^{(1)} \right] z^{-1} + d_l^{(1)} \end{bmatrix} \quad a_l^{(1)} = x_{2l} + \left(\frac{\sqrt{3}}{4} + \frac{\sqrt{3-2}}{4} z\right) d_l^{(1)}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3+1}}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3-1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a_l^1 \\ d_l^2 \end{bmatrix} \quad d_l^{(2)} = a_l^{(1)} z^{-1} + d_l^{(1)}$$

Em que $a_l^{(i)}$ e $d_l^{(i)}$ representam os valores dos coeficientes de aproximação e de detalhe, na iteração l , e no passo intermédio i .

O algoritmo para a implementação da DWT segundo o esquema progressivo compreende os seguintes cálculos:

$$\begin{aligned}
 d_l^{(1)} &= x_{2l+1} - \sqrt{3} \cdot x_{2l} & d_l^{(1)} &\approx x_{2l+1} - 1,732 \cdot x_{2l} \\
 a_l^{(1)} &= x_{2l} + \frac{\sqrt{3}}{4} \cdot d_l^{(1)} + \frac{\sqrt{3}-2}{4} \cdot d_{l+1}^{(1)} & a_l^{(1)} &\approx x_{2l} + 0,433 \cdot d_l^{(1)} - 0,067 \cdot d_{l+1}^{(1)} \\
 d_l^{(2)} &= d_l^{(1)} + a_{l-1}^{(1)} & d_l^{(2)} &\approx d_l^{(1)} + a_{l-1}^{(1)} \\
 a_l &= \frac{\sqrt{3}+1}{\sqrt{2}} \cdot a_l^{(1)} & a_l &\approx 1,9318 \cdot a_l^{(1)} \\
 d_l &= \frac{\sqrt{3}-1}{\sqrt{2}} \cdot d_l^{(2)} & d_l &\approx 0,5176 \cdot d_l^{(2)}
 \end{aligned}$$

A DWT^{-1} correspondente à ôndula de Daubechies 2 obtém-se simplesmente invertendo os passos seguidos na DWT e trocando os sinais:

$$\begin{aligned}
 d_l^{(2)} &= \frac{\sqrt{3}+1}{\sqrt{2}} \cdot d_l & d_l^{(2)} &\approx 1,9318 \cdot d_l \\
 a_l^{(1)} &= \frac{\sqrt{3}-1}{\sqrt{2}} \cdot a_l & a_l^{(1)} &\approx 0,5176 \cdot a_l \\
 d_l^{(1)} &= d_l^{(2)} - a_{l-1}^{(1)} & d_l^{(1)} &\approx d_l^{(2)} - a_{l-1}^{(1)} \\
 x_{2l} &= a_l^{(1)} - \frac{\sqrt{3}}{4} \cdot d_l^{(1)} - \frac{\sqrt{3}-2}{4} \cdot d_{l+1}^{(1)} & x_{2l} &\approx a_l^{(1)} - 0,433 \cdot d_l^{(1)} + 0,067 \cdot d_{l+1}^{(1)} \\
 x_{2l+1} &= d_l^{(1)} + \sqrt{3} \cdot x_{2l} & x_{2l+1} &\approx d_l^{(1)} + 1,732 \cdot x_{2l}
 \end{aligned}$$

4.3 DWT e DWT⁻¹ para Ôndulas Biortogonais (4,4)

Para as ôndulas biortogonais (4,4), ou par de filtros 9/7, os filtros passa-alto de análise e de síntese, \tilde{g} e g , apresentam 4 momentos nulos cada um. Os filtros 9/7 apresentam simetria em torno do ponto $\frac{N-1}{2}$ ($N=9$), apresentando, portanto, característica de fase linear.

A factorização pode ser feita a partir do filtro de análise:

$$\tilde{h}(z) = h_4 z^{-2} + h_3 z^{-1} + h_2 z^{-1} + h_1 + h_0 + h_1 z + h_2 z + h_3 z^2 + h_4 z^2 \quad (4.7)$$

Agrupando os coeficientes pares (e) e ímpares (o) do filtro, com vista à sua representação polifásica, tem-se:

$$\tilde{h}_e(z) = h_4(z^2 + z^{-2}) + h_2(z + z^{-1}) + h_0 \quad (4.8)$$

$$\tilde{h}_o(z) = h_3(z^{-1} + z^2) + h_1(1 + z) \quad (4.9)$$

Os coeficientes dos filtros das ôndulas biortogonais de Daubechies (ver apêndice A) são obtidos em [Daub_92], na tabela 8.3 (pp. 279), afectados do factor $\sqrt{2}$ (coeficiente de conservação da área, referido em 3.3.1):

$$h_0 \approx \sqrt{2} \times 0,6029 \approx 0,8527 \quad h_1 \approx \sqrt{2} \times 0,2668 \approx 0,3774 \quad h_2 \approx \sqrt{2} \times (-0,0782) \approx -0,1106$$

$$h_3 \approx \sqrt{2} \times (-0,0168) \approx -0,0238 \quad h_4 \approx \sqrt{2} \times 0,2675 \approx 0,0378$$

Pelo algoritmo de factorização dos polinómios de Laurent [Daub+98], tem-se:

$$\begin{aligned} r_0 &= h_0 - \frac{2h_4 h_1}{h_3} & r_1 &= h_2 - h_4 - \frac{h_4 h_1}{h_3} \\ s_0 &= h_1 - h_3 - \frac{h_3 r_0}{r_1} & t_0 &= r_0 - 2r_1 \end{aligned}$$

Donde se obtém:

$$\begin{aligned} \alpha &= \frac{h_4}{h_3} \approx -1,5861 & \beta &= \frac{h_3}{r_1} \approx -0,0529 & \gamma &= \frac{r_1}{s_0} \approx 0,8829 \\ \delta &= \frac{s_0}{t_0} \approx 0,4435 & \varphi &= r_0 - 2r_1 \approx 1,1496 \end{aligned}$$

Uma possível fatorização da matriz polifásica conduz às seguintes expressões:

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \quad (4.10)$$

$$\tilde{\mathbf{P}}(z^{-1})^T = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha(1+z) & 1 \end{bmatrix} \quad (4.11)$$

Pela factorização definida em (4.11) os coeficientes da DWT relativos às ôndulas biortogonais (4.4), são obtidos a partir de uma sequência de valores $x = \{x_l, l \in \mathbb{Z}\}$. Tal como no caso anterior, aplicando à sequência a ôndula passiva obtém-se a separação das amostras de índices pares e ímpares, x_{2l} e x_{2l+1} , para obtenção dos coeficientes de aproximação e de detalhe (a e d), através das seguintes expressões:

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha(1+z) & 1 \end{bmatrix} \begin{bmatrix} x_{2l} \\ x_{2l+1} \end{bmatrix}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2l} \\ \alpha(1+z)x_{2l} + x_{2l+1} \end{bmatrix}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} x_{2l} + \beta(1+z^{-1})d_l^{(1)} \\ d_l^{(1)} \end{bmatrix} \quad d_l^{(1)} = \alpha(1+z)x_{2l} + x_{2l+1}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_l^{(1)} \\ \gamma(1+z)a_l^{(1)} + d_l^{(1)} \end{bmatrix} \quad a_l^{(1)} = x_{2l} + \beta(1+z^{-1})d_l^{(1)}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \begin{bmatrix} a_l^{(1)} + \delta(1+z^{-1})d_l^{(2)} \\ d_l^{(2)} \end{bmatrix} \quad d_l^{(2)} = \gamma(1+z)a_l^{(1)} + d_l^{(1)}$$

$$\begin{bmatrix} a_l \\ d_l \end{bmatrix} = \begin{bmatrix} \varphi \cdot a_l^{(2)} \\ 1/\varphi \cdot d_l^{(2)} \end{bmatrix} \quad a_l^{(2)} = a_l^{(1)} + \delta(1+z^{-1})d_l^{(2)}$$

A implementação da DWT corresponde à realização dos seguintes cálculos:

$$a_l^{(0)} = x_{2l}$$

$$d_l^{(0)} = x_{2l+1}$$

$$\begin{aligned}
d_l^{(1)} &= d_l^{(0)} + \alpha \cdot (a_l^{(0)} + a_{l+1}^{(0)}) & d_l^{(1)} &\approx d_l^{(0)} - 1,5861 \cdot (a_l^{(0)} + a_{l+1}^{(0)}) \\
a_l^{(1)} &= a_l^{(0)} + \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) & a_l^{(1)} &\approx a_l^{(0)} - 0,0529 \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \\
d_l^{(2)} &= d_l^{(1)} + \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) & d_l^{(2)} &\approx d_l^{(1)} + 0,8829 \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \\
a_l^{(2)} &= a_l^{(1)} + \delta \cdot (d_l^{(2)} + d_{l-1}^{(2)}) & a_l^{(2)} &\approx a_l^{(1)} + 0,4435 \cdot (d_l^{(2)} + d_{l-1}^{(2)}) \\
a_l &= \varphi \cdot a_l^{(2)} & a_l &\approx 1,1496 \cdot a_l^{(2)} \\
d_l &= \frac{1}{\varphi} \cdot d_l^{(2)} & d_l &\approx 0,8698 \cdot d_l^{(2)}
\end{aligned}$$

A DWT^{-1} relativa às ondas biortogonais (4.4) pode ser obtida seguindo os cálculos efectuados na DWT , mas pela ordem inversa e trocando os sinais:

$$\begin{aligned}
d_l^{(2)} &= \varphi \cdot d_l & d_l^{(2)} &\approx 1,1496 \cdot d_l \\
a_l^{(2)} &= \frac{1}{\varphi} \cdot a_l & a_l^{(2)} &\approx 0,8698 \cdot a_l \\
a_l^{(1)} &= a_l^{(2)} - \delta \cdot (d_l^{(2)} + d_{l-1}^{(2)}) & a_l^{(1)} &\approx a_l^{(2)} - 0,4435 \cdot (d_l^{(2)} + d_{l-1}^{(2)}) \\
d_l^{(1)} &= d_l^{(2)} - \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) & d_l^{(1)} &\approx d_l^{(2)} - 0,8829 \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \\
a_l^{(0)} &= a_l^{(1)} - \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) & a_l^{(0)} &\approx a_l^{(1)} + 0,0529 \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \\
d_l^{(0)} &= d_l^{(1)} - \alpha \cdot (a_l^{(0)} + a_{l+1}^{(0)}) & d_l^{(0)} &\approx d_l^{(1)} + 1,5861 \cdot (a_l^{(0)} + a_{l+1}^{(0)}) \\
x_{2l} &= a_l^{(0)} \\
x_{2l+1} &= d_l^{(0)}
\end{aligned}$$

4.4 Implementação das Transformadas de Ôndulas num Processador de Uso Geral

No CoDec de vídeo realizado usam-se transformadas de ôndulas, nos domínios do tempo e do espaço: a DWT/DWT^{-1} relativa à ôndula ortogonal Daubechies 2 no domínio do tempo e no domínio do espaço, a DWT/DWT^{-1} relativa às ôndulas biortogonais (4.4), ou em alternativa a ôndula Daubechies 2. As sequências de vídeo usadas têm resolução espacial no formato QCIF (144×176) e sub amostragem espacial (4:2:0). O processamento faz-se em grupos de quatro imagens consecutivas (GOF – *Group of Frames*), sendo cada uma das quatro imagens representada por três matrizes, uma para a componente de luminância e as restantes duas matrizes para as componentes de crominância. Os valores dos *pixels* são representados por 8 *bits*, com valores que variam entre 0 e 255. Primeiro aplica-se a transformada no domínio do tempo, com decomposição em dois níveis, aos grupos das quatro matrizes representando cada uma das componentes; em seguida aplica-se a transformada no espaço, com três níveis de decomposição, a cada uma 12 matrizes resultantes da transformação no tempo, isto é, 4 matrizes por cada uma das componentes. Devido às características do esquema progressivo, não é necessário espaço de memória adicional, isto é, as matrizes resultantes das transformadas ocupam o espaço inicialmente ocupado pelas matrizes originais. Assim, atendendo aos valores máximos possíveis, os coeficientes da sequência são representados por inteiros de 16 *bits* (tipo **short**). Os coeficientes obtidos pela factorização são valores reais, que na implementação em processador de uso geral são representados por números em vírgula flutuante. Visando minimizar os erros, os resultados de todas as operações intermédias são igualmente números reais em vírgula flutuante (tipo **float**). Em cada iteração, os valores finais dos coeficientes de aproximação e de detalhe são armazenados, representados por números inteiros de 16 *bits* (tipo **short**). Porém, no esquema progressivo, caso não existiam cuidados no arredondamento dos coeficientes [Cald+98], a representação dos coeficientes em números inteiros, em vez de números reais, leva à não obtenção de reconstrução perfeita, devido ao erro introduzido pela conversão para inteiros. O erro devido ao arredondamento é tanto maior quanto maior for o número de níveis de decomposição. Visando minimizar os erros (diferenças) entre os valores dos coeficientes das matrizes originais e das matrizes recuperadas, na DWT os coeficientes são arredondados por defeito, enquanto na DWT^{-1} são arredondados por excesso.

Dadas as características recursivas do algoritmo de cálculo das transformadas segundo o esquema progressivo, e os seus requisitos de memória, é comum consideram-se nulos os

coeficientes relativos às regiões limite, isto é, nas condições iniciais e futuras, nos limites iniciais e finais da sequência, respectivamente.

Ao se considerarem nulos os coeficientes nas regiões fronteira, provoca-se uma degradação das imagens obtidas pelo cálculo da transformada inversa, nas regiões limite, que se propaga tanto mais quanto maior for o número de níveis de aplicação da transformada. De modo a eliminar este efeito é comum considerar a extensão das sequências nas regiões limite. Os métodos para extensão das regiões limites das sequências são referidos em [Brisl_96]. No presente trabalho consideram-se duas situações na extensão das sequências de trabalho, para o cálculo das transformadas inversas: a duplicação dos valores limite das sequências, com espelhamento; e o cálculo dos coeficientes da extensão a partir dos coeficientes das regiões limite, através de um processo de cálculo dpor regressão. Este processo de regressão, que se descreve a seguir de forma sucinta e com maior detalhe no apêndice B, é muito fácil para a ôndula Daubechies 2, e por essa razão apenas nesse caso é usado.

De modo a garantir reconstrução perfeita, na DWT^{-1} pelo esquema progressivo convém considerar os coeficientes obtidos na DWT para os casos limite, e que foram desprezados na sequência resultante. Para o caso da ôndula DB2, este facto não constitui um problema, uma vez que os coeficientes da extensão da sequência podem ser obtidos a partir dos valores da sequência final por regressão. Com base na sequência transformada e nos passos para o cálculo da DWT descritos na secção 4.2, é possível estabelecer uma relação entre os coeficientes da sequência transformada de forma a obter os coeficientes de aproximação e de detalhe para as regiões limite, no cálculo da DWT^{-1} , sendo:

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} \times a_{l-1}^{(1)}, \text{ em que } a_{l-1}^{(1)} = d_l^{(2)} - d_l^{(1)}$$

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} (d_l^{(2)} - d_l^{(1)}) = \frac{\sqrt{3}+1}{\sqrt{2}} \left(d_l^{(2)} - \frac{4 \cdot d_l^{(2)}}{4 + \sqrt{3} - 2} \right) = \frac{\sqrt{3}+1}{\sqrt{2}} \cdot d_l^{(2)} \cdot \left(1 - \frac{4}{4 + \sqrt{3} - 2} \right)$$

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} \cdot \frac{\sqrt{2}}{\sqrt{3}-1} \cdot d_l \cdot \left(1 - \frac{4}{4 + \sqrt{3} - 2} \right) = -0,2676 \times d_l$$

$$d_{l+1} = \frac{\sqrt{3}-1}{\sqrt{2}} \times d_{l+1}^{(2)}, \text{ em que } d_{l+1}^{(2)} = a_l^{(1)} + d_{l+1}^{(1)}$$

$$d_{l+1} = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot (a_l^{(1)} + d_{l+1}^{(1)}) = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot a_l^{(1)} = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot \frac{\sqrt{2}}{\sqrt{3}+1} \cdot a_l = 0,2679 \cdot a_l$$

Para as ôndulas biortogonais (4,4) adoptou-se a extensão simétrica da sequência de dimensão N , quer em torno do coeficiente inicial, quer em torno do coeficiente final, obtendo-se uma sequência para processamento de dimensão $N+8$, como se mostra na figura 4.1.

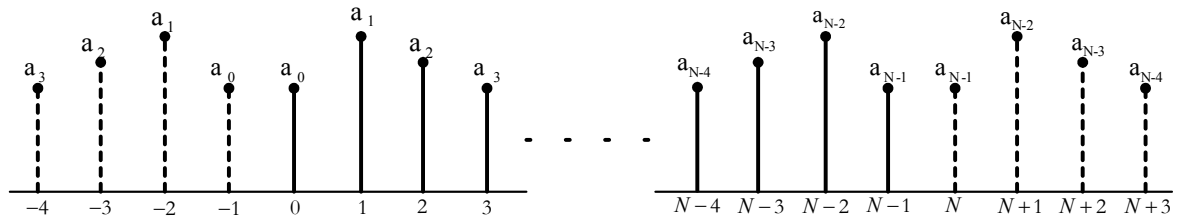


Figura 4.1: Extensão simétrica da sequência nas regiões limite, para o cálculo da DWT^{-1} com os filtros Daubechies 9/7.

Os coeficientes relativos à extensão da sequência, no início e no fim, apenas são usados nos cálculos intermédios, não sendo considerados na actualização dos valores finais da sequência recuperada. Mantém-se pois inalterados os requisitos de memória iniciais, preservando-se, portanto, uma das vantagens do esquema progressivo. Este método de extensão é adequado aos filtros Daubechies 9/7, devido às suas características de simetria, permitindo obter bons resultados nas imagens recuperadas.

Na extensão das sequências nas regiões limite consideram-se os seguintes casos: 1 par de coeficientes (aproximação e detalhe), no início e no fim da sequência, e dois pares de coeficientes (aproximação e detalhe), no início e no fim da sequência, respectivamente para as ôndulas ortogonais DB2 e biortogonais (4,4). Este facto está directamente ligado às características de recursividade dos algoritmos.

As rotinas para cálculo das transformadas no tempo e no espaço foram programadas em C, num computador equipado com um processador de uso geral (Intel Pentium™ III a 550 MHz). Na figura 4.2 apresenta-se um grupo de imagens originais, constituído pelas imagens 5 a 8 da sequência “Coastguard”, que formam o 2º GOF. Na figura 4.3, apresentam-se as imagens relativas à transformação no tempo (DWT do GOF no tempo), e na figura 4.4 apresentam-se as imagens após a aplicação da DWT no espaço a cada imagem do GOF transformado, com 3 níveis de decomposição. As sequências das figuras 4.2 a 4.4 contemplam todas as componentes das imagens, isto é, a luminância e as duas crominâncias.



Figura 4.2: Imagens originais: tramas 5 a 8 (2° GOF) da sequência "Coastguard".

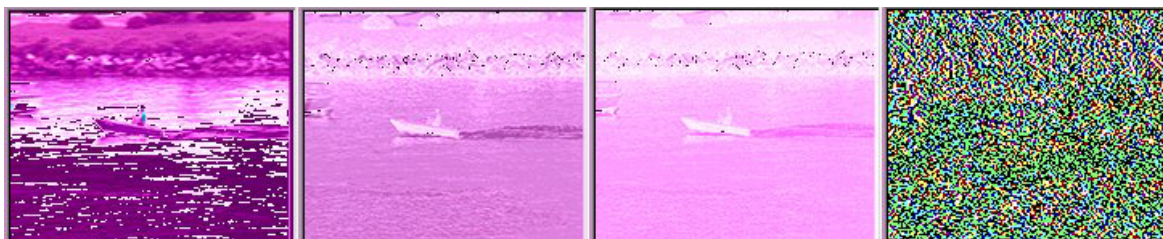


Figura 4.3: Imagens transformadas no tempo: tramas 5 a 8 da sequência "Coastguard".

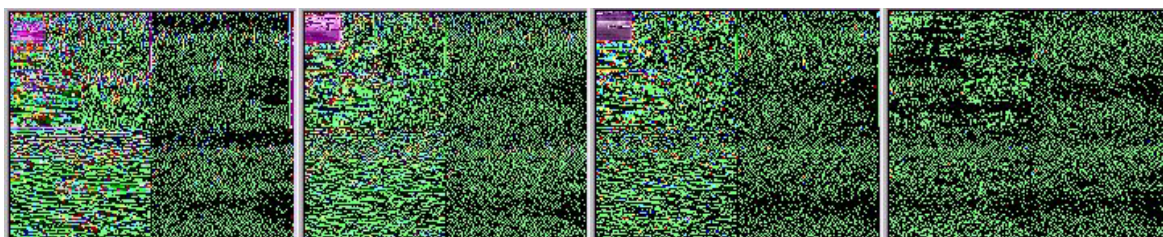


Figura 4.4: Imagens transformadas no tempo e no espaço: tramas 5 a 8 da sequência "Coastguard".

Na sequência da figura 4.3 nota-se a maior intensidade dos coeficientes da primeira trama do GOF, face aos coeficientes das restantes tramas, e também o decréscimo do valor dos coeficientes à medida que o número da trama aumenta dentro do GOF. Os coeficientes da última trama têm os valores mais baixos. Nota-se também que as duas primeiras tramas do GOF contêm os coeficientes de maior valor (maior significado). Isto deve-se à propriedade de decorrelacionamento de informação e de concentração de energia da DWT, neste caso em 2 níveis (recorde-se que se usam 2 níveis de decomposição na DWT no tempo). A primeira trama do GOF contém os coeficientes de aproximação, e as restantes três contêm os coeficientes de detalhe. Numa primeira fase (primeiro nível) as tramas 1 e 2 do GOF transformado correspondem às aproximações aos pares de tramas (1,2) e (3,4) do GOF original, e as tramas 3 e 4 correspondem aos detalhes. Numa segunda fase (2° nível), a trama 1

da figura 4.3 é uma aproximação às tramas 1 e 2 na fase anterior, e a trama 2 corresponde aos detalhes. As tramas 3 e 4 permanecem inalteradas.

Isto mesmo pode ser confirmado pela observação dos objectos contidos nas imagens, tendo como referência a proa do navio à esquerda na imagem. Fazendo a correspondência com as imagens da sequência original na figura 4.2, por analogia, notam-se semelhanças entre as tramas 1 e 2 da figura 4.3 e as tramas 1-2 e 3-4 da sequência da figura 4.2; a terceira trama na figura 4.3 em particular é muito semelhante à primeira trama da figura 4.2 (é o detalhe das tramas 1 e 2 da figura 4.2). A última trama da figura 4.3, devido ao valor baixo dos seus coeficientes, não permite qualquer informação visual útil.

As tramas da figura 4.4 correspondem à aplicação da DWT no espaço, com 3 níveis de decomposição, às tramas da figura 4.3. Observa-se a concentração da informação numa região correspondente a $\frac{1}{64}$ da dimensão da imagem, a separação entre os níveis da DWT no espaço, e a diminuição do valor dos coeficientes com o aumento da banda de frequências na trama.

Os valores de desempenho computacional que se apresentam na tabela 4.1, referem-se a grupos de 4 imagens em formato QCIF. Cada imagem é composta por uma matriz de luminância, Y (144×176 pixels), e duas matrizes de crominância, U (72×88 pixels) e V (72×88 pixels).

	Luminâncias Y tempo de cálculo [ms]	Crominâncias U/V tempo de cálculo [ms]	Total tempo de cálculo [ms]
DWT no tempo (DB2)	70	20	110
DWT⁻¹ no tempo (DB2)	190	50	290
DWT no espaço (DB2)	180	55	290
DWT⁻¹ no espaço (DB2)	580	160	900
DWT no espaço (9/7)	150	50	250
DWT⁻¹ no espaço (9/7)	170	65	300

Tabela 4.1: Tempos aproximados de processamento necessários ao cálculo da DWT e da DWT⁻¹, para grupos de 4 imagens.

Analisando os valores do tempo total por GOF na tabela 4.1, destacam-se os seguintes casos:

1. Em qualquer dos casos, o cálculo da DWT é mais rápido que o cálculo da respectiva DWT^{-1} ;
2. O cálculo da DWT e da DWT^{-1} no tempo é mais rápido do que o cálculo da DWT e da DWT^{-1} no espaço, em qualquer das situações consideradas;
3. Para a ôndula DB2, o cálculo da DWT no tempo é cerca de 3 vezes mais rápido do que o cálculo da DWT^{-1} , e o mesmo se verifica para a DWT/ DWT^{-1} no espaço.

Várias razões explicam os valores da tabela 4.1. Em primeiro lugar, na DWT^{-1} há necessidade de realizar a extensão das sequências para recuperação. No caso das ôndulas biortogonais 4,4 (DWT^{-1} no espaço) faz-se a extensão em simetria, preenchendo-se as regiões limite da sequência de processamento de acordo com a figura 4.1. Para a ôndula DB2 faz-se a extensão obtendo os valores por regressão, tanto para a DWT^{-1} no tempo como para a DWT^{-1} no espaço, sendo as operações intermédias realizadas em vírgula flutuante, de modo a minimizar os erros por arredondamento.

Por outro lado, na DWT no tempo processam-se 2 níveis de decomposição em sequências de quatro coeficientes¹⁷, tanto para as matrizes relativas às luminâncias como para as matrizes relativas às crominâncias, isto é, $(144 \times 176) S_{[4]} + (2 \times 72 \times 88) S_{[4]}$, enquanto que na DWT^{-1} processam-se $(144 \times 176) S_{[8]} + (2 \times 72 \times 88) S_{[8]}$. Na DWT e na DWT^{-1} no espaço efectua-se o processamento com 3 níveis de decomposição. Para a DWT no espaço, quer para a ôndula Daubechies 2, quer para as ôndulas biortogonais (4,4), tem-se $4 \times [144 S_{[176]} + 72 S_{[88]} + 36 S_{[44]} + 176 S_{[144]} + 88 S_{[72]} + 44 S_{[36]}]$ para as matrizes relativas às luminâncias e $2 \times 4 \times [72 S_{[88]} + 36 S_{[44]} + 18 S_{[22]} + 88 S_{[72]} + 44 S_{[36]} + 22 S_{[18]}]$ para as matrizes relativas às crominâncias. Por outro lado, na DWT^{-1} no espaço a dimensão das sequências é diferente, e depende da ôndula considerada. Para a ôndula DB2 tem-se $4 \times [144 S_{[180]} + 72 S_{[92]} + 36 S_{[48]} + 176 S_{[148]} + 88 S_{[76]} + 44 S_{[40]}]$, e

¹⁷ As sequências de coeficientes a processar são referenciadas por $S_{[n]}$, em que n indica o número de coeficientes. Neste caso em particular, como $n = 4$, são referenciadas por $S_{[4]}$.

$2 \times 4 \times [72 S_{[92]} + 36 S_{[48]} + 18 S_{[26]} + 88 S_{[76]} + 44 S_{[40]} + 22 S_{[22]}]$, para as matrizes das luminâncias e para as matrizes das crominâncias, respectivamente; Para Daubechies 9/7 tem-se $4 \times [144 S_{[184]} + 72 S_{[96]} + 36 S_{[52]} + 176 S_{[152]} + 88 S_{[80]} + 44 S_{[44]}]$, para as matrizes das luminâncias e $2 \times 4 \times [72 S_{[96]} + 36 S_{[52]} + 18 S_{[30]} + 88 S_{[80]} + 44 S_{[44]} + 22 S_{[26]}]$ para as matrizes das crominâncias. As sequências de processamento (sequências de trabalho) são portanto maiores na DWT^{-1} e, portanto, o seu tempo de processamento é maior do que para a DWT correspondente. Além disso, na DWT^{-1} no tempo (no decodificador) recuperam-se as tramas com base nos valores gerados na codificação.

Por outro lado ainda, no algoritmo para cálculo da DWT e DWT^{-1} relativa às ôndulas biortogonais (4,4), apesar deste ser mais complexo do que o correspondente à ôndula Daubechies 2, apresenta menor peso computacional. Tal justifica-se pelas características de recursividade dos algoritmos relativos às ôndulas Daubechies 2 e biortogonais (4,4) (ver apêndice B). No algoritmo para o cálculo da DWT e DWT^{-1} para a ôndula DB2, em cada iteração calcula-se um par de coeficientes (aproximação e detalhe) finais, e um par de coeficientes intermédios. No algoritmo da DWT e DWT^{-1} para os filtros 9/7, calcula-se um par de coeficientes finais e dois pares de coeficientes intermédios, em cada iteração. Além disso, devido às diferenças no processo de compensação das sequências nas regiões limite, não sendo necessário efectuar operações de multiplicação adicionais, em vírgula flutuante, o cálculo da DWT^{-1} para os filtros 9/7 é mais rápido do que o cálculo da DWT^{-1} para a ôndula DB2.

Visando aumentar o desempenho, programaram-se as rotinas de cálculo da DWT e da DWT^{-1} num processador digital de sinal de elevado desempenho, o DSP TMS320C6201 da Texas Instruments [Tex_187, Tex_198].

4.5 Principais Características do DSP TMS320C6201

O TMS3206201 é um DSP em vírgula fixa, baseado na plataforma C6000 [Tex_197, Tex_198], com arquitectura VelocíTI™. A arquitectura VelocíTI™ é uma arquitectura¹⁸ avançada do tipo VLIW, de elevado desempenho, desenvolvida pela Texas Instruments e que mantém espaços

¹⁸ VelocíTI é uma marca registada da Texas Instruments Incorporated.

de memória distintos para dados e programa (arquitetura do tipo Harvard) que possui uma *cascata* com 11 andares de profundidade, e que permite executar até 8 instruções em paralelo, num único ciclo de relógio. Por exemplo, para o TMS320C6201 a 167 MHz tem-se uma capacidade de processamento de pico de $8 \times 167 \times 10^6 = 1336$ MIPS.

Na figura 4.5 apresenta-se o diagrama de blocos da arquitectura do DSP C6201, onde se destaca a existência de dois blocos de processamento independentes, cada um com quatro unidades funcionais: 1 multiplicador de 16×16 bits e 3 ALU de $32/40$ bits, i.e., no total possui 8 unidades funcionais independentes (6 ALU e 2 multiplicadores). A palavra de instrução tem dimensão 256 bits (8 instruções de 32 bits para as 8 unidades funcionais).

Assim, na situação de máximo aproveitamento dos recursos (total independência das instruções), os DSPs C6201 pode executar até 8 instruções num único ciclo de relógio, e realizar operações com dados de dimensão variável (8/16/32 e 40 bits), cujos tipos e formatos se apresentam na tabela 4.2.

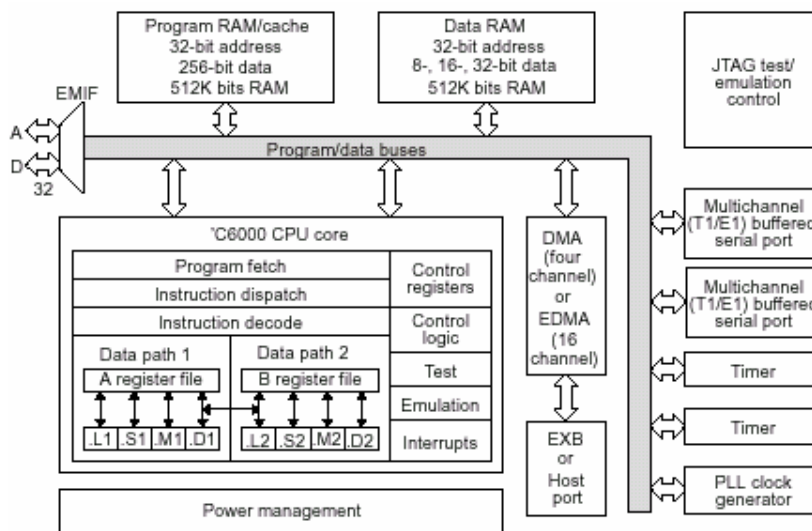


Figura 4.5: Diagrama de blocos do DSP TMS320C6201

O núcleo de CPU¹⁹ do DSP C6201 contém unidades funcionais para preparação (*program fetch*), fraccionamento e descodificação (*instruction dispatch* e *instruction decode*), dois blocos para processamento de dados (*data path A* e *B*) cada um com 4 unidades funcionais e dois bancos de registos, compostos por 16 registos de 32 bits cada. Possui ainda unidades para

¹⁹ Do Inglês, *Central Processing Unit*, Unidade Central de Processamento.

processamento de interrupções, para ligação a periféricos, memórias tampão para programa (*cache*) e controlador de acesso directo à memória (DMA).

Tipo (C)	char	Short	int
Dimensão	8 <i>bits</i>	16 <i>bits</i>	32 <i>bits</i>

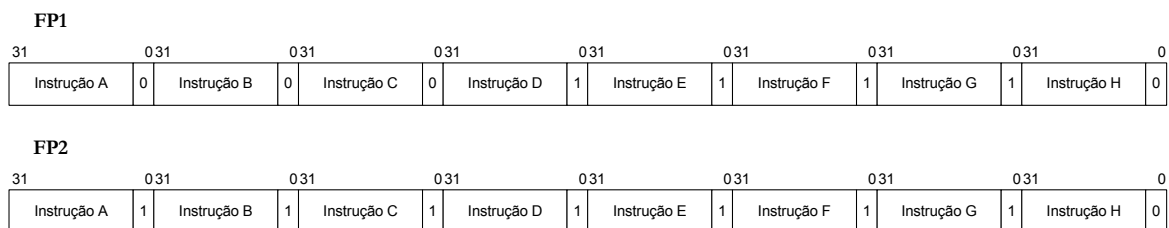
Tabela 4.2: Tipos e formatos dos dados suportados pelo DSP TMS320C6201.

As arquitecturas VLIW tradicionais caracterizam-se pela existência de várias unidades funcionais, utilizadas na execução de várias instruções em paralelo. Contudo, neste tipo de arquitecturas, a execução de instruções de controlo de fluxo de programa (*branching instructions*) pode originar um acréscimo no tempo total de execução, devido ao elevado número de andares da *casca*. A arquitectura VelociTI™ mantém o paralelismo das arquitecturas VLIW tradicionais apresentando ainda as seguintes vantagens:

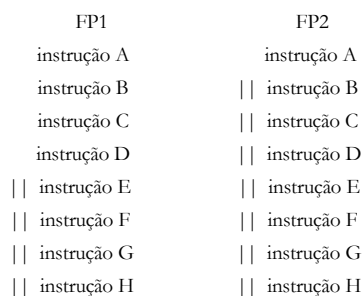
- Empacotamento de instruções, que podem ter dimensão variável (no máximo 256 *bits*);
- Todas as instruções são executadas condicionalmente e em *casca*, incluindo as instruções de controlo do fluxo do programa, em relação ao valor de um dos registos.

Um pacote de instruções FP (*fetch packet*), representado por 256 *bits*, compreende um número fixo de oito instruções (32 *bits* cada) que devem ser executadas sequencialmente ou em paralelo. No ciclo de processamento, procede-se à formação do pacote de execução EP (*execute packet*) que compreende a preparação (*fetch*) do pacote de instruções, o fraccionamento (*instruction dispatch*) e a decodificação (*instruction decode*) das instruções.

Um pacote de execução é constituído por um número variável de instruções, entre uma e oito instruções, que se executam em paralelo. O *bit* menos significativo (LSB) do código de uma instrução (*bit p*) indica se a instrução seguinte do pacote de instruções deve ou não ser executada em paralelo com a instrução actual. Caso o *bit p* da instrução *n* tenha o valor ‘1’, a instrução *n+1* é executada no mesmo ciclo que a instrução *n*; caso contrário, a instrução *n+1* é executada no ciclo seguinte ao da instrução *n* (figura 4.6). Isto significa que o *bit* menos significativo do código das instruções que compõem um pacote de execução tem o valor ‘1’, com excepção da última instrução em que o valor é ‘0’, indicando que a próxima instrução deve ser executada em série.



a)



b)

Figura 4.6: Pacote de instruções parcialmente sequencial e totalmente em paralelo: a) codificação; b) paralelismo (| |) na execução.

A situação de máximo aproveitamento dos recursos corresponde à formação de um pacote de execução com oito instruções em paralelo. De modo a facilitar o controlo do processador, o *bit* *p* da última instrução é sempre colocado a '0', iniciando-se assim novo pacote de execução e tornando impossível a execução em paralelo de instruções de pacotes diferentes.

Este esquema de empacotamento das instruções constitui uma das principais inovações do DSP utilizado, que permite ultrapassar o problema do desperdício de memória de programa verificado nas arquitecturas VLIW tradicionais.

4.5.1 Execução das Instruções

A execução das instruções tem lugar nas unidades funcionais (.L, .S, .M e .D) existentes em cada um dos blocos de processamento (*data path* A e B), representados na figura 4.7. Cada um dos blocos de processamento de dados compreende 16 registos genéricos de 32 *bits*, que podem ser usados para armazenar dados, apontadores ou ainda como registos condicionais. Os registos genéricos suportam operandos em vírgula fixa, com 32 ou 40 *bits* de dimensão.

Para acomodar as operações com operandos de 40 *bits*, são utilizados pares de registos, como por exemplo: A1:A0, A3:A2, B9:B8, etc; os 32 LSB da palavra de dados são colocados num

registo par; e os restantes 8 *bits* mais significativos (MSB) são colocados na posição correspondente aos 8 LSB do registo imediato (registo ímpar).

As unidades funcionais de um bloco de processamento, cujas operações se enumeram na tabela 4.3, têm unidades homólogas no outro bloco de processamento. Cada unidade funcional possui portos de 32 *bits*, um para escrita (*dst*) e dois para leitura (*src1*) e (*src2*), de operandos dos registos genéricos (ver figura 4.7). As unidades funcionais de um bloco de processamento têm acesso directo aos registos do banco associado. Os dois bancos de registos estão interligados através dos caminhos de cruzamento (*cross path*) 1X e 2X, de modo que as unidades funcionais .M, .S, e .L de um bloco de processamento podem ter acesso a apenas um registo do bloco homólogo, em cada ciclo de relógio. As unidades funcionais .L e .S dispõem de dois portos extra de 8 *bits* (*long dst* e *long src*), para operar directamente sobre operandos de 40 *bits*. Nas unidades .M, .S e .L, a informação nos portos *src2*, resulta da selecção por *multiplexer* dos dados do banco de registos associado, ou do banco de registos associado ao bloco homólogo. Nas unidades .L, esta característica é também extensiva ao porto *src1*.

Como cada unidade funcional possui um porto independente de 32 *bits*, podem-se utilizar em paralelo a totalidade das unidades funcionais, em cada ciclo de relógio, salvaguardando as seguintes restrições:

- Num pacote de execução, cada unidade funcional apenas pode ser usada por uma instrução (conflito na utilização dos recursos);
- Num pacote de execução, apenas se pode usar um registo do banco homólogo, dado existir apenas um caminho de cruzamento para leitura de operandos (conflito nos caminhos de cruzamento);
- Num pacote de execução, não pode existir mais do que uma operação sobre o mesmo registo (conflito no acesso aos registos);
- A cada pacote de execução apenas pode ser afectada uma operação com dados de 40 *bits*, porque as unidades .S e .L partilham o mesmo porto de leitura para operações a 40 *bits*.

A unidade .S2 permite ainda o acesso ao banco de registos de controlo (*control register file*). Os caminhos: LD1, LD2, ST1 e ST2 são caminhos de dados de 32 *bits*, para leitura de dados de memória nos registos e escrita de dados dos registos em memória, respectivamente.

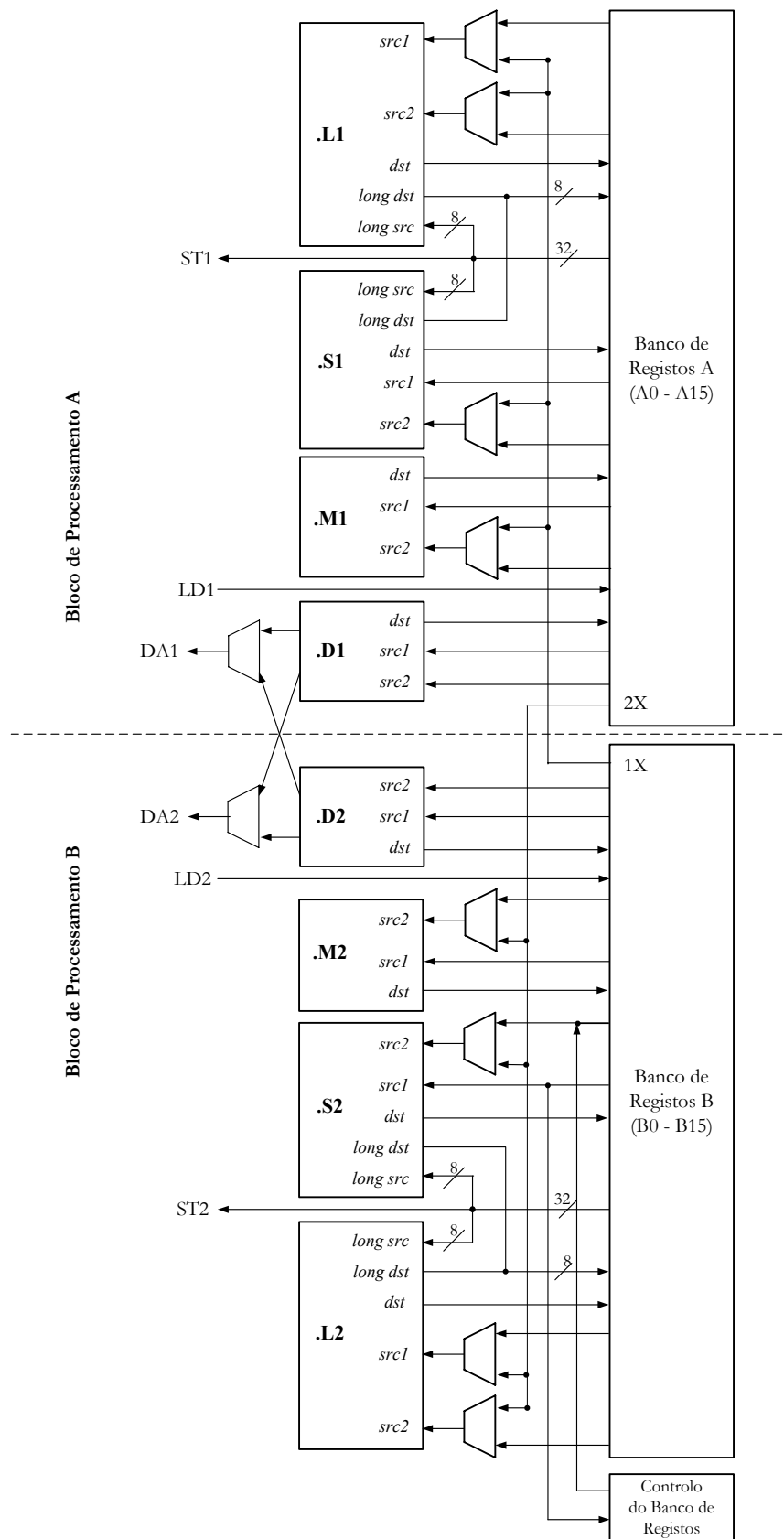


Figura 4.7: Blocos de processamento de instruções no DSP TMS320C6201.

Os caminhos ST1 e ST2 são partilhados com entradas das unidades funcionais .L e .S, para suportar operações com operandos de 40 *bits* (tipo *long*).

Unidade Funcional	Operações em vírgula fixa
.L1 e .L2	Operações aritméticas e de comparação sobre operandos de 32/40 <i>bits</i> ; Operações lógicas a 32 <i>bits</i> ; Normalização de resultados de 32/40 <i>bits</i> (eliminação de <i>bits</i> de sinal redundantes);
.S1 e .S2	Operações aritméticas e lógicas a 32 <i>bits</i> ; Deslocamentos em operandos de 32/40 <i>bits</i> ; Saltos para troços de programa (<i>branching</i>); Geração de valores imediatos; Transferência de dados de e para os registos de controlo (apenas em .S2)
.M1 e .M2	Operações de multiplicação com operandos a 16 <i>bits</i> ;
.D1 e .D2	Endereçamento; Acessos a memória;

Tabela 4.3: Operações realizadas nas diversas unidades funcionais.

Existem ainda caminhos para endereços (*data address paths*) DA1 e DA2, à saída das unidades funcionais .D²⁰, permitindo-se que num banco de registos se efectuem operações de acesso a memória, usando endereços gerados no outro banco de registos.

4.5.2 Funcionamento em *Cascata*

A operação da *cascata* do DSP TMS320C6201 é composta por três grandes fases (figura 4.8): preparação, descodificação e execução. A *cascata* do DSP TMS320C6201 tem 11 andares e o seu funcionamento é completamente síncrono. A existência de um grande número de andares da *cascata* permite aumentar o ritmo de processamento, mas aumenta também a latência de execução das instruções.

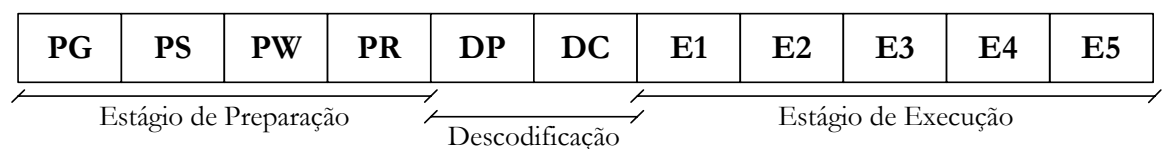


Figura 4.8: Fases e andares que compõem a *cascata*.

²⁰ Note-se que as unidades funcionais .D são responsáveis pelas operações de endereçamento.

No DSP TMS320C6201, todas as instruções cumprem as três fases de operação da *cascata*, na mesma sequência. O número de andares das fases de preparação e de descodificação é comum em todas as instruções, i.e., na fase de preparação todas as instruções passam por quatro andares e na fase de descodificação por dois andares. O número de andares da fase de execução é variável, consoante o tipo de instrução a executar. Na tabela 4.4 apresenta-se o processamento realizado nos vários andares que compõem a *cascata*.

As instruções em paralelo avançam na *cascata*, simultaneamente, em cada fase; as instruções sequenciais avançam na *cascata* com uma diferença de fase relativa constante. Na situação de máximo aproveitamento de recursos O DSP tem capacidade para processar até oito instruções num único ciclo, i.e., a cada pacote de instruções corresponde um pacote de execução com oito instruções em paralelo, como ilustra a figura 4.9. O processamento é feito pelas diferentes unidades funcionais do DSP, à medida que o código avança nas diferentes fases da *cascata*.

Ref.	Fase	Andar	Operação
PG	Preparação	<i>Program Address Generate</i>	Determinação endereço do pacote de instruções
PS		<i>Program Address Send</i>	End. pacote de preparação enviado a memória
PW		<i>Program Wait</i>	CPU aguarda dados obtidos por acesso a memória ...
PR		<i>Program Data Receive</i>	Recepção dos dados e finalização.
DP	Descd.	<i>Instruction Dispatch</i>	O próximo pacote de execução, do pacote de instruções, é formado e enviado às unidades funcionais para descodificação;
DC		<i>Instruction Decode</i>	Descodificação de instruções (unid. funcionais).
E1	Execução	<i>Execute 1</i>	Avaliação condicional da instrução e leitura dos operandos (todas as instruções); Geração de endereços e colocação em registos (op. acesso a memória); Escreve o resultado das instruções de 1 ciclo;
E2		<i>Execute 2</i>	Envio endereços a memória (op. leitura) e envio de endereços e dados a memória (op. escrita); Escreve o resultado da multiplicação em registos;
E3		<i>Execute 3</i>	Acessos a memória de dados;
E4		<i>Execute 4</i>	Leitura de dados e escrita em registo (nas operações de leitura) ...
E5		<i>Execute 5</i>	Escritos num dos dois bancos de registos.

Tabela 4.4: Descrição das operações executadas nas fases que compõem a *cascata*.

pacote de instruções	pacote de execução															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	m	PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5				
$n+1$	$m+1$		PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5			
$n+2$	$m+2$			PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5		
$n+3$	$m+3$				PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5	
$n+4$	$m+4$					PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5
$n+5$	$m+5$						PG	PS	PW	PR	DP	DC	E1	E2	E3	E4
$n+6$	$m+6$							PG	PS	PW	PR	DP	DC	E1	E2	E3
$n+7$	$m+7$								PG	PS	PW	PR	DP	DC	E1	E2
$n+8$	$m+8$									PG	PS	PW	PR	DP	DC	E1
$n+9$	$m+9$										PG	PS	PW	PR	DP	DC

Figura 4.9: Exemplo do funcionamento da *cascata*, na situação de máxima eficiência.

O ciclo de CPU é o intervalo de tempo, durante o qual um determinado pacote de execução está numa determinada fase da *cascata*. Os limites dos ciclos CPU ocorrem sempre em simultâneo com os limites dos ciclos de relógio. A maioria das instruções é executada em apenas um ciclo, o que pressupõe, nesses casos, a existência apenas do andar E1 da fase de execução. Porém, outras instruções necessitam mais do que um ciclo e, conseqüentemente, mais do que um andar na fase de execução, como se observa na tabela 4.4.

Como o funcionamento da *cascata* é síncrono, a execução das instruções pode ser definida em termos de fatias temporais de atraso, ou latência (*delay slots*), correspondentes a ciclos de CPU que ocorrem após a fase E1 de uma instrução. O resultado das instruções com *delay slots*, só está disponível após o último *delay slot* (ver tabela 4.5). Para manter a cadência do funcionamento da *cascata*, a instrução *branch* requer cinco *delay slots*, correspondentes ao preenchimento das fases PS, PW, PR DP e DC precedentes à sua fase E1, isto é, antes do início do salto para o endereço de destino. No entanto, como o processador utiliza salto retardado, os cinco *delay slots* podem ser preenchidos com instruções úteis.

	1 ciclo	Multiplicação	Escrita	Leitura	branch
E1	Determinação do resultado e escrita em registo.	Leitura dos operandos e início da computação ...	Determinação de endereço ...	Determinação de endereços...	Endereço de destino na fase PG.
E2		Determinação do resultado e escrita em registo.	Envio de dados e endereços para a memória ...	Envio de endereços à memória ...	
E3			Acesso a memória.	Acesso a memória...	
E4				Leitura de dados e envio ao CPU ...	
E5				Escrita dos dados em registo.	
delay slots	0	1	0	4	5

Tabela 4.5: Andares da fase de execução e latência associada, por tipo de instrução.

4.5.3 Limitações à Eficiência da *Cascata*

Na figura 4.9 apresentou-se o funcionamento da *cascata* na situação de desempenho máximo. O fluxo das instruções através da *cascata* é determinado pelo número de pacotes de execução num dado pacote de instruções. Como as várias instruções cumprem fases diferentes de execução, o funcionamento da *cascata* considera os instantes de conclusão das instruções. A conjugação destes dois factores origina diminuição do desempenho, por paragem temporária da *cascata* (*pipeline stall*) e do processamento das instruções, como se apresenta na figura 4.10.

Considere-se, por exemplo, o pacote de instruções $n + 1$, representado da figura 4.10. No ciclo 6, na fase DP do processamento do pacote de execução $m + 1$, o CPU avalia o bit p de cada instrução detectando 5 pacotes de execução. Isto obriga a uma paragem temporária da *cascata* para que se iniciem as fases DP dos restantes pacotes de execução ($m + 2$ a $m + 5$), do pacote de instruções $n + 1$. O processamento dos pacotes de instruções $n + 2$ a $n + 5$, entretanto iniciado, é suspenso, sendo apenas retomado quando o pacote de execução $m + 5$ passa à fase DC (ciclo 11). O processamento do pacote de execução $m + 10$, que deveria ter-se iniciado no ciclo 7, é igualmente suspenso, tendo início 5 ciclos mais tarde. O caso mais desfavorável ocorre para a situação de um pacote de instruções com oito pacotes de execução, isto é, oito instruções executadas sequencialmente.

Podem ainda ocorrer situações de degradação do desempenho, por paragem da *cascata*, nos acessos a memória (*memory stalls*). Esta situação verifica-se quando os tempos de acesso a memória são maiores do que a cadência de funcionamento da *cascata* ou quando há conflitos no acesso à memória.

pacote de instruções	pacote de execução															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	m	PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5				
$n+1$	$m+1$		PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5			
$n+1$	$m+2$						DP	DC	E1	E2	E3	E4	E5			
$n+1$	$m+3$							DP	DC	E1	E2	E3	E4	E5		
$n+1$	$m+4$								DP	DC	E1	E2	E3	E4	E5	
$n+1$	$m+5$									DP	DC	E1	E2	E3	E4	
$n+2$	$m+6$	PG	PS	PW	PR						DP	DC	E1	E2	E3	
$n+3$	$m+7$		PG	PS	PW						PR	DP	DC	E1	E2	
$n+4$	$m+8$			PG	PS						PW	PR	DP	DC	E1	
$n+5$	$m+9$				PG						PS	PW	PR	DP	DC	

Paragem da cascata
(*pipeline stall*)

Figura 4.10: Exemplo de perda de eficiência no processamento por paragem da *cascata*.

Apesar do espaços de memória de dados e de programa serem distintos, os acessos a memória de programa para preparação (*program fetch*) e a memória de dados para leitura, cumprem o mesmo tipo de operação na *cascata*, embora se completem em fases diferentes. Como o ritmo dos acessos a memória é determinado pelo ritmo do CPU, a *cascata* pode parar para garantir o adequado controlo das operações.

A eficiência da cascata pode também ser limitada pelo agendamento estático das instruções na *cascata*. As técnicas de agendamento de instruções na *cascata* têm por objectivo o início de uma nova iteração o mais breve possível, possibilitando a execução de múltiplas iterações em paralelo, tendo em conta os tempos de latência das instruções. Por exemplo, na leitura de um operando (LDH, LDW) ou na execução de uma instrução de controlo de fluxo de programa (*Branch*), os resultados da execução das instruções só é efectivo após o cumprimento dos tempos de latência associados, isto é, 4 e 5 *delay slots*, respectivamente. Caso exista uma dependência dos dados em relação à próxima instrução a executar, é necessário preencher o tempo que se espera até que os resultados das instruções estejam disponíveis. No caso de não ser possível o preenchimento com instruções úteis, isto pode ser feito com a instrução NOP (*No Operation*). Porém, caso as instruções sejam executadas iterativamente num ciclo de

programa, é possível iniciar uma nova iteração enquanto os resultados não estão disponíveis, respeitando o intervalo mínimo por iteração. Desta forma, em vez da instrução NOP, os tempos de latência são ocupados com a execução de instruções referentes a novas iterações.

4.6 Ambiente de Desenvolvimento para o DSP TMS320C6201

A Texas Instruments disponibiliza um conjunto de ferramentas de desenvolvimento para os seus DSPs em geral, e em particular para os DSPs TMS320C62x, como sejam linguagens de programação em alto ou baixo nível, (“C” e *assembly*), que apoiam a programação eficiente daqueles DSPs e ferramentas que permitem avaliar o seu desempenho.

O conjunto de ferramentas de desenvolvimento mais recentes [Tex_198, Tex_328] inclui um compilador de “C/C++” (suporta as linguagens ANSI “C”, ”C++” e funções específicas do DSP), o otimizador de C [Tex_187], o *Assembler* e o otimizador de *Assembly* [Tex_186, Tex_189], o *Linker* e o *Debugger*. O formato dos dados para o compilador de “C”/”C++” do DSP C62x é o apresentado na tabela 4.2.

Para além destas ferramentas para programação, a Texas Instruments disponibiliza ainda um módulo de *hardware* (TMS320C6201 EVM), para suporte ao desenvolvimento de aplicações. É um módulo para computador pessoal, com barramento PCI, que integra o DSP TMS320C6201 a 167 MHz (1336 MIPS), e que foi usado no trabalho que se apresenta nesta tese. O módulo de avaliação TMS320C6201 EVM possui dois bancos de memória síncronos: um banco de 256 kB de memória estática (SBSRAM)²¹, a 133 MHz, e um banco de memória dinâmica (SDRAM)²², de dimensão de 1 MB, expansível até 8 MB. O desenvolvimento de aplicações envolve, regra geral, as fases de escrita e refinamento de código em “C/C++”, com recurso às funções específicas e às diversas opções do compilador e do otimizador de “C/C++”. É um processo iterativo que compreende a escrita, a avaliação do desempenho, o refinamento e nova avaliação de desempenho. Obtém-se assim uma optimização significativa, sem grande dispêndio de tempo. Porém, podem ser obtidos desempenhos melhores programando em linguagens de baixo nível, em *linear assembly*²³ e/ou *assembly*.

²¹ Do Inglês, *Synchronous Burst Static Random-Access Memory*.

²² Do Inglês, *Synchronous Dynamic Random-Access Memory*.

²³ *Linear Assembly* é uma fase de escrita de código intermédia (entre “C/C++” e *Assembly*), muito próxima da escrita em *Assembly*, sem necessidade de especificar registos ou unidades funcionais.

4.7 Implementação das Transformadas de Ôndulas no DSP TMS320C6201

Foram desenvolvidas rotinas de cálculo da DWT e DWT^{-1} para as ôndulas consideradas, que tiram partido das capacidades do processador digital de sinal TMS320C6201, que pode funcionar como co-processador. A optimização das rotinas incidiu essencialmente nas facilidades disponibilizadas com as ferramentas de desenvolvimento para o DSP C6201, para código escrito em linguagem de alto nível. A optimização visou obter a máxima eficiência possível, tendo em conta a arquitectura do DSP, aplicando o seguinte conjunto de técnicas:

- ✓ Elaboração de tabelas dos coeficientes dos filtros, representados por números inteiros a 16 *bits* (tipo **short**), em vírgula fixa, em notação Q14;
- ✓ Realização, sempre que possível²⁴, de operações de leitura e escrita em memória de operandos a 32 *bits* (tipo **int**), isto é, dois operandos de 16 *bits* de cada vez;
- ✓ Uso de instruções específicas do C62x, visando maximizar a eficiência das unidades de multiplicação: **_mpy** (oper A, oper B), **_mpylh** (oper A, oper B), **_mpyhl** (oper A, oper B), **_mpyh** (oper A, oper B);
- ✓ Eliminação de instruções condicionais (do tipo **if – then – else**) no ciclo central (*kernel*) do processamento pois a sua existência origina a desqualificação do ciclo para agendamento para processamento em *cascata*, por programação (“*software pipeline*”);
- ✓ Uso de uma instrução específica do DSP C6201 para controlo dos ciclos, visando a optimização do agendamento e fluxo de dados na *cascata*: **_nassert** (ctrl>=n_iter);

Dado que todos os operandos são representados em vírgula fixa por 16 *bits*, não há necessidade particular de arredondamentos dos valores finais. Da implementação das transformadas no TMS320C6201 EVM obtiveram-se os resultados que se apresentam nas tabelas 4.6 e 4.7.

²⁴ A máxima rentabilização dos recursos só é obtida quando os dados estão alinhados sequencialmente, com dimensão 16 *bits*.

	Luminâncias Y	Crominâncias U/V	Total
DWT no tempo (DB2)	760.320	190.080	1.140.480
DWT⁻¹ no tempo (DB2)	2.483.712	620.928	3.725.568
DWT no espaço (DB2)	16.302.528	4.075.632	24.453.792
DWT⁻¹ no espaço (DB2)	26.210.448	6.552.612	39.315.672
DWT no espaço (9/7)	24.719.112	6.179.778	37.078.668
DWT⁻¹ no espaço (9/7)	23.145.804	5.786.451	34.718.706

Tabela 4.6: Número de ciclos de relógio necessários ao processamento da DWT e da DWT⁻¹, para grupos de 4 imagens.

O número de ciclos de relógio permite estabelecer o tempo necessário ao processamento, para várias versões do DSP, e diferentes frequências de relógio. Na tabela 4.7 apresentam-se os tempos computacionais relativos às rotinas referidas, para a versão DSP do TMS320C6201 EVM, e para a versão do DSP com a frequência de relógio mais próxima da frequência do processador de uso geral usado. Apresentam-se ainda os resultados que se inferem para uma possível versão do TMS320C6201 com valor de frequência de relógio igual ao do processador de uso geral (*). Nas figuras 4.11 e 4.12 apresenta-se a aceleração de processamento obtida relativamente à implementação no processador de uso geral (ver tabela 4.1).

Da comparação entre os valores da tabela 4.7 (para as várias versões do DSP) e os valores de tempo total da tabela 4.1, verifica-se um aumento generalizado na rapidez de processamento com os DSPs TMS320C62x face ao processador de uso geral. Com base nos valores do tempo de processamento obtidos no TMS320C6201 EVM, verifica-se que o peso computacional associado à DWT⁻¹ é maior do que na DWT, sendo essa diferença mais significativa nuns casos do que noutros. No cálculo da DWT e da DWT⁻¹ no espaço para os filtros 9/7, a diferença é pouco significativa. Porém, nos restantes dois casos, para a ôndula DB2 (DWT/DWT⁻¹ no tempo e DWT/DWT⁻¹ no espaço), essa diferença é muito maior (entre 1,5 e 3 vezes maior).

O ganho em rapidez que se obtém é, pois, mais significativo no cálculo de transformadas com a ôndula DB2, registando o seu maior valor para a DWT⁻¹ no tempo. As razões que explicam as diferenças no peso computacional entre o cálculo da DWT e da DWT⁻¹ foram referidas na secção 4.4.

	TMS320C6201 167 MHz $T_{CLK} = 6 \text{ ns}$ tempo de cálculo [ms]	TMS320C6203 300 MHz $T_{CLK} = 3,3 \text{ ns}$ Tempo de cálculo [ms]	TMS320C62x 550 MHz(*) $T_{CLK} = 1,82 \text{ ns}$ tempo de cálculo [ms]
DWT tmp (DB2)	7	4	2
DWT⁻¹ tmp (DB2)	23	13	7
DWT esp. (DB2)	147	81	45
DWT⁻¹ esp. (DB2)	236	130	72
DWT esp. (9-7)	223	123	68
DWT⁻¹ esp. (9-7)	209	115	64

Tabela 4.7: Tempos aproximados de processamento para grupos de 4 imagens, considerando as versões do DSP TMS320C62x a 167 e 300 MHz, e uma possível versão a 550 MHz.

Neste caso, porém, importa referir que as características dos algoritmos de processamento, nomeadamente o agendamento de instruções e a dependência dos dados, determinam o grau de paralelismo que é possível obter com as ferramentas de programação do TMS320C6201 EVM. Além disso as operações são realizadas em vírgula fixa, sendo os operandos e os resultados representados por inteiros de 16 *bits*.

Por um lado, na DWT e DWT⁻¹ no tempo calcula-se a transformada e a transformada inversa com dois níveis de decomposição, na mesma rotina, reduzindo o acréscimo de tempo por chamada às rotinas. Por outro lado, os ciclos do núcleo central de processamento (*kernel*) são mais pequenos (2 a 3 iterações), reduzindo o intervalo mínimo entre iterações no agendamento das instruções. Isto permite que o início de várias iterações no mesmo ciclo de processamento.

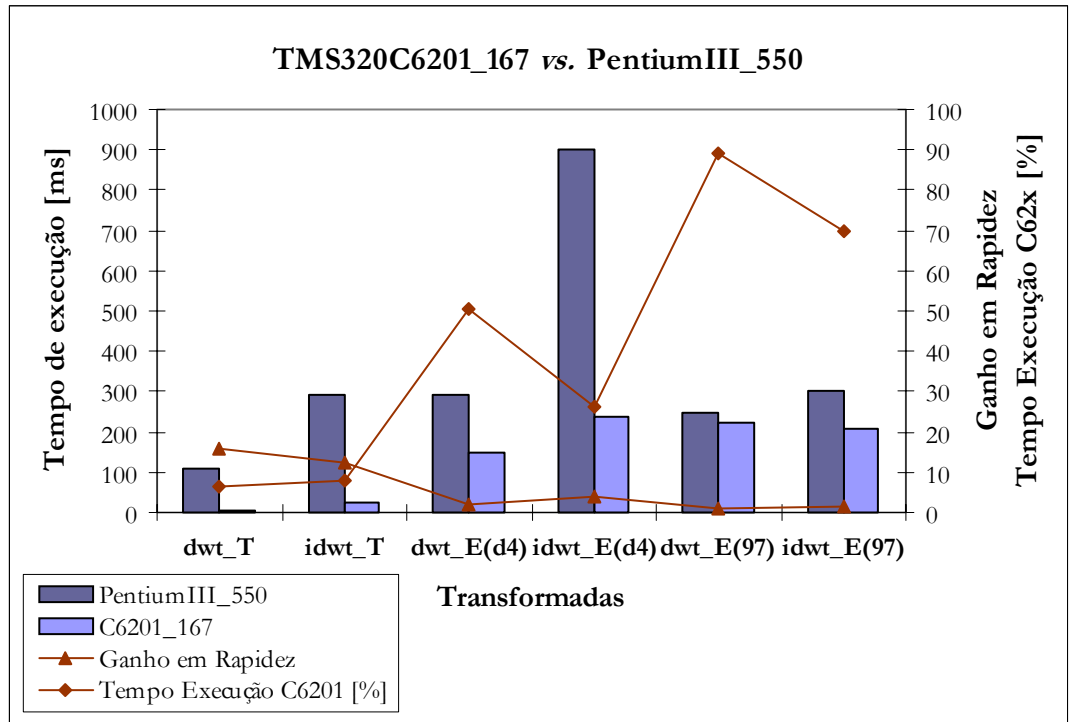


Figura 4.11: Comparação entre o desempenho no TMS320C6201 EVM e no PC equipado com o processador PentiumIII (550 MHz).

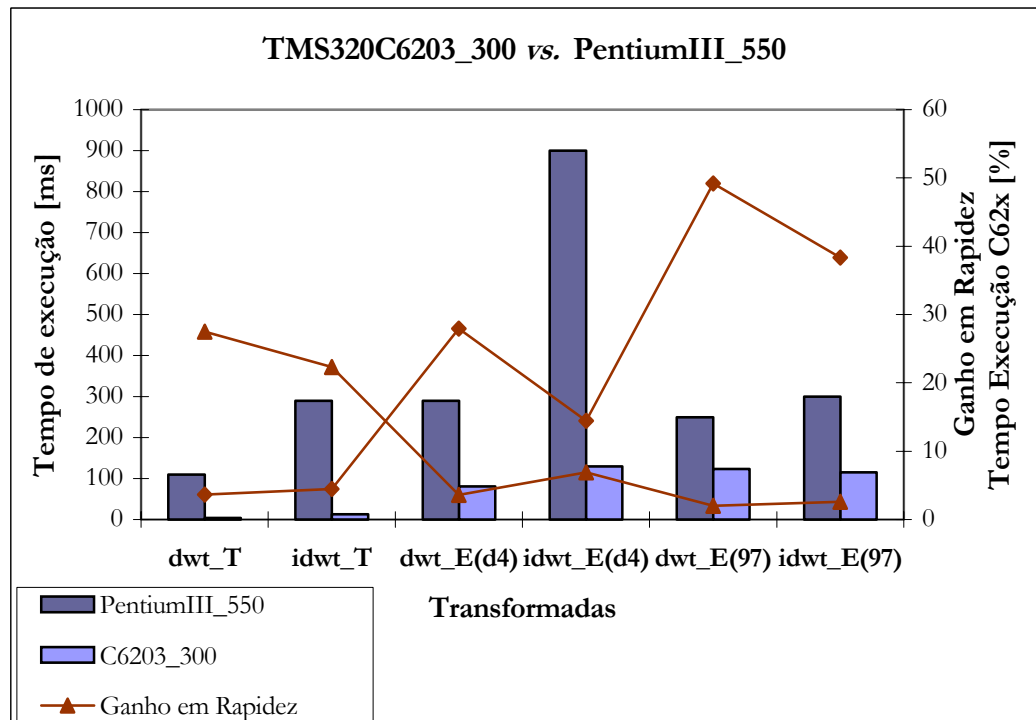


Figura 4.12: Comparação entre o desempenho no DSP TMS320C6203 (300 MHz) e no PC equipado com o processador PentiumIII (550 MHz).

```

; ** -----*
L19:      ; PIPED LOOP KERNEL

          MPY      .M2      B1,B6,B1      ; @|214|
||      MV      .L2X      A6,B6          ; @|216|
||      SHR      .S2      B7,14,B7       ; @|195|
||      MV      .L1      A9,A0          ; @@Inserted to split a long life
||      SUB      .S1      A0,A4,A6       ; @@@|185|
||      LDH      .D1T1     *A5++(4),A0    ; @@@@|185|
||      LDW      .D2T2     *+SP(8),B4     ; @@@@|195|

          SHR      .S1      A7,14,A4      ; |214|
||      SADD     .L2      B8,B7,B8       ; @|195|
||      MPY      .M2      B4,B6,B1       ; @@@|195|
||      MV      .L1      A8,A9          ; @@@Inserted to split a long life
||      MV      .S2      B1,B7          ; @@@@|185|
||      MPYHL    .M1X     B1,A4,A7       ; @@@@|185|
||      LDW      .D1T1     *+A2(20),A4    ; @@@@|185|
||      LDW      .D2T2     *+SP(4),B1     ; @@@@|185|

          AND      .S1      A1,A4,A4      ; |214|
||      MV      .D2      B5,B4          ; @Inserted to split a long life
||      SHL      .S2      B1,2,B8        ; @|214|
||      SADD     .L1X     A3,B8,A3       ; @|195|
||      MV      .L2      B4,B5          ; @@|201|
||      LDH      .D1T1     *+A2(20),A8    ; @@@@|195|

          OR       .L1X     B2,A4,A7      ; |214|
|| [ B0] SUB     .D2      B0,1,B0        ; @|218|
||      AND      .L2      B9,B8,B2       ; @|214|
||      MV      .D1      A3,A3          ; @|201|
||      SHR      .S2      B1,14,B8       ; @@@|195|
||      MPY      .M2X     B3,A6,B7       ; @@@|195|
||      SHR      .S1      A7,14,A4       ; @@@@|185|

          STW      .D1T1     A7,*A2++     ; |214|
|| [ B0] B       .S1      L19            ; @|218|
||      MPYHL    .M1X     B4,A3,A7       ; @|214|
||      MV      .L1      A0,A3          ; @@Inserted to split a long life
||      ADD      .L2X     A3,B6,B6       ; @@@|199|
||      MV      .S2      B7,B3          ; @@@Inserted to split a long life
||      LDW      .D2T2     *+SP(12),B1    ; @@@@|214|

; ** -----*
L20:      ; PIPED LOOP EPILOG

          MPY      .M2      B1,B6,B1      ; @@@|214|
||      MV      .L2X     A6,B6          ; @@@|216|
||      SHR      .S2      B7,14,B7       ; @@@|195|
||      MV      .L1      A9,A0          ; @@@Inserted to split a long life
||      SUB      .S1      A0,A4,A6       ; @@@@|185|
||      LDH      .D1T1     *A5++(4),A0    ; @@@@|185|
||      LDW      .D2T2     *+SP(8),B4     ; @@@@|195|

```

Figura 4.13: Código em *assembly* para o núcleo (*kernel*) de processamento no cálculo da DWT no tempo (onda DB2), e início do epílogo.

No final do ciclo, as iterações entretanto iniciadas são realizadas por um conjunto de instruções, agendadas mais tarde, e que constituem o epílogo do núcleo de processamento. Na figura 4.13 apresenta-se uma parte de código em *assembly* que se obtém para o cálculo a DWT no tempo (ôndula DB2), referente ao núcleo central de processamento, contendo 5 pacotes de execução (em paralelo) e o início do epílogo.

Na DWT e DWT^{-1} no espaço para a ôndula DB2, o algoritmo tem características idênticas, no que se refere ao número de iterações. Porém, por um lado é necessário executar mais operações, como foi referido em 4.4 (ver pp. 75 e 76), por outro lado realizam-se 3 níveis de decomposição na estrutura 2D, não permitindo a realização do cálculo na mesma chamada à rotina. Isto determina o aumento do número de vezes que a rotina tem que ser chamada.

Na DWT e DWT^{-1} no espaço para os filtros Daubechies 9/7, registam-se os ganhos de rapidez menos significativos, porém importantes para o conjunto da cadeia de processamento. Por um lado, porque o algoritmo relativo aos filtros 9/7 é mais complicado do que o algoritmo relativo à ôndula DB2, por outro lado, porque existe uma maior dependência dos dados (dos valores finais face aos intermédios). Isto origina que os resultados do paralelismo e do agendamento na *cascata* não sejam tão bons quanto os que se obtêm para a ôndula DB2.

4.8 Conclusões

Neste capítulo foram referidos os aspectos relativos à implementação prática das rotinas para cálculo das transformadas directas de ôndulas e das transformadas inversas, pelo esquema progressivo. Fez-se a descrição da factorização dos algoritmos de cálculo pelo esquema progressivo, e apresentaram-se os resultados dos tempos de cálculo, tendo por referência a implementação num processador de uso geral, Intel Pentium III a 550 MHz.

Visando reduzir o tempo de processamento, implementaram-se as rotinas de cálculo das transformadas no módulo de avaliação do processador digital de sinal: TMS320C6201 EVM. Neste capítulo referiram-se as principais características do DSP TMS320C6201, apresentando-se os seus blocos funcionais, as ferramentas de desenvolvimento e as técnicas usadas na optimização do código, em linguagem de alto nível.

Os valores de desempenho obtidos pela implementação no DSP, que pode funcionar como co-processador, foram estabelecidos em número de ciclos de relógio, de modo a avaliar o tempo de processamento para várias versões do DSP, com diferentes frequências de relógio.

O agendamento das instruções e a dependência dos dados (no algoritmo de cálculo) condicionam o grau de paralelismo que é possível obter. Por outro lado, o número de operações matemáticas a realizar influencia directamente o tempo computacional. A conjugação destes factores, e o número de chamadas às rotinas de cálculo da DWT/DWT^{-1} permitem justificar os diferentes ganhos no desempenho obtidos para o cálculo das várias transformadas.

Com a implementação das rotinas de cálculo da DWT e da DWT^{-1} no DSP, no tempo e no espaço, é possível obter ganhos generalizados no tempo de processamento. Os ganhos obtidos no desempenho são mais significativos para a DWT/DWT^{-1} no tempo do que para a DWT/DWT^{-1} no espaço. No caso particular da DWT/DWT^{-1} no espaço, obtém-se maiores ganhos no desempenho para a ôndula Daubechies 2 do que para as ôndulas Daubechies 9/7. Tal deve-se às características do algoritmo da DWT e DWT^{-1} para a ôndula DB2, no que se refere à dependência dos dados, permitindo um melhor agendamento de instruções e um maior paralelismo das instruções para o seu cálculo.

Capítulo 5

CODEC DE VÍDEO BASEADO NA TRANSFORMADA DE ÔNDULAS

Neste capítulo descreve-se o CoDec de vídeo baseado na transformada de ôndulas, desenvolvido no âmbito desta dissertação. A eficiência do codificador é avaliada com base nos resultados experimentais, obtidos a partir da codificação de seqüências de teste. O codificador foi implementado num computador pessoal equipado com um processador de uso geral (Intel Pentium™ III a 550 MHz).

5.1 Introdução

Na codificação de imagem e vídeo com base na DCT, a imagem é fraccionada em blocos aos quais se aplica a transformada. Na codificação com base na DWT aplica-se a transformada à totalidade da imagem, evitando assim o aparecimento de efeito de bloco. Por outro lado, a organização hierárquica dos coeficientes da DWT facilita a obtenção de aproximações razoáveis às imagens, apenas pela transmissão dos coeficientes relativos às escalas mais altas, isto é, os coeficientes de aproximação (componentes de baixa frequência). Assim, as técnicas de codificação baseadas na transformada de ôndulas são adequadas para a realização de codificadores de vídeo com ritmos variáveis, nomeadamente para ritmos binários reduzidos.

Neste capítulo apresenta-se o CoDec de vídeo desenvolvido que realiza a codificação com perdas, com base na DWT, atendendo também ao tempo de processamento, que deve garantir

um ritmo mínimo de imagens por segundo, de forma a criar a ilusão de continuidade de movimento na sequência de imagens.

5.2 Estrutura do CoDec de Vídeo Baseado na DWT 3D

A figura 5.1 mostra o diagrama de blocos do codificador e do decodificador de vídeo desenvolvido no âmbito desta dissertação.

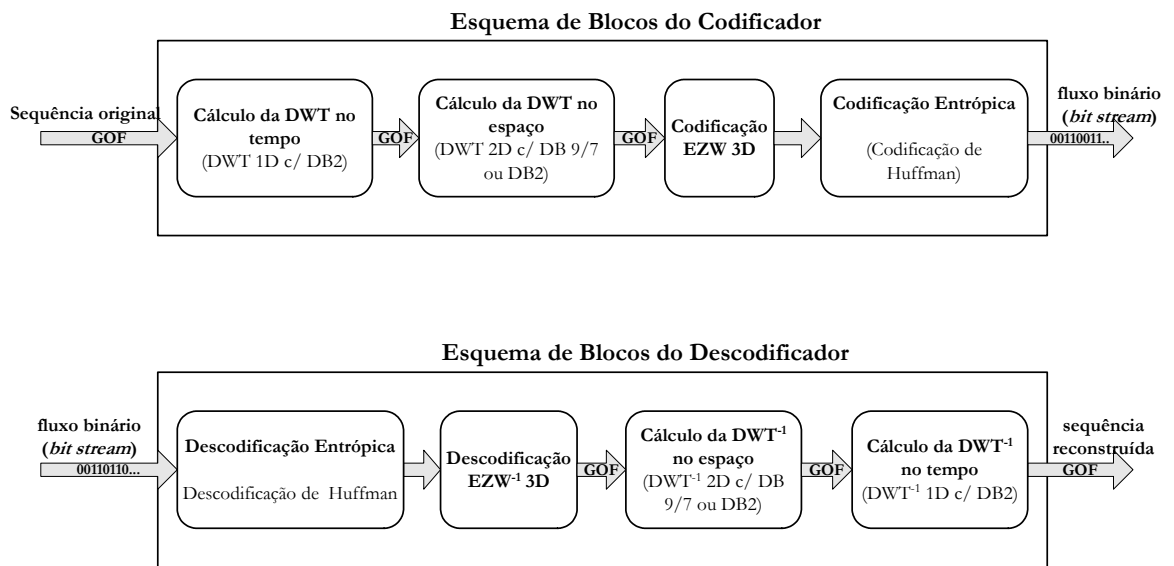


Figura 5.1: Diagrama de blocos da constituição do Codificador/Decodificador de vídeo baseado na transformada de ôndulas.

O esquema de codificação de vídeo realizado é semelhante ao esquema descrito em [Tham1+98], podendo-se apontar as seguintes diferenças: i) a utilização do esquema progressivo para cálculo da DWT; ii) a adopção de um esquema mais simples (mais eficiente) de formação das “árvores” hierárquicas 3D de coeficientes das transformadas; iii) a avaliação do desempenho de alguns blocos computacionalmente mais exigentes, utilizando um processador de sinal com arquitectura VLIW como co-processador.

No esquema da figura 5.1, a sequência de imagens original é inicialmente fraccionada em grupos de imagens, ou tramas (*frames*), ou GOF (*group of frames*). Por razões práticas de implementação, convém ter $GOF = 2^k$, com $k \in \mathbb{N}$. No CoDec realizado neste trabalho considerou-se $GOF = 4$, sendo no entanto possível considerar GOFs com um maior número de tramas, como se refere mais adiante.

Nos dois primeiros blocos do codificador explora-se a correlação temporal associada às imagens do GOF e a redundância espacial de cada trama do GOF previamente transformado no domínio do tempo. Para eliminar a redundância temporal calcula-se a DWT no tempo, relativamente às imagens do GOF, e para explorar as redundâncias espaciais, calcula-se a DWT no espaço em cada trama resultante da DWT no tempo. Em ambos os casos, o cálculo da DWT faz-se através do esquema progressivo. Os coeficientes resultantes das transformações no tempo e no espaço são de seguida codificados no bloco seguinte, através de um esquema de codificação progressivo designado EZW 3D. Neste esquema, os coeficientes das transformadas de ôndulas são organizados num esquema em árvore tridimensional, traduzindo a ordem relativa dos coeficientes, para proceder à sua quantificação. Por último, no 4º bloco aplica-se um codificador de símbolos do tipo entrópico, nomeadamente a codificação de Huffman, com base nas tabelas definidas para a codificação JPEG *Baseline* (ver apêndice C).

Nos CoDec de vídeo baseados na DCT usa-se, normalmente, estimação e compensação de movimento, considerando regiões ou blocos de tramas consecutivas. Define-se uma área de pesquisa de blocos candidatos, visando obter os vectores de movimento, que são posteriormente codificados. Normalmente usam-se algoritmos heurísticos de pesquisa [ReLi+94, Bliu+93, LaiPo+96], por questões de custo computacional.

No CoDec realizado neste trabalho não se utiliza estimação de movimento, tendo-se deixado essa opção para uma versão futura do codificador. Faz-se no entanto notar que o estimador de movimento a desenvolver tem algumas diferenças relativamente aos estimadores de movimento aplicados na codificação baseada na DCT, nomeadamente: i) a transformação no tempo; ii) a estimação de movimento deve fazer-se tomando como referência a primeira trama do GOF, estimando os vectores de movimento para cada uma das restantes tramas, não sendo calculada a diferença entre imagens [Ohm_94, Seck+01]. Desta forma explora-se melhor a correlação temporal, conseguindo-se, ainda, uma maior compactação de energia. Os vectores de movimento devem também ser posteriormente passados por um codificador do tipo entrópico, tal como o codificador de Huffman.

A estimação de movimento representa, no entanto, um acréscimo no tempo computacional. Em [Tham1+98] é utilizado um esquema de estimação e compensação de movimento diferente dos mais comuns, centrando a área de pesquisa numa região em forma de losango [Tham2+98], e que segundo [Tham1+98] permite um ganho de rapidez em cerca de 30% face ao método de pesquisa em quatro passos [LaiPo+96].

5.3 Descrição do Codificador e do Descodificador

Para a transformação do GOF no domínio do tempo efectua-se o cálculo da DWT usando a ôndula ortogonal Daubechies 2, com decomposição em 2 níveis. Para a DWT no tempo, consideram-se os *pixels* “homólogos” de cada trama do GOF, isto é, os *pixels* de cada trama com as mesmas coordenadas espaciais. Cada sequência de quatro *pixels* “homólogos” no GOF é decomposta pela DWT, em dois níveis, como se ilustra na figura 5.2, em que t referencia o nível de decomposição, $a(t_2)$ e $d(t_2)$ referenciam as tramas onde se localizam os coeficientes de aproximação e de detalhe do 2º nível, respectivamente, e $d(t_1)$ as tramas onde se localizam os coeficientes de detalhe do 1º nível. Após a transformação no tempo, o grupo de quatro *pixels* “homólogos” iniciais resulta numa sequência de quatro coeficientes, como se mostra na figura 5.3: um coeficiente de aproximação e um coeficiente de detalhe no nível mais alto (2), e dois coeficientes de detalhe no nível inferior (1). Na figura 5.3, a_{m_i} e d_{m_i} referenciam os coeficientes de aproximação e de detalhe relativos ao par de amostras de ordem m , no nível l . Usa-se a ôndula ortogonal DB2 para o cálculo da DWT no tempo, por reduzir o número de operações aritméticas requeridas, já que os filtros associados têm apenas quatro coeficientes. O algoritmo de cálculo (pelo esquema progressivo) é mais simples e requer um número reduzido de operações.

É possível usar um esquema semelhante com $\text{GOF} = 8$ e, neste caso em particular, é mais vantajoso considerar três níveis de decomposição para a DWT no tempo. Assim, se por um lado seria possível obter maiores factores de compressão, por outro resultaria num acréscimo do tempo computacional, e uma estrutura 3D com mais dependências, isto é, mais complicada na codificação EZW 3D, como se verá mais adiante.

Na transformação de cada trama do GOF no espaço, calcula-se a DWT no espaço (DWT 2D) com decomposição em três níveis, usando as ôndulas biortogonais (4,4) ou a ôndula ortogonal DB2, como se ilustra a figura 5.4. O cálculo da DWT 2D faz-se estendendo a DWT 1D a duas dimensões, por aplicação a sequências separáveis, primeiro segundo a direcção das linhas e depois segundo a direcção das colunas, como foi referido na secção 3.6. Para o cálculo da DWT no espaço usam-se as ôndulas biortogonais (4,4), porque os filtros associados a este tipo de ôndulas apresentam característica de fase linear, e por se ter concluído que são as mais adequadas para a compressão de imagem em [Anton+92, Villa+95].

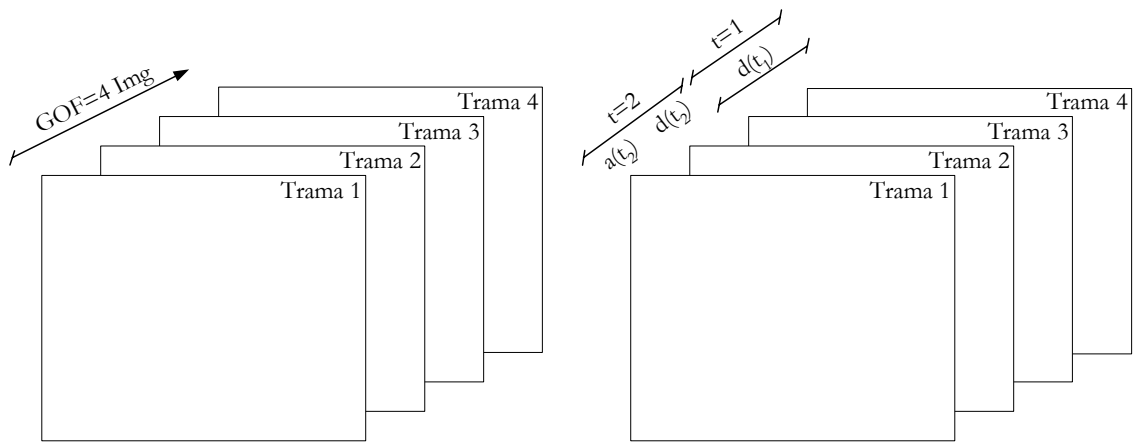


Figura 5.2: Decomposição do GOF em 2 níveis pela DWT no tempo (ôndula DB2).

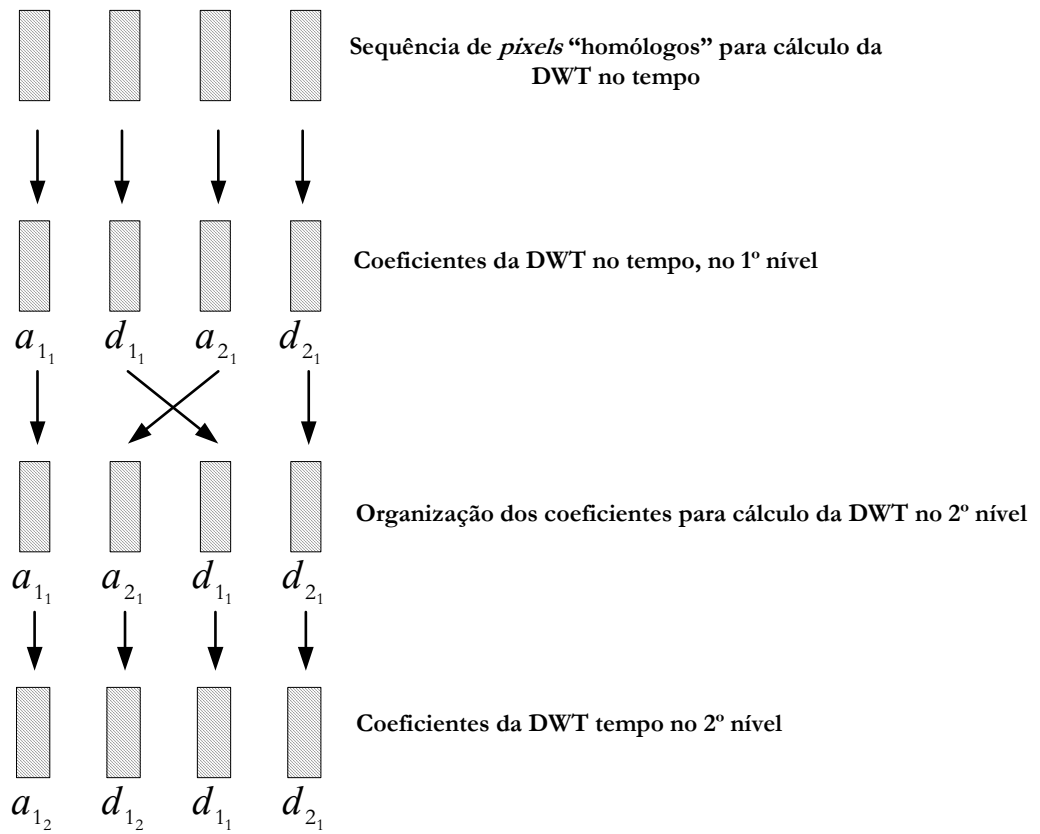


Figura 5.3: Organização dos coeficientes da sequência resultante da DWT no tempo.

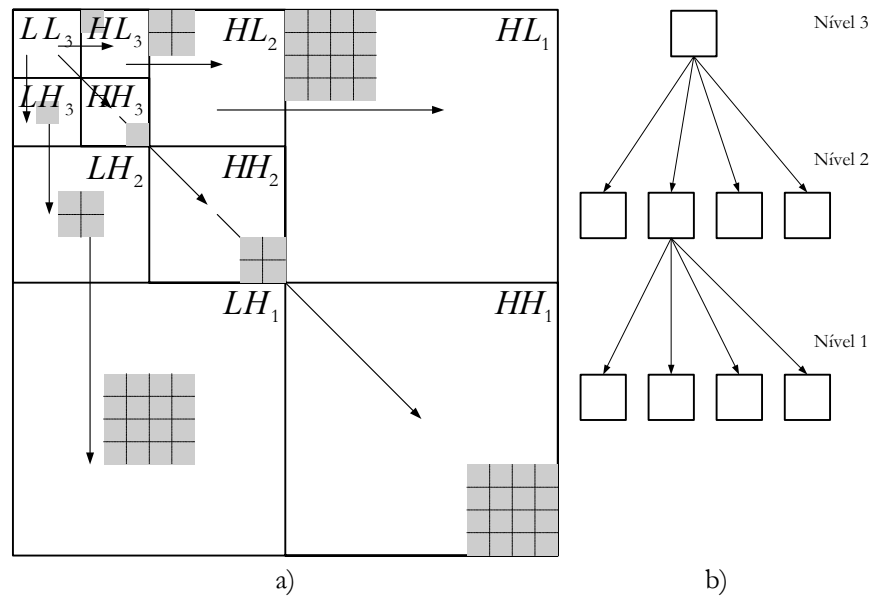


Figura 5.4: Decomposição hierárquica em 3 níveis pela DWT no espaço: a) organização em sub-bandas; b) dependências entre os coeficientes em cada nível.

Dada a necessidade de realizar o cálculo da DWT e DWT^{-1} para a ôndula DB2 no domínio do tempo, entendeu-se que haveria também interesse em estudar uma solução alternativa ao uso das ôndulas biortogonais no espaço, usando a ôndula DB2. Esta solução não é no entanto a melhor, quer relativamente aos tempos de processamento (como se viu no capítulo anterior), quer relativamente ao factor de compressão ou à qualidade das imagens recuperadas, como se verá mais adiante.

Para a DWT e DWT^{-1} no espaço é preferível o uso de ôndulas biortogonais, embora se possam usar ôndulas ortogonais. Neste último caso, convém que os respectivos filtros tenham um maior número de coeficientes, dado que nos filtros de ôndulas ortogonais de Daubechies, quanto maior fôr o número de coeficientes menor é a distorção de fase.

Refira-se que é possível o uso de qualquer ôndula ortogonal ou biortogonal para a DWT e DWT^{-1} no tempo, ou no espaço. Em qualquer dos casos, o aumento do número de coeficientes dos filtros implica um aumento do tempo de processamento.

Os processos de transformação do GOF no tempo e no espaço esultam numa estrutura 3D, cujos coeficientes estão organizados em regiões (bandas), apresentando relações de dependência hierárquica bem definidas. Esta estrutura hierárquica é posteriormente codificada através do esquema de codificação EZW 3D. A formação da estrutura 3D de coeficientes da

DWT, a sua organização hierárquica, as suas dependências, e a codificação EZW 3D são explicadas mais adiante.

De modo a obter factores de compressão mais elevados, aplica-se um codificador de Huffman aos coeficientes da transformada. Porém, no caso da codificação EZW 3D em particular, dados não serem conhecidos dados estatísticos concretos relativos à probabilidade de ocorrência dos valores a codificar, consideram-se as tabelas definidas para a codificação de Huffman na norma JPEG *Baseline* (ver anexo C). Consideram-se séries de 16 valores de coeficientes da DWT a 3D, representados por 16 *bits* no procedimento de codificação, devido à organização das tabelas. O código binário resultante da codificação de cada série tem um alinhamento de 8 *bits*, correspondendo a um octecto (um *byte*). Caso o número de *bits* resultante da codificação da série de valores não seja um múltiplo de 8, considera-se a dimensão imposta pelo múltiplo de 8 imediato, preenchendo os restantes *bits* com zeros.

O ritmo binário gerado pelo esquema de codificação descrito é escalável, permitindo adaptar as condições do CoDec à largura de banda disponível no canal de transmissão. Esta característica tem vantagens em sistemas de codificação e decodificação cujo canal de transmissão apresente características de largura de banda variantes no tempo, como é por exemplo o caso das ligações via Internet.

No decodificador, realizam-se as operações inversas das realizadas no codificador, partindo da sequência binária (*bitstream*) que representa a sequência de imagens codificadas. Aplica-se em primeiro lugar um decodificador de Huffman, usando as mesmas tabelas da codificação, de modo a reconstruir o código para a decodificação EZW⁻¹ 3D. Em seguida realiza-se a decodificação EZW⁻¹ 3D e reconstrói-se a estrutura 3D do GOF a recuperar. Os elementos das matrizes do GOF a reconstruir são previamente colocados a zero. A estrutura 3D recuperada mantém as mesmas características de dependência hierárquica dos seus coeficientes, tal como no codificador. Respeitando as relações hierárquicas entre os coeficientes, aplica-se primeiro a DWT⁻¹ no espaço para cada trama do GOF reconstruído, e a seguir a DWT⁻¹ no tempo.

De modo a garantir o sincronismo no decodificador, a trama de *bits* gerada na codificação contém a seguinte informação: i) o número de tramas codificadas; ii) o nível mínimo de decisão usado na codificação; iii) o valor do nível de decisão inicial usado na codificação do GOF; iv) os valores máximos dos coeficientes em cada trama do GOF codificado.

Usando uma representação em pseudo-código, com uma notação semelhante à linguagem de programação “C”, descrevem-se, em seguida, os algoritmos implementados para a codificação e para a decodificação.

No codificador, tem-se:

```
codif_video (video_org, n_tramas_codificar, nivel_decis_min)
{
    k=0;
    do{

        dwt_tempo(Y,U,V);           // com a ôndula DB2
        dwt_espaço (Y,U,V);         // com a ôndula DB2 ou bior(4.4)
        cod_ezw_3d(decis_min,Y,U,V);
        cod_huffman();             // gera bitstream

        k+=4;                       // Alinhamento para GOF = 4

    }while (k<n_tramas_a_codificar)
}
```

No decodificador tem-se:

```
descodif_video (bitstream) // ficheiro relativo à codificado
{
    n_tramas_descod = le_inf_sincron_bitstream();
    nivel_decis_min = le_inf_sincron_bitstream();

    k=0;
    do{
        descod_huffman(valor);
        descod_ezw_3d(nivel_decis_min,Y,U,V);
        idwt_espaço (Y,U,V);       // com a ôndula DB2 ou bior(4.4)
        idwt_tempo(Y,U,V);        // com a ôndula DB2

        k=k+4;                     // GOF = 4
    }while (k<n_tramas_descod)
}
```

5.3.1 Formação da Estrutura 3D de Coeficientes em “Árvore”

A decomposição de uma sequência de quatro *pixels* pela DWT no tempo, em 2 níveis, resulta numa estrutura hierárquica de coeficientes 1D, em que a primeira trama do GOF contém o coeficiente mais significativo (em módulo). Os coeficientes das tramas seguintes decrescem em grau de importância (significado), à medida que a ordem da trama aumenta dentro do *GOF*.

Caso não existiam diferenças entre o pixels da sequência, ou essas diferenças sejam mínimas, existe uma probabilidade elevada de que o coeficiente relativo à última posição da sequência (última trama) seja nulo, ou de valor muito baixo (em módulo).

A decomposição no espaço de cada trama filtrada no tempo, através da DWT 2D com 3 escalas de resolução, forma uma estrutura hierárquica 2D com dependências a 3 níveis. Os coeficientes no nível mais alto de decomposição, na estrutura resultante, designam-se raízes da árvore (*tree-roots*). Um coeficiente na raiz tem três coeficientes dependentes, um por cada sub-banda no nível mais alto; um coeficiente no nível intermédio tem quatro coeficientes dependentes e um coeficiente no nível mais baixo não tem coeficientes dependentes.

Na decomposição piramidal de uma imagem de dimensões $M \times N$ em p níveis, as dimensões da região dos coeficientes raiz é dada por $M_h \times N_h$, em que $M_h = \frac{M}{2^p}$ e $N_h = \frac{N}{2^p}$. Referenciando as coordenadas dos coeficientes por (m, n) , podem-se estabelecer as relações de dependência hierárquica 2D:

$$(m, n) \rightarrow \{(m + M_h, n), (m, n + N_h), (m + M_h, n + N_h)\} \quad \text{se } (m, n) \in \mathbf{R} \quad (5.1 \text{ a})$$

$$(m, n) \rightarrow \{(2m, 2n), (2m + 1, 2n), (2m, 2n + 1), (2m + 1, 2n + 1)\} \quad \text{se } (m, n) \in \mathbf{I} \quad (5.1 \text{ b})$$

$$(m, n) \rightarrow \{ \} \quad \text{se } (m, n) \in \mathbf{B} \quad (5.1 \text{ c})$$

Os conjuntos, \mathbf{R} , \mathbf{I} e \mathbf{B} referenciam o conjunto dos coeficientes no nível mais alto da árvore (coeficientes da raiz), o conjunto dos coeficientes no nível intermédio, e o conjunto dos coeficientes no nível mais baixo da árvore, respectivamente.

Na decomposição de uma imagem (ou trama) pela DWT 2D os coeficientes apresentam relações hierárquicas de dependência, por grau de importância, como se ilustra na figura 5.5. Porém, como a DWT no espaço é aplicada ao GOF previamente transformado no tempo, a conjugação dos dois processos resulta numa estrutura hierárquica 3D, com dependências em árvore, como se ilustra na figura 5.6.

Para a estrutura 3D também se podem estabelecer as relações de dependência hierárquica, considerando uma nova coordenada, l , para referenciar a posição da trama no GOF. Definindo a dimensão do GOF por L , e considerando a sua decomposição em q níveis,

então a dimensão da região relativa aos coeficientes de aproximação no nível mais alto é

$$L_h = \frac{L}{2^q}.$$

Mantendo a notação já usada para referenciar os conjuntos de nós, isto é, os conjuntos \mathbf{R} , \mathbf{I} e \mathbf{B} , e referenciando as coordenadas dos coeficientes por (l, m, n) , tem-se:

$$(l, m, n) \rightarrow \left\{ (l + L_h, m, n), (l, m + M_h, n), (l, m, n + N_h), \right. \\ \left. (l + L_h, m + M_h, n), (l + L_h, m, n + N_h), (l, m + M_h, n + N_h), \right. \\ \left. (l + L_h, m + M_h, n + N_h) \right\} \text{ se } (l, m, n) \in \mathbf{R} \quad (5.2 \text{ a})$$

$$(l, m, n) \rightarrow \left\{ (2l, 2m, 2n), (2l + 1, 2m, 2n), (2l, 2m + 1, 2n), (2l, 2m, 2n + 1), \right. \\ \left. (2l + 1, 2m + 1, 2n), (2l + 1, 2m, 2n + 1), (2l, 2m + 1, 2n + 1), \right. \\ \left. (2l + 1, 2m + 1, 2n + 1) \right\} \text{ se } (l, m, n) \in \mathbf{I} \quad (5.2 \text{ b})$$

$$(l, m, n) \rightarrow \{ \} \text{ se } (l, m, n) \in \mathbf{B} \quad (5.2 \text{ c})$$

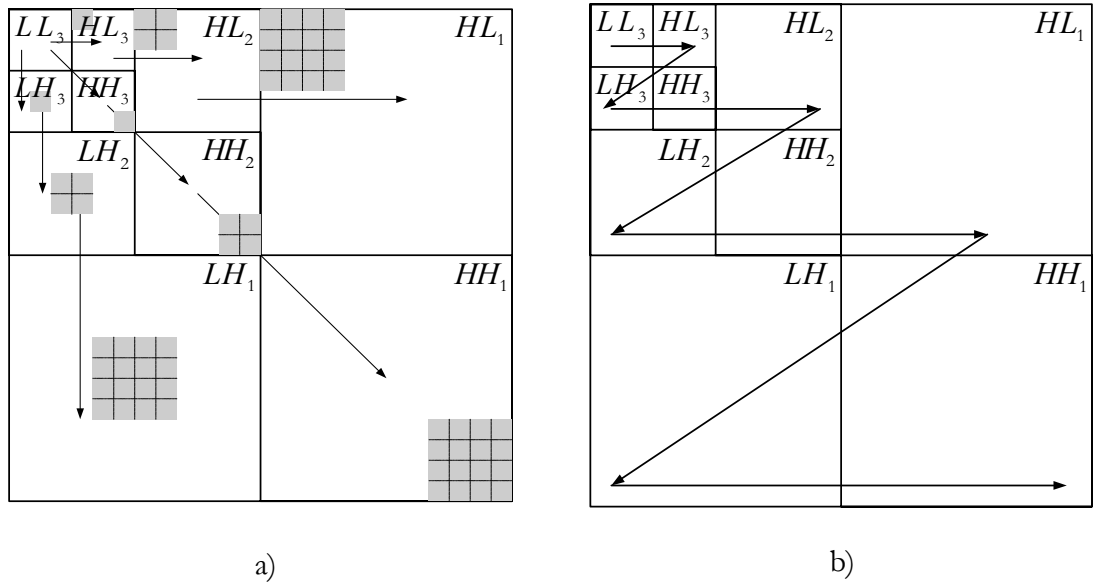


Figura 5.5: Decomposição de uma imagem pela DWT 2D, em 3 níveis: a) relação de dependência de coeficientes, b) ordem de pesquisa dos coeficientes.

Com base nas dependências indicadas nas expressões 5.2 a) 5.2 b) e 5.2 c) verifica-se o seguinte (ver figura 5.6): i) um coeficiente na raiz tem 7 coeficientes dependentes, 3 nas sub-bandas inferiores, no mesmo nível e na mesma trama, e 4 nas sub-bandas dependentes na trama seguinte (com coordenadas espaciais homólogas); ii) um coeficiente nas sub-bandas

intermédias tem 8 coeficientes dependentes, 4 no nível inferior na mesma trama e 4 no nível inferior na trama seguinte; um coeficiente do nível mais baixo não tem coeficientes dependentes.

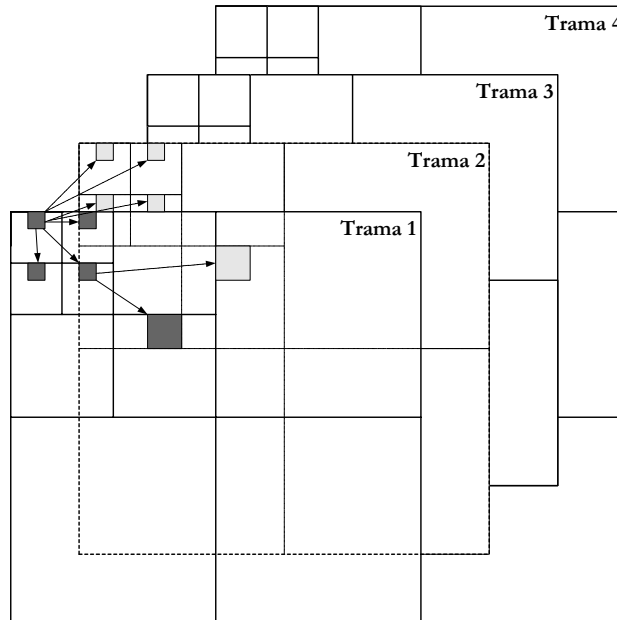


Figura 5.6: Estrutura hierárquica da DWT 3D e relações de dependência.

5.3.2 Codificação e Descodificação EZW 3D

Como foi referido no capítulo 3, a organização dos coeficientes da DWT 2D concentra a maior parte da energia associada à imagem nos níveis (escalas) mais elevados, isto é, os coeficientes mais significativos localizam-se predominantemente nas bandas de frequência mais baixas. Por outras palavras, a energia nas sub-bandas decresce à medida que a escala decresce (na escala mais baixa a resolução é mais alta, e vice-versa). Existe, assim, uma grande probabilidade que os coeficientes situados nos níveis mais baixos (frequências mais altas), sejam menos significativos face a um determinado nível de decisão. A estrutura de coeficientes é uma estrutura hierarquizada por grau de importância. Por conterem mais energia, os coeficientes das bandas de frequência mais baixas (coeficientes de maior valor) são mais importantes que os das bandas de frequência mais altas. Este facto sugere a adopção de um esquema hierárquico de codificação, enviando em primeiro lugar a informação relativa aos coeficientes mais significativos (aproximações), e de seguida a informação referente aos coeficientes menos significativos (detalhes).

O esquema de codificação EZW [Shapr_93] foi inicialmente proposto para a codificação de imagens, tendo por base uma decomposição hierárquica em sub-bandas através da transformada de ôndulas. É um esquema de codificação progressivo, isto é, a sequência de *bits* gerada no codificador (*bitstream*) contém apenas a informação necessária para se obter uma aproximação “grosseira” à imagem original, (no decodificador) sem necessidade de completar a recepção de toda a sequência de *bits* gerada na codificação. À medida que são recebidos mais *bits*, o decodificador vai refinando a qualidade da imagem, até se atingir o fim da sequência, ou até que a qualidade de imagem obtida seja satisfatória. A adoção de esquemas de codificação progressivos tem vantagens, pois permite descartar informação considerada não relevante, ou que não seja possível processar em tempo útil.

O esquema de codificação EZW pode também ser usado na compressão de sinais de vídeo, por extensão do algoritmo da codificação EZW para imagens (2D) a 3 dimensões [Chen+96]. Tal como no caso 2D, a codificação EZW 3D gera uma sequência de *bits* correspondente à codificação dos coeficientes da árvore considerados significativos, isto é, aqueles cujo módulo é maior ou igual a um dado nível de discriminação (*threshold*).

O Algoritmo da Codificação EZW 3D

A estrutura 3D de coeficientes do GOF, após aplicação da DWT no tempo e no espaço, apresenta as relações de dependência referidas nas eq. 5.2 a) a 5.2 c). Um coeficiente dessa estrutura, $x_{l,m,n}$, é considerado insignificante face a um dado nível de decisão (*threshold*) T , se:

$$|x_{l,m,n}| < T \quad (5.3)$$

O processo de codificação EZW 3D baseia-se na elevada probabilidade de localização dos coeficientes mais significativos nos níveis mais altos (bandas de frequências mais baixas), enquanto os menos significativos se situam nos níveis mais baixos (bandas de frequências mais altas). Assim, se um coeficiente numa determinada banda (ou escala) é considerado insignificante, face a um dado nível de decisão, então todos os coeficientes dependentes deste serão também considerados insignificantes (*zero-tree*).

A codificação do mapa de importância dos coeficientes requer um determinado modo de pesquisa, tal que nenhum coeficiente dependente seja codificado antes dos coeficientes hierarquicamente superiores, isto é, mais importantes. A pesquisa inicia-se nas bandas mais baixas (no nível mais alto) da trama de ordem mais baixa dentro do GOF, progredindo em

direcção às bandas mais elevadas segundo as relações de dependência estabelecidas, e termina nas bandas mais elevadas da trama de maior ordem dentro do GOF.

Definido o nível de decisão, a codificação de cada coeficiente da árvore 3D pode resultar num de quatro símbolos possíveis: raiz da árvore (**ZTR** – *Zero-Tree-Root*); zero isolado (**IZ** – *Isolated Zero*); significativo e positivo (**POS**), e significativo e negativo (**NEG**). Face ao nível de decisão, um coeficiente é considerado:

- ✓ **POS**, se for significativo (em valor absoluto) e o seu sinal positivo;
- ✓ **NEG**, se for significativo (em valor absoluto) e o seu sinal negativo;
- ✓ **ZTR**, se for insignificante, e todos os coeficientes dependentes também forem insignificantes;
- ✓ **IZ**, se for insignificante, mas existe pelo menos um coeficiente dependente que seja considerado significativo;

O algoritmo de codificação utiliza um processo de quantificação por aproximações sucessivas (SAQ), mantendo duas listas independentes de coeficientes: uma lista dominante e uma lista subordinada (*dominant list* e *subordinate list*). A lista dominante é do tipo FIFO (*first in – first out*) e está associada ao passo dominante (*dominant pass*), ou passo principal; a lista subordinada está associada ao passo subordinado (*subordinate pass*), ou passo secundário.

Usando uma representação em pseudo-código, pode-se descrever o algoritmo da codificação EZW 3D da seguinte forma:

```
EZW3D_Cod(nivel_decis_min)
{
  Ik=0; max_coef=0;

  do{
    max_coefIk = max(Ik(i,j)); //determina máximos de cada trama
    if (max_coefIk > max_coef)
      max_coef = max_coefIk; //determina maior valor (máx) GOF
    Ik++;
  }while (Ik < GOF);

  nivel_decis_inic = 2⌊log2 max_coef⌋; // determina decisão inicial
  nivel_decis = nivel_decis_inic;

  do{
    for(Ik=0; Ik<GOF, Ik++){
      if (max_coefIk >= nivel_decis)
        passo_dominante_cod(Ik, Ik+1);
    }
  }
```

```

    passo_subordinado_cod(nivel_decis/2);
    nivel_decis = nivel_decis/2;

}while (nivel_decis > nivel_decis_min);
}

```

No passo dominante procede-se a uma aproximação “grosseira” ao valor do coeficiente. No passo dominante, faz-se um varrimento dos coeficientes da trama, codificando-os de acordo com o alfabeto de símbolos considerado, isto é, **POS**, **NEG**, **IZ** ou **ZTR**. Os coeficientes codificados com **POS** ou **NEG**, isto é, cujo valor absoluto é superior ao nível de decisão corrente, são removidos da trama original; o seu valor absoluto é colocado na lista subordinada, e os coeficientes na trama correspondentes às suas coordenadas são colocados a zero, de modo a evitar que sejam novamente codificados. No passo subordinado é feito o refinamento ao valor do coeficiente codificado, usando um nível de decisão igual a metade do nível de decisão corrente.

Após o cumprimento de um passo dominante e de um passo subordinado, reduz-se o nível de decisão actual para metade, e repete-se o passo dominante e o passo subordinado, codificando os coeficientes face ao novo valor de decisão, até que todos os coeficientes tenham sido codificados, face a um dado nível de decisão mínimo.

Como anteriormente, apresentam-se também os algoritmos do passo dominante e do passo subordinado, em pseudo-código:

```

passo_dominante_cod(Ik, Ik+1)      // considerando as dependências 3D
{
    inicializa_fifo();
    while(fifo_vazia==false){
        recebe_coeficiente_codificado();
        if (codigo_coeficiente != ZTR){
            codifica_novo_coeficiente_pesquisa();
            coloca_coeficiente_codificado_na_fifo();

            if (codigo_coeficiente == POS ou ==NEG){
                adiciona_módulo_coeficiente_a_lista_subordinada();
                coloca_posição_ocupada_pelo_coeficiente_a_zero();
            }
        }
    }
}

```

```

passo_subordinado_cod(nivel_decis_subord)
{
    // nivel_decis_subord = nivel_decis_corrente/2;
do{
    if coef_lista_subord > nivel_decis_subord{
        adiciona "1" à stream_de_saida();
        coef_lista_subord = coef_lista_subord - nivel_decis_subord;
    }
    else adiciona "0" à stream_de_saida();
}while(!fim_lista_subordinada) //para_todos_elem_lst_subordinada
}

```

Varrimento dos Coeficientes na Estrutura 3D

Nos processos de codificação e decodificação, o método de varrimento dos coeficientes é pré-definido, de modo que o varrimento dos coeficientes das sub-bandas de frequências mais baixas (hierarquicamente superiores) se faz antes do varrimento às sub-bandas de frequências mais altas (hierarquicamente dependentes). Em [Shapr_93] usa-se o varrimento no modo *raster* em cada sub-banda, sendo a lista do passo subordinado ordenada de tal modo que os coeficientes mais significativos são sempre enviados em primeiro lugar.

Em [Algaz+97] são estudados vários métodos de pesquisa e a sua influência no desempenho final do codificador, entre os quais se destacam o varrimento em modo *raster* e o varrimento de Morton. Porém, de acordo com [Algaz+97], o varrimento em modo *raster* utilizado em [Shapr_93] resulta num algoritmo mais complicado, e num conseqüente aumento do custo computacional. Por isso, no CoDec realizado no âmbito deste trabalho usa-se o varrimento no modo de Morton.

Para cada nível de decisão, o varrimento faz-se da primeira para a última trama (no GOF), por cada par de tramas contíguas, explorando-se as dependências dos coeficientes nas sub bandas na trama de menor ordem (no GOF), e desta para a trama seguinte. Porém, para os pares de tramas seguintes, o varrimento só se faz caso o valor máximo absoluto dos seus coeficientes seja considerado significativo face ao nível de decisão corrente.

O Algoritmo da Decodificação EZW⁻¹ 3D

O algoritmo da decodificação EZW⁻¹ 3D é idêntico ao de codificação, isto é, a partir dos níveis de decisão aplicam-se os passos dominantes e os passos subordinados. Além disso, os coeficientes lidos da *stream* codificada são colocados em matrizes de reconstrução segundo a

mesma sequência estabelecida na ordem de varrimento do codificador. Para obter a adequada recuperação dos valores dos coeficientes, os coeficientes das matrizes de reconstrução são previamente colocados a zero. O decodificador recebe também informação sobre o nível de decisão usado e o nível máximo dos coeficientes em cada trama do GOF. O algoritmo da descodificação EZW⁻¹ 3D, pode ser descrito da seguinte forma:

```
EZW3D_Decode(nivel_decis,nivel_decis_min,maxcoef[GOF])
{
  Ik=0; max_coef=0;
  do{
    for(Ik=0; Ik<GOF, Ik++){
      max_coefIk =maxcoef[Ik];
      if((max_coefIk >=nivel_decis)&&(nivel_decis
>=nivel_decis_min))
        passo_dominante_dec(Ik, Ik+1);
    }
    passo_subordinado_dec(nivel_decis/2);
    nivel_decis = nivel_decis/2;
  }while (nivel_decis > nivel_decis_min);
}
```

```
passo_dominante_dec(Ik, Ik+1)
{
  inicializa_fifo();
  while(fifo_vazia==false){
    le_elemento_bitstream ();
    if (elem_bitstream != ZTR){
      adiciona_elemento_fifo();
      if (coeficiente == POS ou ==NEG)
        coloca_coeficiente_em_posição_na_matriz();
    }
  }
}
```

```
passo_subordinado_dec(nivel_decis_subord)
{
  //nivel_decis_subord = nivel_decis_corrente/2;
  le_elemento_lista_subordinada();
  do{
    bit = le_bitstream();
    if ((bit==1) && (elemento_matriz==POS))
      coef_na_matriz += nivel_decis_subord;
    else if ((bit==1) && (elemento_matriz==NEG))
      coef_na_matriz -= nivel_decis_subord;
  }while(!fim_lista_subordinada) //para_todos_elem_lst_subordinada
}
```

5.4 Aplicação Desenvolvida para a Codificação e Descodificação de Vídeo

Para a codificação e descodificação de vídeo com base na transformada de ôndulas, desenvolveu-se uma aplicação para ambiente Windows®, utilizando as ferramentas de programação Borland C++ Builder™ v5.0 [Holl+00]. Programou-se um conjunto de funções e rotinas que implementam o CoDec de vídeo baseado nas transformadas de ôndulas, e uma interface gráfica com o utilizador, cujo aspecto se apresenta na figura 5.7.

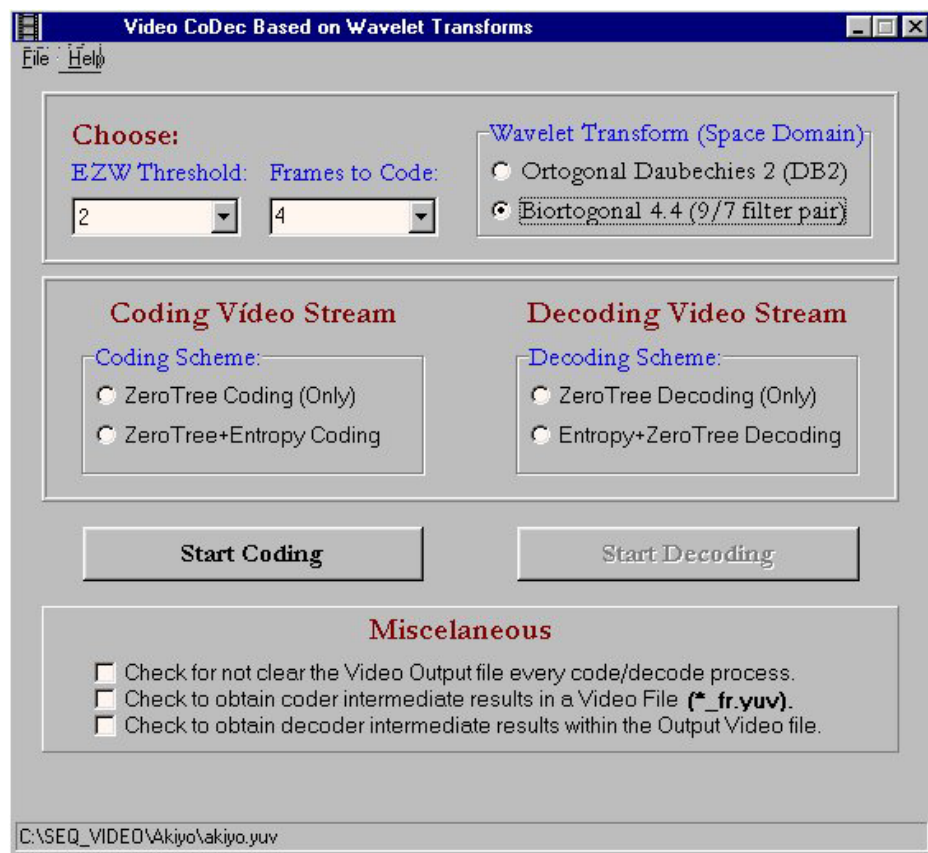


Figura 5.7: Aspecto da interface gráfica da aplicação desenvolvida para codificação e descodificação de vídeo baseada nas transformadas de ôndulas.

Através desta interface gráfica, o utilizador decide qual a sequência de vídeo a codificar e os parâmetros a considerar na codificação, tais como, o nível de decisão, o número de tramas a codificar e a ôndula a usar para a transformada no domínio do espaço. A entrada e saída de dados é feita, nesta versão da aplicação, apenas através de ficheiros armazenados em memória de massa. O ficheiro contendo a sequência de vídeo pode estar localizado em qualquer

directório, sendo a sua localização e abertura facilitadas através de uma janela de diálogo (ver figura 5.8), activada a partir do menu principal. Uma vez aberto o ficheiro que contém a sequência de vídeo (com extensão *.yuv), o seu nome e localização aparecem referenciados na região inferior da aplicação. O acesso aos ficheiros codificados também se faz através de uma janela de diálogo, mas neste caso os ficheiros têm extensão *.ztr.

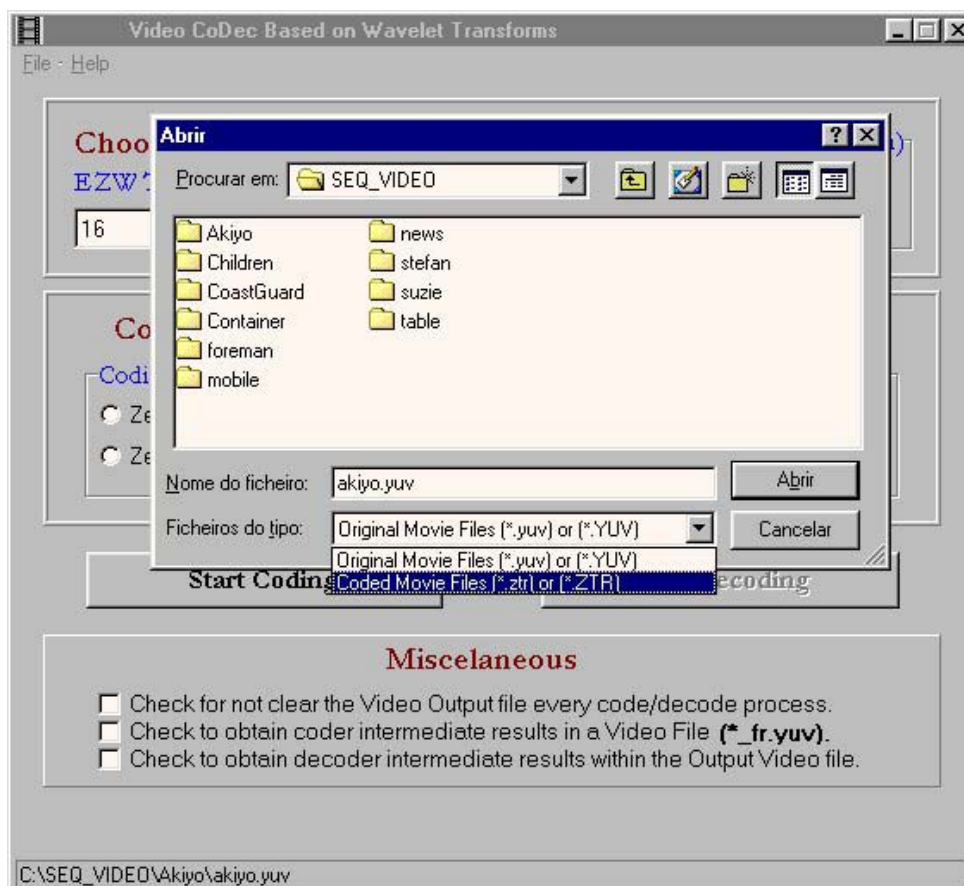


Figura 5.8: Aspecto da janela de diálogo para selecção dos ficheiros a codificar ou a decodificar.

Por defeito, o acesso aos botões para comando da codificação e da decodificação é bloqueado. Este acesso apenas é permitido após a abertura bem sucedida dos ficheiros. Caso o ficheiro tenha extensão “.yuv” apenas é permitido acesso ao botão de comando para codificação, mantendo-se o comando do botão para decodificação bloqueado; caso o ficheiro aberto tenha extensão “.ztr”, verifica-se a situação inversa.

A atribuição de valores aos parâmetros de codificação é feita de forma simples. Os valores do nível de decisão e do número de tramas a codificar são seleccionados entre um conjunto de valores pré-definidos, bastando ao utilizador fazer deslizar o cursor e activar os valores pretendidos, como se apresenta na figura 5.9. Dada a dimensão do GOF, os valores do

número de tramas a codificar são sempre múltiplos de 4. A escolha dos restantes parâmetros faz-se também numa forma simples, através da selecção por simples marcação.

Na descodificação, o utilizador apenas tem que seleccionar o ficheiro a considerar na descodificação (*.ztr), e o tipo de descodificação pretendida, seleccionando apenas entre a descodificação EZW 3D, ou EZW 3D seguida de codificação de Huffman.

De modo a facilitar o estudo do processo de codificação, disponibilizam-se ainda um conjunto de possibilidades (miscelânea) que permitem obter informação visual acerca das fases intermédias do processo de codificação, e que foram utilizadas, por exemplo para registar os resultados das transformadas no capítulo 4 apresentadas nas figuras 4.2 a 4.4.

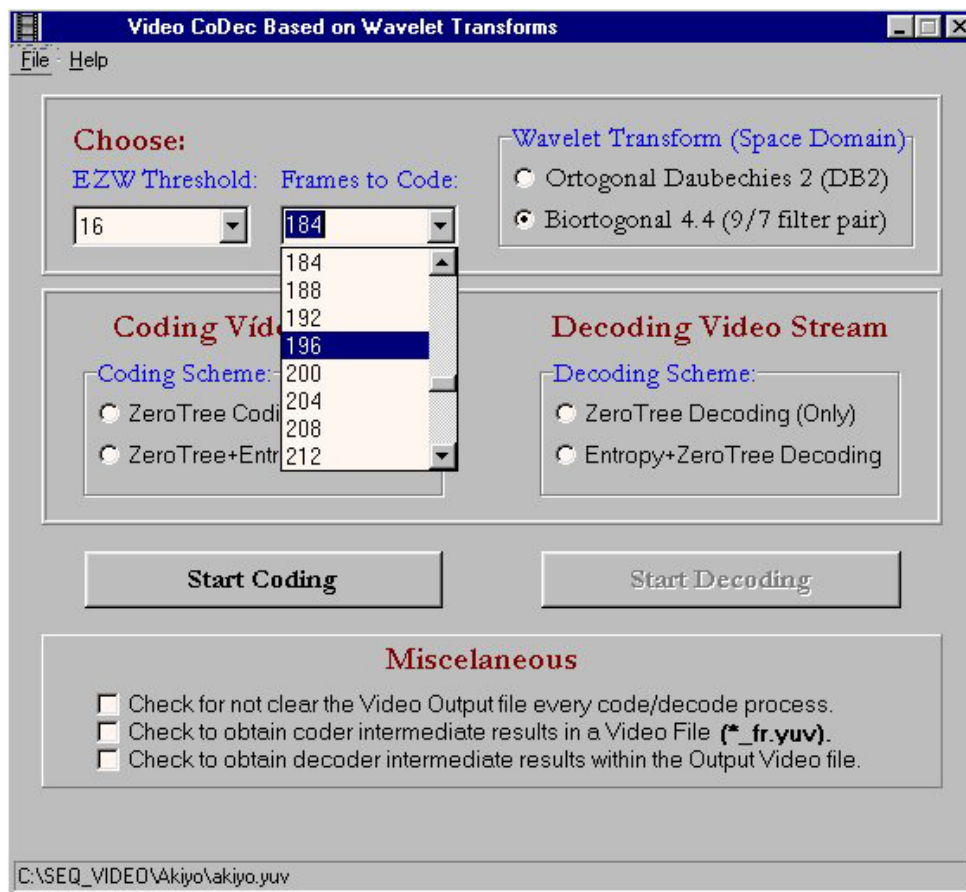


Figura 5.9: Aspecto da aplicação: selecção dos parâmetros a considerar na codificação.

5.5 Avaliação do Desempenho do CoDec de Vídeo

Para a avaliar o desempenho do CoDec de vídeo baseado na transformada de ôndulas, consideram-se sequências de vídeo em formato QCIF (144×176), com subamostragem espacial 4:2:0, 8 bits por *pixel* para representar a componente de luminância, armazenadas em ficheiros (em binário) *.yuv (ITU-R 601). Estas sequências de vídeo são normalmente usadas nos testes de codificadores e decodificadores, nomeadamente na avaliação de codificadores do tipo H.263 e MPEG. De acordo com [MPEG_01], podem-se enquadrar as sequências de vídeo em classes entre A e E (ver tabela 5.1). Na tabela 5.1 apresentam-se ainda as sequências das diferentes classes que foram usadas para avaliar o desempenho do CoDec desenvolvido.

Classe	Características	Sequências Consideradas
A	Detalhe espacial reduzido e quantidade de movimento reduzida	“Akiyo”, “Container Ship”
B	Detalhe espacial médio e quantidade de movimento reduzida, ou vice-versa	“Foreman”, “News”
C	Elevado detalhe espacial e quantidade de movimento média, ou vice-versa	“Table Tennis”, “Mobile & Calendar”
E	Imagens híbridas: objectos naturais e sintéticos	“Children”

Tabela 5.1: Sequências de vídeo para teste: classificação, características, e sequências consideradas

No CoDec de vídeo, para o processamento das sequências considera-se 3 sub-GOFs, um para as componentes de luminância e os restantes dois para as componentes de crominância, sendo cada sub-GOF constituído por 4 matrizes ($GOF=4$). Portanto, o GOF tem uma dimensão fixa, e cada uma das sequências para processamento tem o mesmo número de *bits*. Deste modo, quer na codificação, quer na descodificação, o valor máximo possível de coeficientes da DWT a 3D é idêntico, qualquer que seja a sequência considerada, tanto para a DWT no tempo como para a DWT no espaço. Além disso, a formação da estrutura da DWT 3D é a mesma, independentemente da classificação da sequência a usar, e o mesmo se verifica para a DWT^{-1} 3D. Os valores do tempo computacional associados ao cálculo da DWT e DWT^{-1} , no tempo e no espaço foram apresentados no capítulo anterior, tanto para o processador de uso geral, como para o DSP TMS320C6201.

Em seguida apresentam-se os valores dos factores de compressão obtidas para os vários níveis de decisão, para as duas ôndulas utilizadas para a transformada no domínio do espaço, considerando apenas a codificação EZW 3D, ou utilizando esta codificação em conjunto com a codificação de Huffman. Apresentam-se também os tempos médios de processamento para o cálculo da EZW 3D e da EZW⁻¹ 3D, cujos valores dependem fortemente do tipo de sequência, da informação nela contida e do nível de decisão considerado. Nos tempos de processamento que se apresentam não se consideram os tempos gastos na codificação de Huffman, por ser um tipo de codificação usualmente aplicada e à qual correspondem tempos de processamento reduzidos. De modo a avaliar a qualidade das imagens recuperadas, apresentam-se os valores da relação sinal/ruído de pico (PSNR) para as sequências de teste consideradas, e algumas imagens originais e recuperadas a partir das sequências codificadas.

5.5.1 Factor de Compressão

A quantidade de informação gerada à saída do codificador EZW 3D depende não só da informação contida na sequência, mas também do nível de decisão considerado, reflectindo-se directamente na dimensão dos ficheiros codificados.

Para cada uma das sequências originais consideradas (QCIF), codificaram-se 200 tramas. O factor de compressão foi calculado como sendo o quociente entre a dimensão da sequência original (S_o) e da sequência codificada (S_c) (eq. 2.20), isto é, $\gamma = \frac{S_o}{S_c}$. A dimensão do ficheiro original correspondente às 200 tramas é de 7.603.200 bytes ($200 \times [(144 \times 176) + 2 \times (72 \times 88)] = 7\,603\,200$).

A seguir apresentam-se os factores de compressão obtidos com a codificação das sequências de teste consideradas. De modo a melhor avaliar os valores do factor de compressão, consideram-se as seguintes situações de codificação: o uso das duas transformadas de ôndulas diferentes no domínio do espaço e o uso de vários níveis de decisão. Consideram-se ainda as situações de codificação EZW 3D (apenas), e de codificação EZW 3D seguida de codificação de Huffman. Os valores obtidos são apresentados nas tabelas 5.2 a 5.8.

Akiyo

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	1.084.131	7,01	352.233	21,59
	32	755.827	10,06	262.386	28,98
	64	534.692	14,22	192.707	39,45
	128	383.934	19,80	118.559	64,13
DB2	16	1.241.411	6,12	395.857	19,21
	32	856.197	8,88	357.663	21,26
	64	584.912	13,00	186.211	40,83

Tabela 5.2: Factor de compressão para a sequência “Akiyo” (200 tramas).

Container Ship

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	1.287.576	5,91	403.476	18,84
	32	875.797	8,68	159.677	47,62
	64	604.578	12,58	231.889	32,79
	128	416.516	18,25	107.576	70,68
DB2	16	1.413.489	5,38	331.642	22,93
	32	966.771	7,86	292.823	25,97
	64	660.155	11,52	139.081	54,67

Tabela 5.3: Factor de compressão para a sequência "Container Ship" (200 tramas).

Foreman

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	1.439.521	5,28	347.275	21,89
	32	938.416	8,10	242.611	31,34
	64	624.511	12,17	245.109	31,02
	128	426.771	17,82	137.412	55,33
DB2	16	1.674.576	4,54	499.835	15,21
	32	1.082.437	7,02	320.202	23,75
	64	702.276	10,83	187.937	40,46

Tabela 5.4: Factor de compressão para a sequência “Foreman” (200 tramas).

News

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	1.420.449	5,35	464.677	16,36
	32	951.013	7,99	290.040	26,21
	64	632.735	12,02	170.000	44,72
	128	420.943	18,06	111.285	68,32
DB2	16	1.577.492	4,82	438.366	17,34
	32	1.046.807	7,26	256.187	29,68
	64	688.992	11,04	277.210	27,43

Tabela 5.5: Factor de compressão para a sequência "News" (200 tramas).

Table Tennis

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	1.519.997	5,00	473.211	16,07
	32	1.008.592	7,54	233.277	32,59
	64	671.853	11,32	155.224	48,98
	128	449.382	16,92	134.520	56,52
DB2	16	1.686.042	4,51	459.525	16,55
	32	1.119.388	6,79	324.923	23,40
	64	732.704	10,38	235.759	32,25

Tabela 5.6: Factor de compressão para a sequência "Table Tennis" (200 tramas).

Mobile & Calendar

	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	2.561.143	2,97	885.924	8,58
	32	1.504.602	5,05	497.209	15,29
	64	837.243	9,08	192.830	39,43
	128	470.963	16,14	127.857	59,47
DB2	16	2.789.142	2,73	1.049.087	7,25
	32	1.645.260	4,62	554.781	13,70
	64	908.717	8,37	223.927	33,95

Tabela 5.7: Factor de compressão para a sequência "Mobile & Calendar" (200 tramas).

Children					
	Nível decisão	EZW 3D [bytes]	$\gamma = \frac{S_o}{S_c}$	EZW 3D+Huff [bytes]	$\gamma = \frac{S_o}{S_c}$
DB 9/7	16	2.073.089	3,67	717.701	10,59
	32	1.293.510	5,88	383.346	19,83
	64	788.293	9,65	224.110	33,93
	128	484.669	15,69	89.492	84,96
DB2	16	2.344.972	3,24	499.268	15,23
	32	1.453.786	5,23	401.604	18,93
	64	866.731	8,77	263.197	28,89

Tabela 5.8: Factor de compressão para a sequência "Children" (200 tramas).

Para a construção das tabelas 5.2 a 5.8 apenas se consideraram níveis de decisão 2^n ($4 \leq n \leq 7$), para a DWT no domínio do espaço com os filtros 9/7, e níveis 2^n ($4 \leq n \leq 6$) para a DWT no domínio do espaço com a ôndula DB2. A qualidade subjectiva das imagens que se obtêm com o nível de decisão 8 é semelhante à qualidade que se obtêm com o nível 16, porém, a quantidade de informação gerada à saída do codificador é maior. Por este motivo, considera-se o nível mínimo para o esquema de codificação como sendo 16.

A degradação das imagens aumenta com o aumento do nível de decisão usado na codificação, como seria de esperar, piorando a qualidade subjectiva das imagens. Para o nível de decisão 128 nota-se já uma considerável degradação das imagens, que é muito mais acentuada para a DWT no espaço com a ôndula DB2 do que para a DB9/7. No caso da DWT no espaço com os filtros 9/7, apesar da degradação, as imagens mantêm alguma inteligibilidade, sendo possível identificar os objectos de maiores dimensões. Por este motivo considera-se o nível 128 como nível máximo aceitável para a DWT no espaço com os filtros 9/7, enquanto que para a DWT no espaço com a ôndula DB2 esse valor máximo é 64.

As diferenças que se registam na qualidade subjectiva das imagens, em função da ôndula usada, reflectem-se também na quantidade de informação gerada à saída do codificador. De um modo geral, para o mesmo nível de decisão, o factor de compressão que se obtêm considerando a DWT no espaço para a ôndula DB2 é menor do que o que se obtêm no outro caso. Para a ôndula DB2, há coeficientes de valor significativo (na codificação EZW) nas sub-bandas de maior frequência (níveis inferiores), em particular nas regiões de transição entre os níveis de decomposição, sendo este efeito mais significativo para os níveis de decisão mais baixos. Na codificação EZW, isto não favorece a atribuição do símbolo **ZTR** (ver 5.3.2) aos coeficientes

hierarquicamente superiores. Isto reflecte as melhores características das ôndulas biortogonais na descორrelação de sinais, em particular de imagens, e na consequente compactação de energia.

Apesar das tabelas 5.2 a 5.8 não evidenciarem, pode-se referir que para as sequências de teste, para vários níveis de decisão, e para as duas transformadas no domínio do espaço, a *bitstream* gerada para o sub-GOF relativo às luminâncias representa entre 70% a 85% da informação referente ao GOF total.

Os gráficos das figuras 5.10 a 5.13 permitem comparar os resultados obtidos para diferentes condições de codificação, e analisar a influência dos parâmetros pertinentes no processo de codificação. Ao eixo do x corresponde uma escala logarítmica, considerando-se o logaritmo na base 2. Utilizando apenas a codificação EZW 3D (ver figuras 5.10 e 5.11), o factor de compressão regista uma evolução monotónica, aproximadamente linear, para cada uma das sequências e para os dois casos da transformada no espaço. Existe uma relação de proporcionalidade directa (quase constante) entre o factor de compressão e o logaritmo na base 2 do nível de decisão. Conforme foi anteriormente referido, para a DWT 9/7 no espaço obtêm-se maiores factores de compressão do que para a ôndula DB2.

Na codificação EZW 3D seguida da codificação de Huffman regista-se, em geral, também uma evolução semelhante (ver figuras 5.12 e 5.13), exceptuando para as sequências “Container Ship” e “Foreman” quando codificadas com a DWT 9/7 e para a sequência “News”, no caso da ôndula DB2. Nestes casos a evolução não é monotónica, registando-se uma inflexão da curva para o nível de decisão 64, que se deve às características da *bitstream* gerada no codificador e, também, ao facto do conjunto de tabelas usadas na codificação de Huffman não estarem completamente adaptadas para o esquema de codificação utilizado. Porém, e apesar disso, é possível obter ganhos no factor de compressão face à codificação EZW 3D, entre, aproximadamente, 2 e 5 (ver figuras 5.14 a 5.17). Nos gráficos das figuras 5.10 a 5.13, verifica-se ainda que se obtêm factores de compressão mais elevados para as sequências com baixo ou médio detalhe espacial e quantidade de movimento reduzida, isto é, à medida que se avança das sequências da classe E para a classe A (ver tabela 5.1).

Relativamente às sequências originais, a codificação EZW 3D seguida de codificação de Huffman permite obter factores de compressão entre 10 e 50, com uma degradação qualitativa média na qualidade das imagens, e entre 55 e 85, com maior degradação para o nível 128 (apenas no caso da DWT 9/7 no espaço). A obtenção de maiores factores de compressão passa pela adopção de esquemas de estimação de movimento, em conjugação com a

codificação EZW 3D seguida de codificação entrópica, e que se indica como possível trabalho futuro. As tabelas relativas à codificação entrópica também devem ser adequadas às características da codificação.

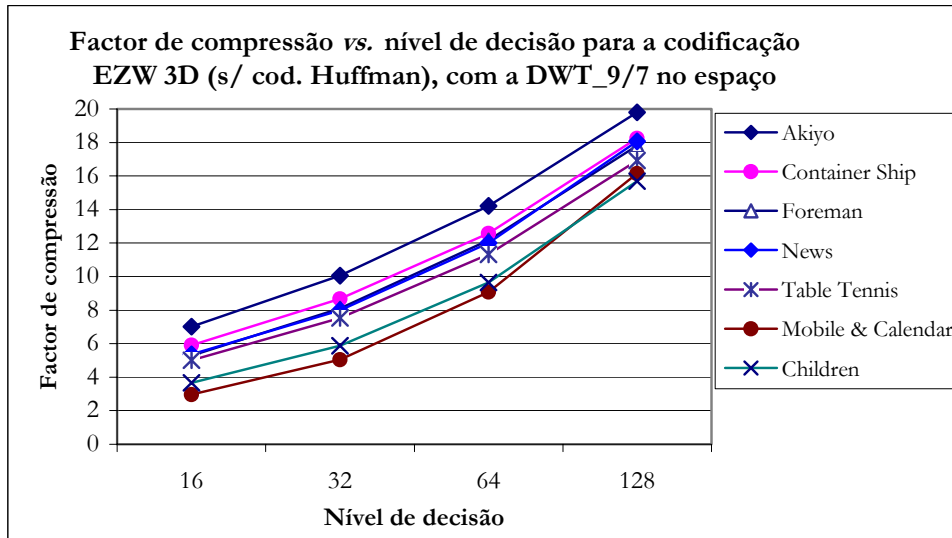


Figura 5.10: Variação do factor de compressão por codificação EZW 3D em função do nível de decisão, com a DWT 9/7 no espaço.

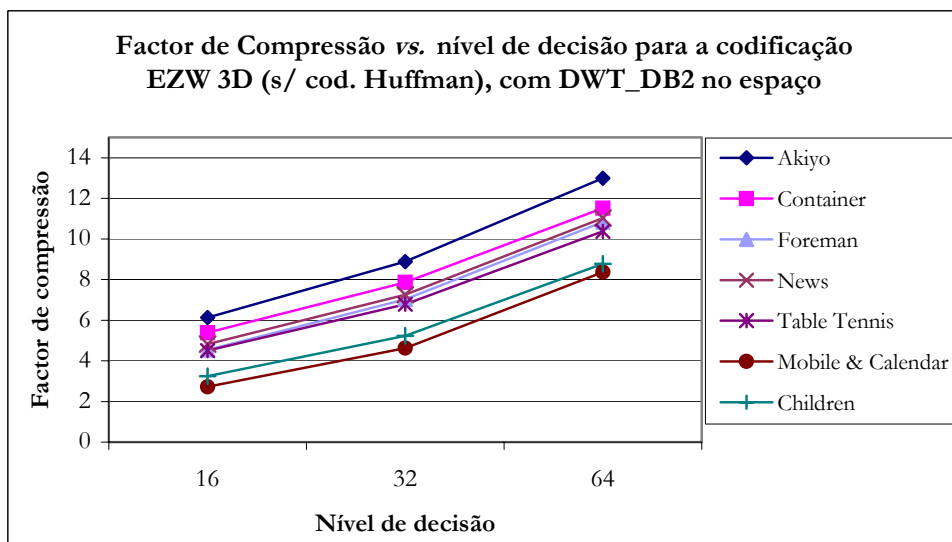


Figura 5.11: Variação do factor de compressão por codificação EZW 3D em função do nível de decisão, com a DWT DB2 no espaço.

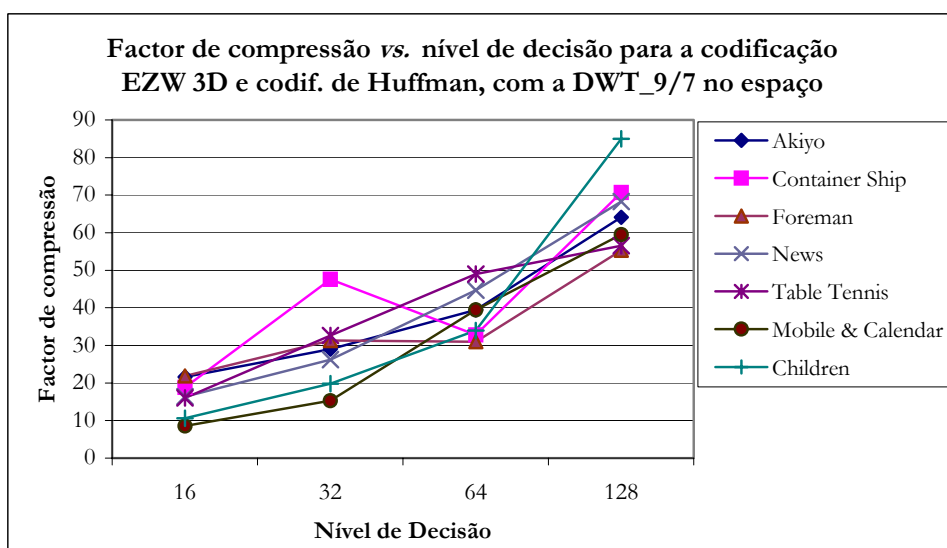


Figura 5.12: Variação do factor de compressão por codificação EZW 3D e codificação de Huffman, em função do nível de decisão, com a DWT 9/7 no espaço.

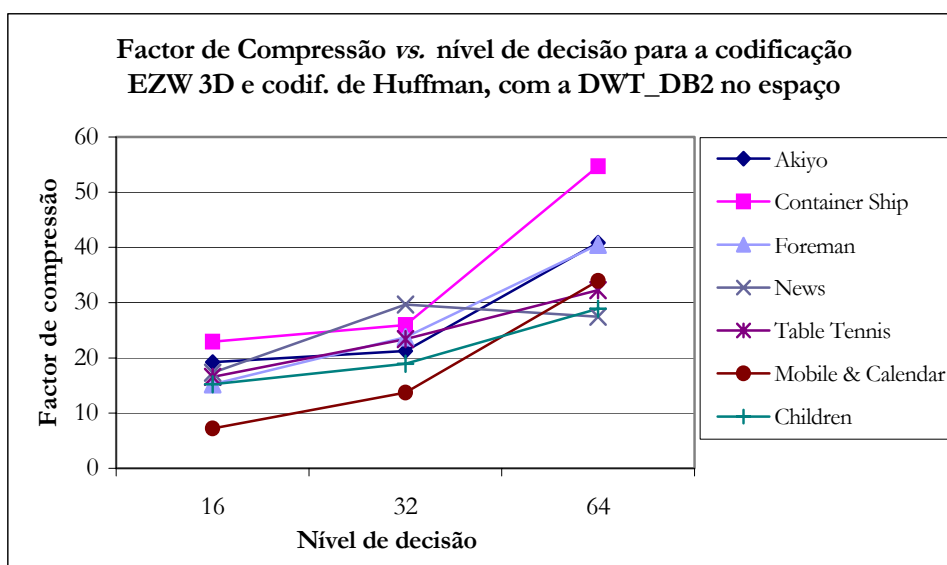


Figura 5.13: Variação do factor de compressão por codificação EZW 3D e codificação de Huffman, em função do nível de decisão, com a DWT DB2 no espaço.

Nos gráficos das figuras 5.14 a 5.17, apresentam-se as variações do factor de compressão, para a codificação EZW 3D seguido de codificação de Huffman, para as duas situações da DWT no espaço, considerando uma sequência de cada uma das classes referidas. Pelos gráficos das figuras 5.14 a 5.17, verifica-se que o factor de compressão que se obtém com a DWT 9/7 no domínio do espaço, em geral, é maior do que o que se obtém com a DWT DB2. A utilização da DWT 9/7 permite um ganho no factor de compressão entre 12% e 44% relativamente à

DWT DB2, exceptuando para as seqüências “Akiyo” e “Foreman”, com o nível de decisão 64, e para a seqüência “Children”, com o nível de decisão 16.

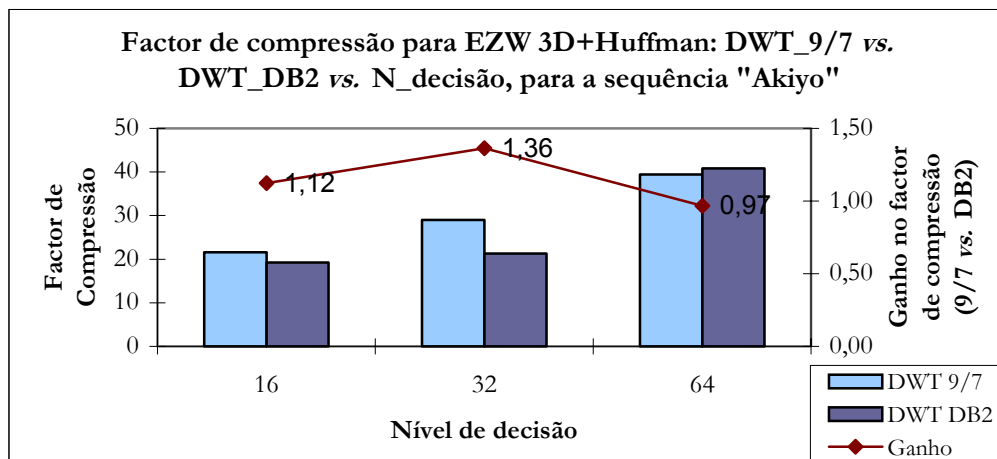


Figura 5.14: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a seqüência "Akiyo" (classe A).

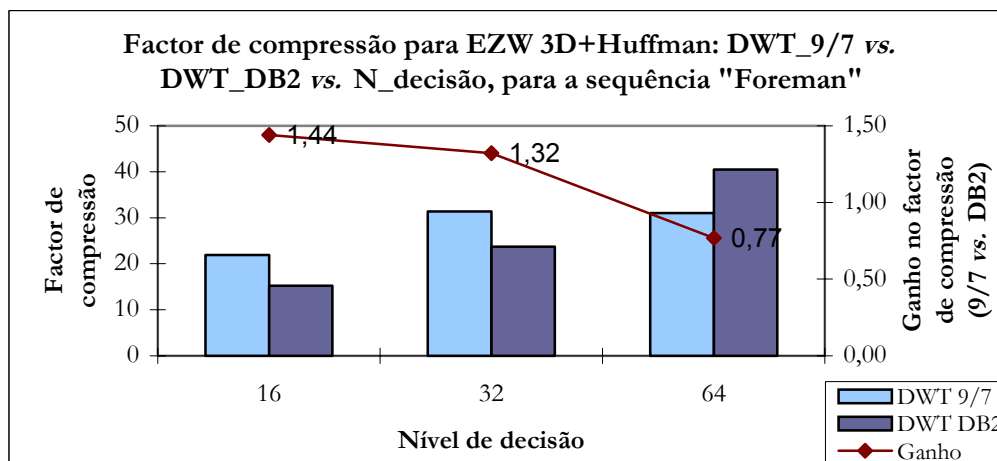


Figura 5.15: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a seqüência "Foreman" (classe B).

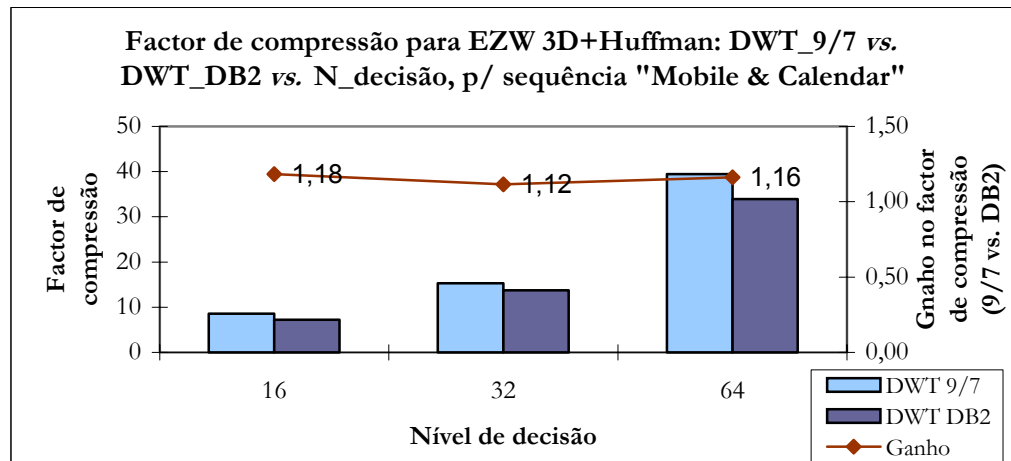


Figura 5.16: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Mobile & Calendar" (classe C).

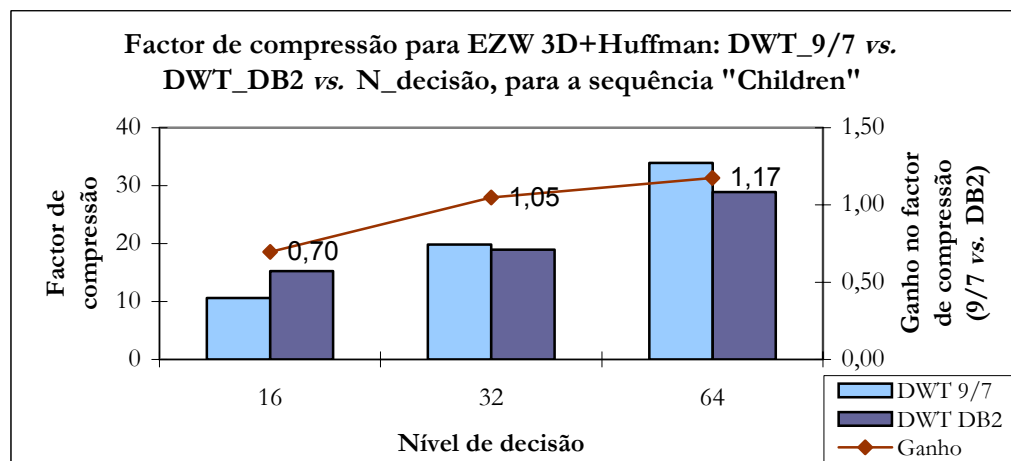


Figura 5.17: Ganho no factor de compressão para a codificação EZW 3D + Huffman, da DWT 9/7 em relação à DWT DB2, para a sequência "Children" (classe E).

Através das tabelas 5.2 a 5.8, para os mesmos valores do nível de decisão, apenas para a codificação EZW 3D, verifica-se que o factor de compressão que se obtém com a DWT 9/7 é sempre maior do que o obtido com a DWT DB2. Porém, tal não se verifica sempre, quando aplicada em conjunto com a codificação de Huffman, como se observa nos gráficos das figuras 5.14 a 5.17. Isto deve-se ao facto das tabelas da codificação de Huffman não se encontrarem completamente adaptadas às características do codificador, como foi referido anteriormente. É de esperar que, com a adequada adaptação dessas tabelas, se obtenham, também, sempre melhores resultados no factor de compressão com a DWT 9/7.

5.5.2 Avaliação da Qualidade das Imagens Recuperadas

Na avaliação da qualidade das imagens recuperadas usam-se, geralmente, critérios objectivos e subjectivos. Nos critérios objectivos incluem-se os critérios descritos no capítulo 2, nomeadamente a relação sinal/ruído (eq. 2.22) e a relação sinal/ruído de pico PSNR (eq. 2.23), definida por:

$$PSNR = 10 \cdot \log_{10} \frac{A^2}{MSE} \quad (\text{dB})$$

A representa a amplitude máxima do *pixel* ($A = 255$ em sistemas onde o *pixel* é representado por 8 *bits*), e MSE representa o erro quadrático médio, assim definido:

$$MSE = \frac{\sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I_l(m,n) - I'_l(m,n)]^2}{L \cdot M \cdot N}$$

Em que $I_l(m,n)$ representa o *pixel* da posição (m,n) da imagem l na sequência original, e $I'_l(m,n)$ representa o *pixel* da posição (m,n) da imagem l na sequência reconstruída.

Neste trabalho, a avaliação da qualidade das imagens recuperadas faz-se através dos valores de PSNR, considerando as componentes de luminância das sequências originais e das sequências recuperadas, e pela avaliação subjectiva da degradação, por comparação dos dois conjuntos de imagens.

Determinou-se a relação sinal ruído de pico para uma sequência de cada uma das classes referidas, considerando 88 tramas. Os valores de PSNR que se obtêm variam com os parâmetros da codificação, nomeadamente com a ôndula usada na DWT no domínio do espaço, e com as características das sequências codificadas.

Nas figuras 5.18 a 5.25 apresentam-se os gráficos de PSNR para as sequências “Akiyo”, “News”, “Mobile & Calendar” e “Children”, para os casos da DWT 9/7 e da DWT DB2.

Existe uma relação de proporcionalidade inversa entre os valores de PSNR e o nível de decisão, representado numa escala logaritmica (logaritmo na base 2). Com o aumento do nível de decisão para o dobro, verifica-se uma degradação na PSNR de todas as tramas do GOF de aproximadamente 4 dB. Em todas as sequências consideradas, e para todos os níveis de decisão, verifica-se que as curvas de PSNR apresentam um comportamento periódico, de período igual à dimensão do GOF ($N = 4$), sendo o valor de PSNR para a primeira trama de

cada GOF mais baixo do que para as restantes tramas do GOF, para um mesmo nível de decisão, o que se deve a erros introduzidos na codificação. Mais ainda, verifica-se que a degradação da primeira trama é sempre mais significativa com a DWT 9/7 do que com a DWT DB2. Isto tem a ver com o facto de, para a mesma sequência de valores, os coeficientes que se obtêm com a ôndula 9/7, na região correspondente às sub-banda de baixas frequências (coeficientes de aproximação), apresentam menor valor do que os que se obtêm para as mesmas regiões com a ôndula Daubechies 2. Por outro lado, para os coeficientes de detalhe verifica-se o inverso. Assim, para o mesmo nível de decisão, a degradação que se verifica para as ôndulas biortogonais é mais significativa.

Tendo em atenção as dependências hierárquicas da DWT 3D (ver figura 5.6), os coeficientes localizados nos níveis mais altos, na primeira trama do GOF, são mais significativos do que os coeficientes nos níveis mais baixos, armazenados nas últimas tramas do GOF. Assim, para um dado nível de decisão, a *bitstream* da codificação não contém informação relativa aos coeficientes não significativos. Após a formação da estrutura 3D para a descodificação, a energia fica concentrada nos níveis mais altos, quer segundo a reconstrução no domínio do espaço, quer segundo a reconstrução no domínio do tempo. Na reconstrução das tramas através da DWT^{-1} no domínio do tempo, dado que os filtros da ôndula DB2 não são simétricos, os coeficientes relativos às duas primeiras tramas diminuem de valor em relação ao esperado, enquanto em relação aos restantes coeficientes se regista um aumento. Visando minimizar este efeito, procedeu-se a uma compensação dos coeficientes relativos às duas primeiras tramas, através de um método heurístico obtendo-se os resultados que se observam na figura 5.26. A degradação é tanto maior quanto maior for o nível de decisão considerado, sendo mais visível quando se observa uma imagem de cada vez. Porém, do ponto de vista subjectivo, quando se observam as imagens em sequência, esse efeito é praticamente imperceptível.

Verifica-se, também, para todas as sequências consideradas, para os mesmos níveis de decisão, que os valores máximos de PSNR que se obtêm com a ôndula DB2 são inferiores ou aproximadamente iguais aos que se obtêm com a DWT 9/7, exceptuando o caso da sequência “Mobile & Calendar”. Neste caso, para os níveis 16 e 32, os valores de PSNR com a DWT DB2 são maiores do que com a DWT 9/7.

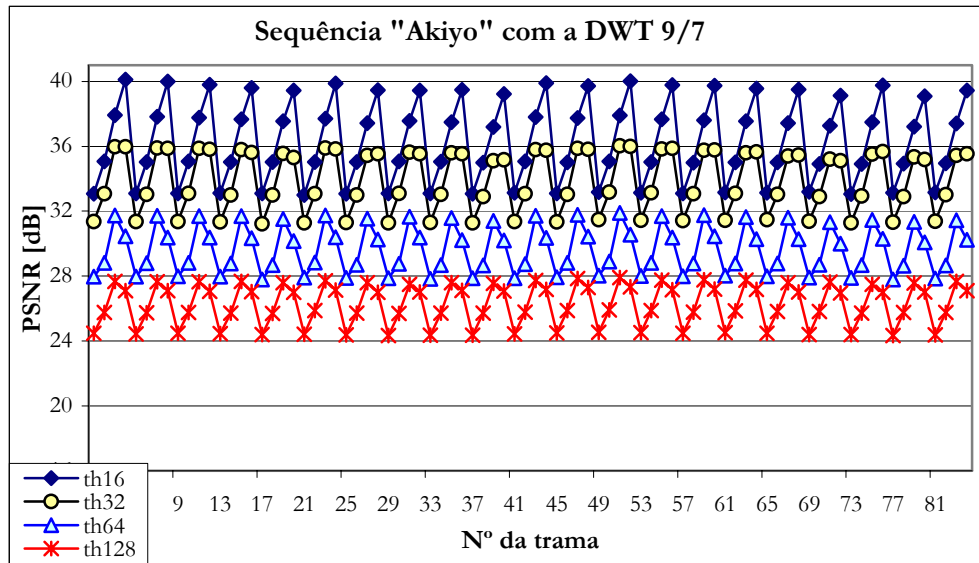


Figura 5.18: Valores de PSNR da luminância, para a sequência "Akiyo" com a DWT 9/7, para vários níveis de decisão.

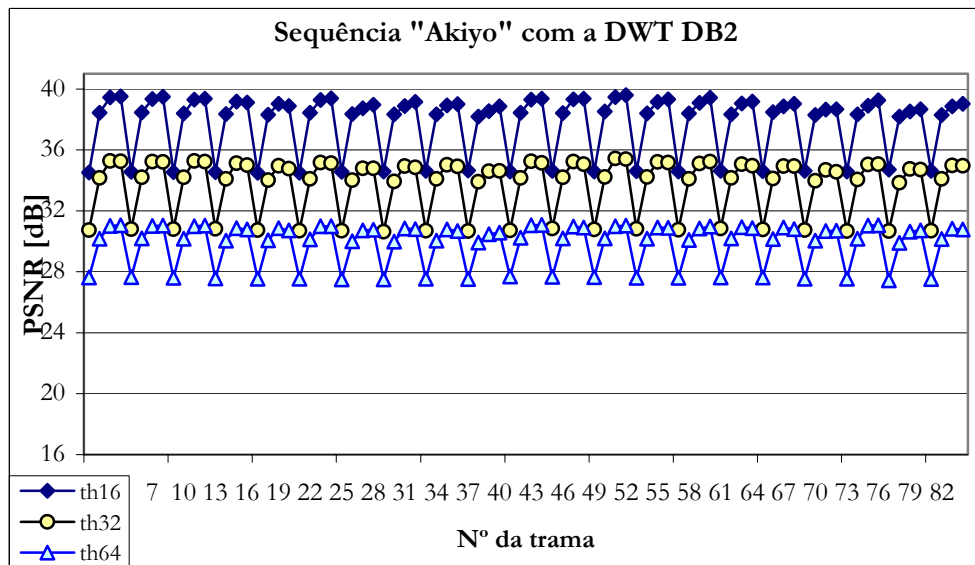


Figura 5.19: Valores de PSNR da luminância, para a sequência "Akiyo" com a DWT DB2, para vários níveis de decisão.

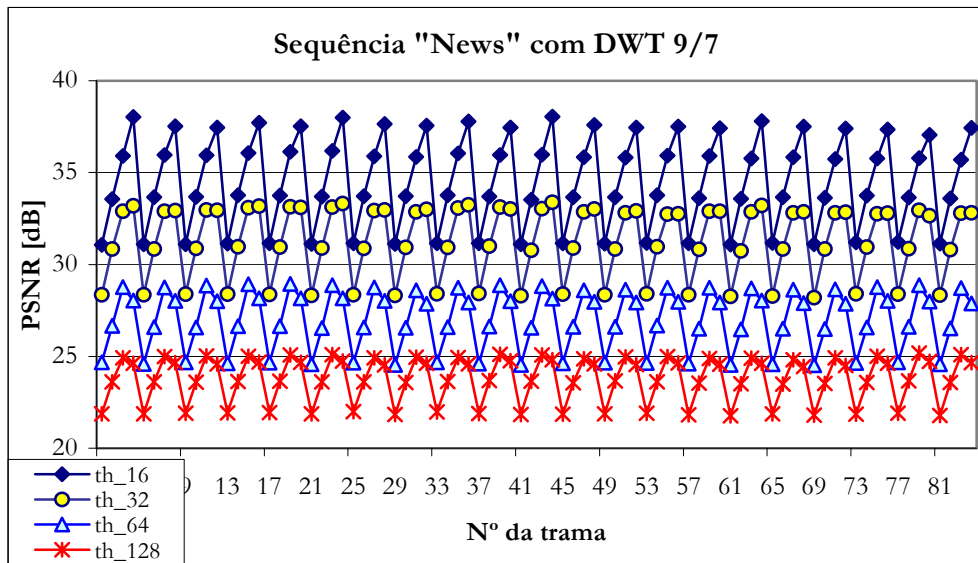


Figura 5.20: Valores de PSNR da luminância, para a sequência “News” com a DWT 9/7, para vários níveis de decisão.

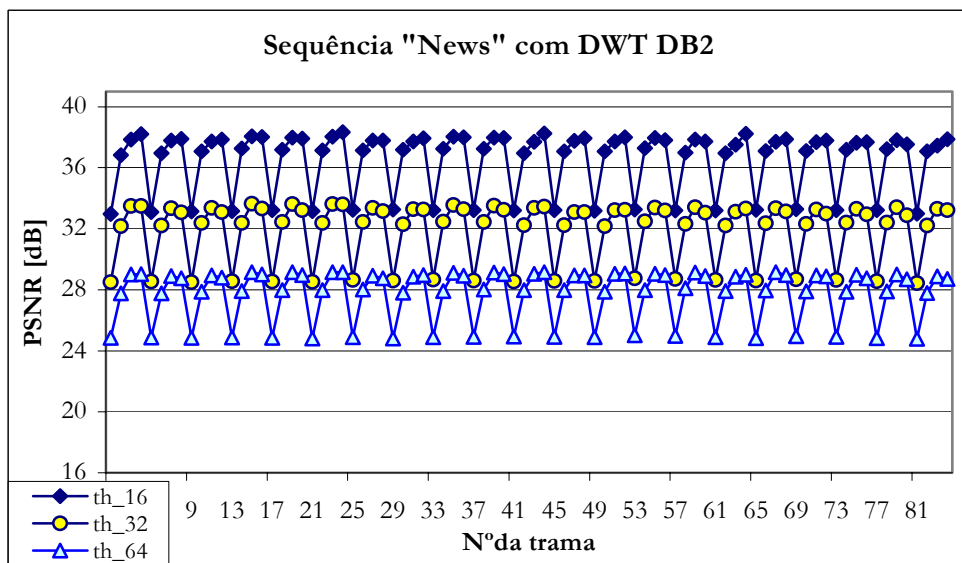


Figura 5.21: Valores de PSNR da luminância, para a sequência “News” com a DWT DB2, para vários níveis de decisão.

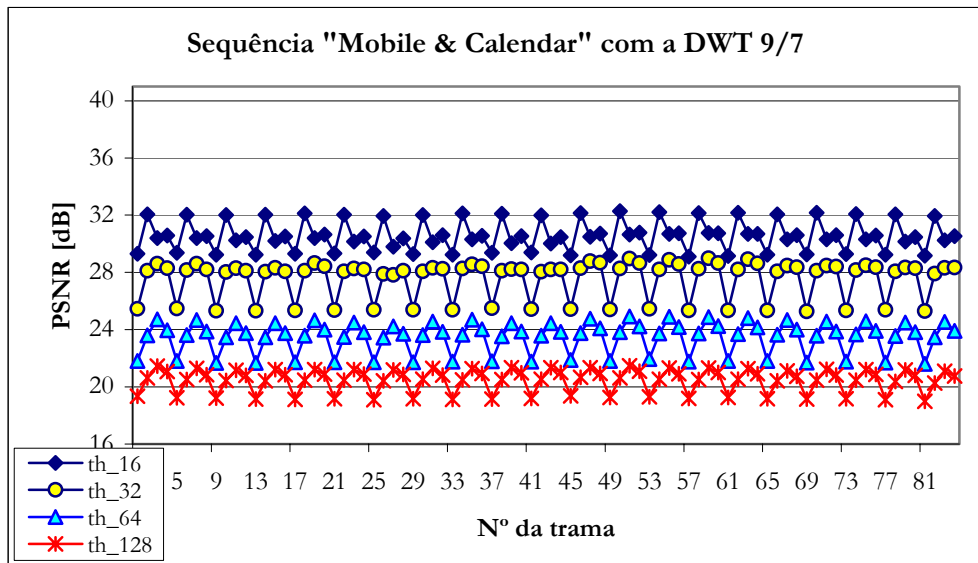


Figura 5.22: Valores de PSNR da luminância, para a sequência “Mobile & Calendar” com a DWT 9/7, para vários níveis de decisão.

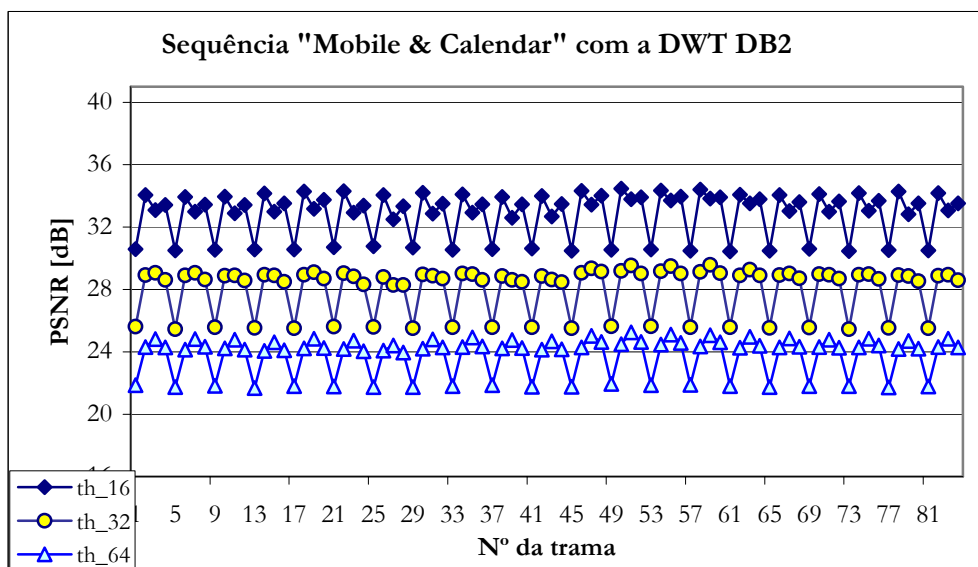


Figura 5.23: Valores de PSNR da luminância, para a sequência “Mobile & Calendar” com DWT DB2, para vários níveis de decisão.

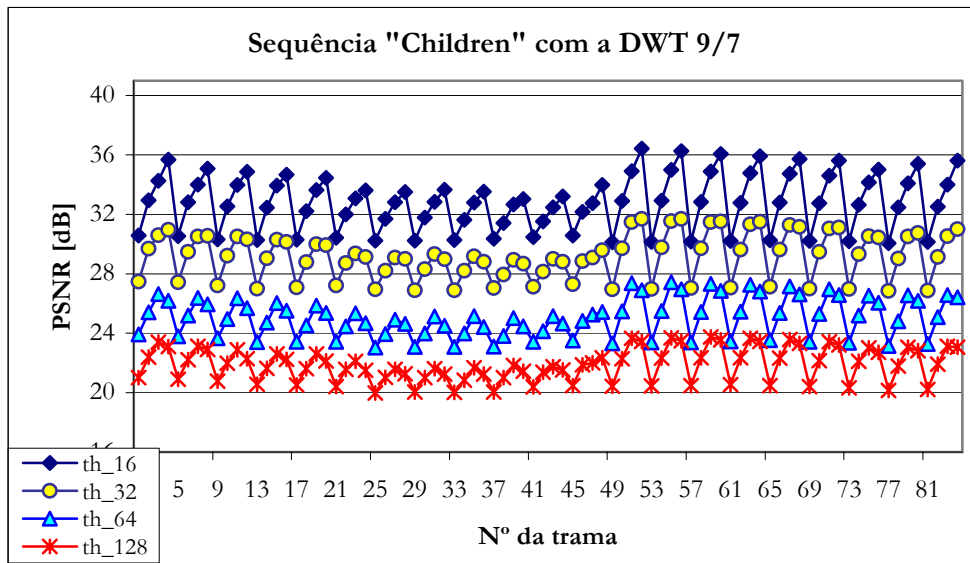


Figura 5.24: Valores de PSNR da luminância, para a seqüência “Children” com DWT 9/7, para vários níveis de decisão.

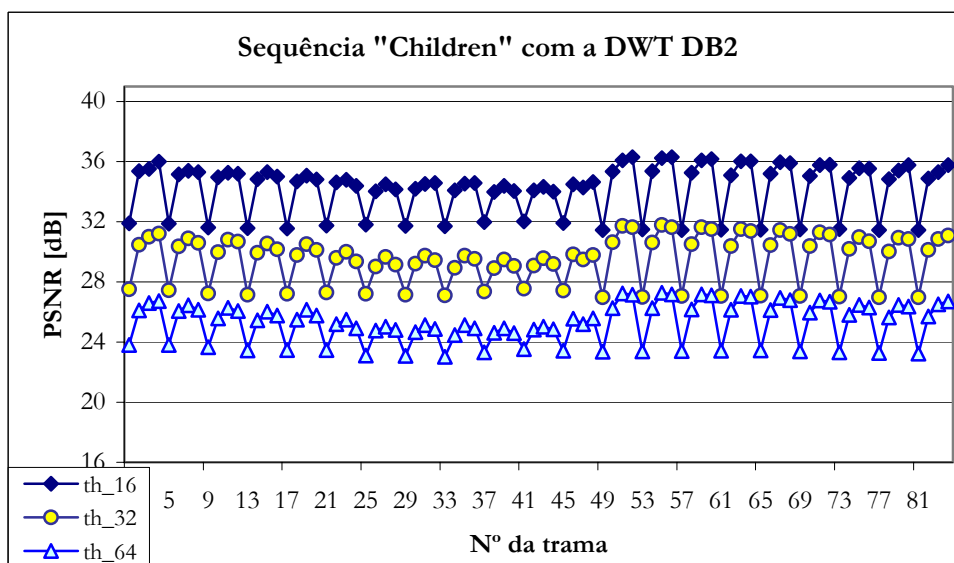


Figura 5.25: Valores de PSNR da luminância, para a seqüência “Children” com DWT DB2, para vários níveis de decisão.



a)



b)



c)

Figura 5.26: Imagens 153 a 156 (um GOF) para a sequência “Foreman”: a) imagens originais; b) imagens resultantes de codificação com DWT_9/7 e nível 16; c) imagens resultantes da codificação com DWT_9/7 e nível 64.

Na figura 5.26 apresentam-se as imagens originais e as imagens recuperadas após codificação, para um GOF completo (imagens 153 a 156) da sequência “Foreman”, para a DWT 9/7 e com os níveis de decisão 16 e 64. Escolheu-se este GOF em particular, porque as imagens que o constituem apresentam uma grande quantidade de movimento, como se observa pela figura 5.26, sendo, portanto, uma situação desfavorável. Como se verifica para o nível 64, a degradação da primeira trama é mais significativa em relação às restantes imagens, mantendo no entanto alguma inteligibilidade.

Na figura 5.27 apresentam-se as imagens originais e recuperadas, 12, 14, 16, e 18, para a sequência “Children”, com a DWT 9/7. Na sequência “Children” existem objectos naturais e sintéticos, registando-se movimento em dois planos: no plano dos objectos ao centro das imagens (movimentos das crianças e da bola), e o movimento da frase “**Welcome MPEG World**” (objecto sintético), deslocando-se do canto inferior direito para o centro da imagem. As imagens apresentadas não são contíguas, de modo a realçar o movimento, que se verifica

através das diferenças na posição dos objectos. Para o nível 32, em geral, a qualidade é boa. Porém, para o nível 128, apesar de se manter alguma inteligibilidade, regista-se uma degradação mais acentuada, em particular para os objectos mais pequenos ou para os que registam mais movimento, nomeadamente a bola (ver figura 5.27 c). Do ponto de vista subjectivo, os objectos estáticos, ou os que registam pouco movimento, ou ainda os objectos de maiores dimensões, apresentam uma menor degradação.

A degradação das imagens devido ao movimento dos objectos é evidente nos gráficos das figuras 5.24 e 5.25. entre as tramas 1 e 50, em que se regista movimento do objecto sintético (a frase “Welcome MPEG world”), dos gestos das crianças, e da bola (que ao ser lançada atinge a 2ª criança e ressalta, iniciando um movimento em sentido contrário).

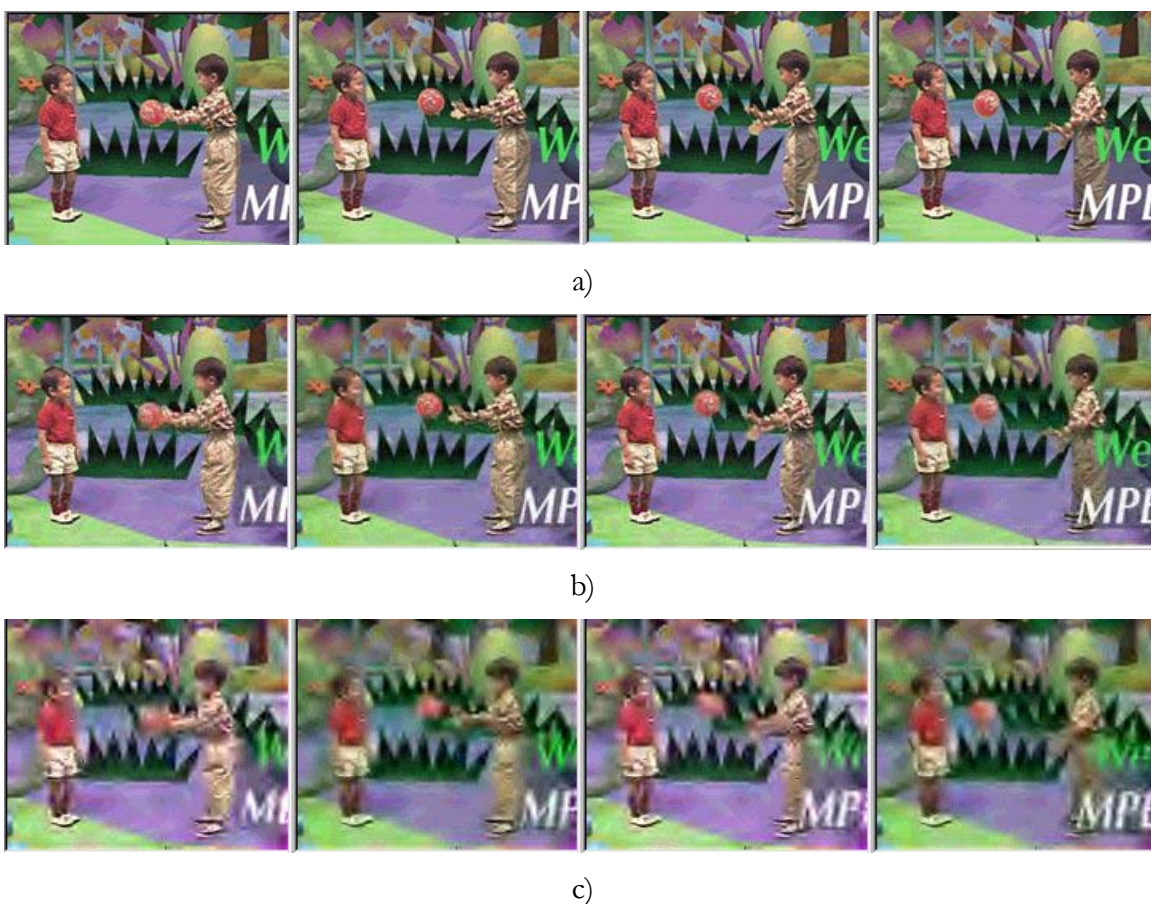


Figura 5.27: Imagens 12, 14, 16 e 18 para a sequência “Children”: a) imagens originais; b) imagens resultantes da codificação com DWT_9/7 e nível 32; c) imagens resultantes da codificação com DWT_9/7 e nível 128.



a)



d)



b)



e)



c)



f)

Figura 5.28: Sequência “Akiyo”: a) Imagem 26 original; b) imagem 26 recuperada da codificação com DWT 9/7 e nível 32; c) imagem 26 recuperada da codificação com DWT DB2 e nível 32. Sequência “Mobile & Calendar”: d) imagem 47 original; e) imagem 47 recuperada da codificação com DWT 9/7 e nível 64 e, f) imagem 47 recuperada da codificação com DWT 9/7 e nível 128.

Através das imagens que se apresentam na figura 5.28 também se pode avaliar a qualidade subjectiva das imagens obtidas por descodificação, considerando as sequências “Akiyo” e “Mobile & Calendar”.

Para a trama 26 da sequência “Akiyo”, avalia-se a degradação devida ao uso das duas ôndulas usadas para o cálculo da DWT no domínio do espaço, comparando a imagem original (5.28 a) com as imagens recuperadas a partir da codificação com a DWT 9/7 (5.28 b), e com a DWT DB2 (5.28 c), ambas codificadas com o nível 32. Verifica-se que a imagem obtida para a DWT DB2 apresenta maior degradação em relação à imagem obtida para a DWT 9/7, existindo uma degradação visual na região próxima do bordo inferior da imagem, não se verificando o mesmo efeito para a imagem obtida com a DWT 9/7. Estes resultados explicam-se com as melhores características das ôndulas biortogonais para a decorrelação em esquemas de codificação de imagens e vídeo, tal como já foi referido anteriormente.

Na sequência “Mobile & Calendar”, para a trama 47, avalia-se a degradação das imagens para níveis de decisão maiores, comparando a imagem original (5.28 d) com as imagens recuperadas a partir da codificação com nível 64 (5.28 e) e com nível 128 (5.28 f), ambas com a DWT 9/7 no domínio do espaço. Nas imagens vizinhas desta imagem, na sequência “Mobile & Calendar”, além do movimento da locomotiva e da bola, da direita para a esquerda, verifica-se, simultaneamente, um movimento do calendário, na direcção vertical, para cima e para baixo. Trata-se de uma sequência em que existem muitos detalhes e que regista muito movimento, de objectos independentes. Na imagem codificada com o nível 64 é possível, em geral, identificar os objectos contidos na imagem, em geral. A degradação é mais visível na região relativa ao calendário, mas ainda assim é possível identificar os números nele contidos. Como se esperava, na imagem codificada com o nível 128, a degradação é muito mais acentuada, sendo apenas possível identificar os objectos de maior dimensão, e em zonas onde se verifica um maior contraste de cor, nomeadamente a locomotiva, a bola e alguns objectos de fundo.

5.5.3 Tempo de Processamento para a EZW 3D e para a EZW⁻¹ 3D

Para avaliar o tempo de processamento associado ao CoDec de vídeo, falta avaliar o tempo de cálculo da codificação EZW 3D e da codificação de Huffman. O algoritmo da codificação de Huffman consiste essencialmente no endereçamento de tabelas pré-definidas, para obtenção dos códigos respectivos, num alinhamento sequencial. É, por isso, muito mais simples do que a codificação EZW 3D, que exige a construção e pesquisa de estruturas em árvore e em listas.

Por isso, os valores que a seguir se apresentam referem-se apenas ao tempo de processamento associado à codificação e à descodificação EZW 3D.

Para avaliar o tempo de processamento respeitante à codificação e à descodificação, mediram-se os tempos necessários à execução de ciclos completos de codificação e de descodificação, para grupos de 4 imagens, com todas as componentes de luminância e de crominância. Esses tempos dependem da informação contida nas tramas que constituem o GOF, podendo variar para GOFs de uma mesma sequência. Interessa, portanto, obter informação quanto aos valores médios do tempo de processamento por GOF.

Para cada uma das sequências de teste consideradas, mediram-se os tempos de codificação e de descodificação de 22 GOFs (88 tramas) e calcularam-se os valores médios por GOF apresentados nas tabelas 5.9 e 5.10. Analisando os resultados, verifica-se que os tempos de codificação e descodificação são muito elevados quando comparados com o tempo necessário para o cálculo das transformadas (apresentados no capítulo 4). Os valores que se obtêm com esta plataforma computacional não permitem a implementação de sistemas para codificação em tempo real. Devido à impotência desta questão, ela é referida com maior detalhe mais adiante nesta secção.

Verifica-se também que o tempo de descodificação, embora também elevado, é menor do que o de codificação, reflectindo a maior simplicidade do descodificador. Por outro lado, para os mesmos valores do nível de decisão, verifica-se que o tempo é maior nos casos em que se considera a óndula DB2 para a transformada no espaço. Tal deve-se à existência de coeficientes considerados significativos nas zonas correspondentes às bandas de maior frequência (no domínio da transformada), em especial nas regiões nos limites entre níveis (escalas). Este facto está directamente relacionado com a maior quantidade de informação gerada à saída do codificador, e que se reflecte no valor do factor de compressão como se referiu na secção 5.5.1.

Apesar de não serem apresentados nas tabelas 5.9 e 5.10, pode-se referir que para as várias sequências, para os vários níveis de decisão, e para as duas óndulas no espaço, o tempo associado à codificação (ou à descodificação) do sub-GOF relativo às componentes de luminância, representa entre 70% a 85% do tempo total associado ao GOF. Isto reflecte a maior quantidade de informação usada para representar as diferentes componentes da imagem.

Tempo médio para GOF=4 [s]									
DWT_9/7 no Espaço	Nível decisão	Akiyo	Container	Foreman	News	Table Tennis	Mobile	Children	
	EZW 3D	16	2,62	6,12	4,89	5,46	4,39	30,00	11,49
		32	1,44	2,73	2,11	2,42	1,70	7,24	4,03
		64	0,88	1,38	1,12	1,21	1,03	2,08	1,68
		128	0,60	0,72	0,69	0,69	0,69	0,81	0,87
IEZW 3D	16	2,43	5,64	4,84	5,02	3,91	29,23	10,73	
	32	1,19	2,41	1,80	2,04	1,64	6,74	3,74	
	64	0,67	1,12	0,84	0,92	0,84	1,83	1,32	
	128	0,43	0,60	0,48	0,48	0,51	0,63	0,62	

Tabela 5.9: Tempo médio, em segundos, de codificação e decodificação EZW 3D, para grupos de 4 tramas (GOF), e para a DWT no espaço com os filtros 9/7.

Tempo médio para GOF=4 [s]									
DWT_DB2 no Espaço	Nível decisão	Akiyo	Container	Foreman	News	Table Tennis	Mobile	Children	
	EZW 3D	16	3,46	5,97	7,42	7,38	4,74	38,58	15,79
		32	1,75	2,70	2,94	2,94	2,02	9,58	5,33
		64	1,00	1,37	1,40	1,37	1,15	2,46	1,99
	IEZW 3D	16	3,29	5,62	7,37	6,61	4,40	37,92	15,10
32		1,52	2,41	2,59	2,58	1,65	9,31	4,95	
64		0,81	1,11	1,11	1,11	0,85	2,20	1,69	

Tabela 5.10: Tempo médio, em segundos, de codificação e decodificação EZW 3D, para grupos de 4 tramas (GOF), e para a DWT no espaço com a ôndula DB2.

Relativamente aos valores elevados para os tempos médios de codificação/decodificação de um GOF, é necessário construir e manter duas listas de coeficientes, uma no passo dominante e outra no passo subordinado. Os elementos dessas listas são estruturas de dados contendo informação do valor dos coeficientes da estrutura da DWT 3D, e das suas coordenadas no GOF. A dimensão destas listas depende do número de elementos que as integram (coeficientes), registando variações ao longo dos processos de codificação ou de decodificação. A inserção ou a remoção de elementos nas listas é um processo dinâmico de gestão de espaços de memória, que podem não ser contíguos, condicionando o desempenho destes processos. A gestão de memória neste tipo de estruturas em linguagens de alto nível não é eficiente, como é o caso da aplicação desenvolvida neste trabalho (em ambiente Windows®), e que explicam os resultados que se obtêm.

Por outro lado, o uso do DSP TMS320C6201 também não é uma solução a considerar. Devido às características da sua arquitectura interna, este DSP é adequado para a realização de algoritmos de processamento de sinal, com estruturas regulares, e que utilizem operações aritméticas como por exemplo multiplicações. Neste caso, a solução passa pela adopção de circuitos (*hardware*) dedicados, optimizados para a realização deste tipo de processamento em particular, e que se deixa como trabalho futuro.

De modo a comparar os resultados das tabelas 5.9 e 5.10 apresenta-se um conjunto de gráficos do tempo de processamento para as seguintes situações: EZW 3D vs. EZW⁻¹ 3D com a DWT_{9/7} no espaço (figuras 5.29 a 5.31) e EZW 3D: DWT_{9/7} no espaço vs. DB2 no espaço (figuras 5.32 a 5.34).

Para cada uma destas situações apresentam-se 3 gráficos: o caso mais favorável, o caso mais desfavorável e um caso intermédio. Na comparação entre a codificação e a decodificação com a DWT_{9/7} no espaço, para os casos apresentados (figuras 5.29 a 5.31), verifica-se que o tempo de codificação é maior que o da decodificação, como é de esperar visto que o algoritmo da decodificação é mais simples. Verifica-se, também, que a relação entre os tempos nas duas situações apresenta um andamento quase linear face às variações do nível de decisão (escala logaritmica no nível de decisão). A relação entre os tempos (EZW⁻¹ vs. EZW) varia entre 69% (“Akiyo” com nível de decisão 128) e 98% (“Mobile & Calendar” com nível de decisão 16). Para a situação em que se compara a EZW⁻¹ e a EZW para a DWT DB2 no espaço, verifica-se uma relação idêntica.

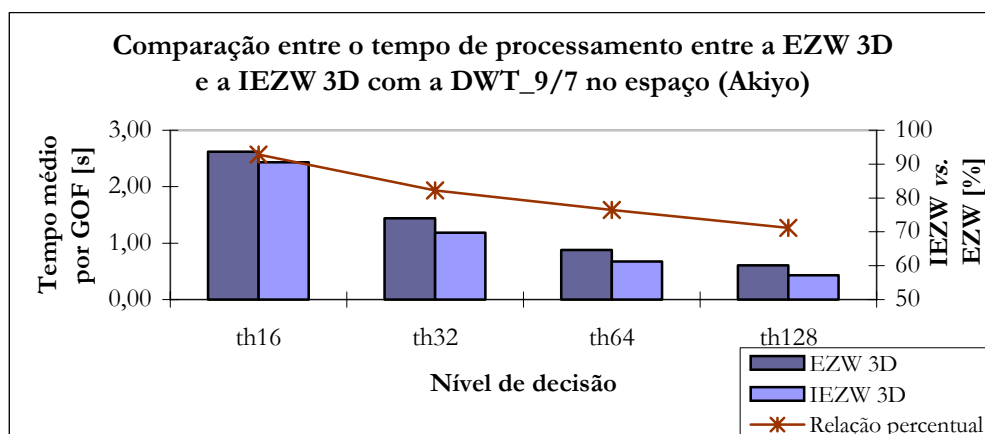


Figura 5.29: Tempo de processamento: EZW⁻¹ 3D vs. EZW 3D (9/7), no caso mais favorável.

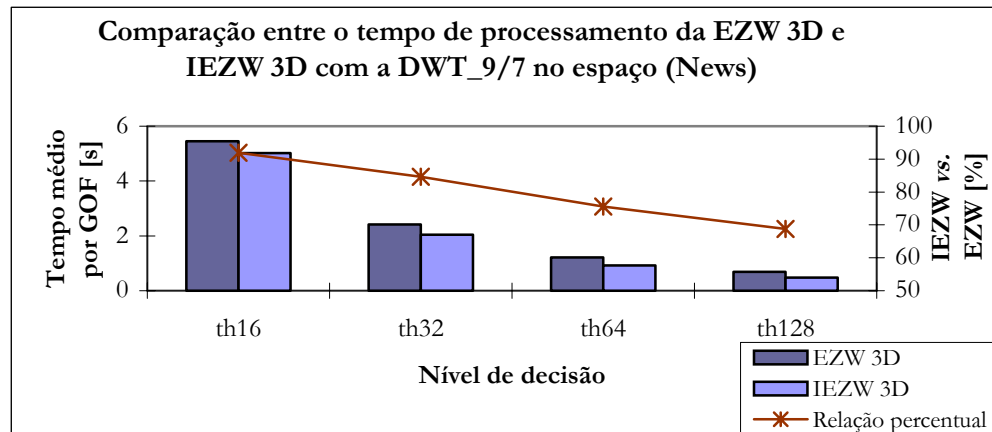


Figura 5.30: Tempo de processamento: EZW⁻¹ 3D vs. EZW 3D (9/7), no caso intermédio.

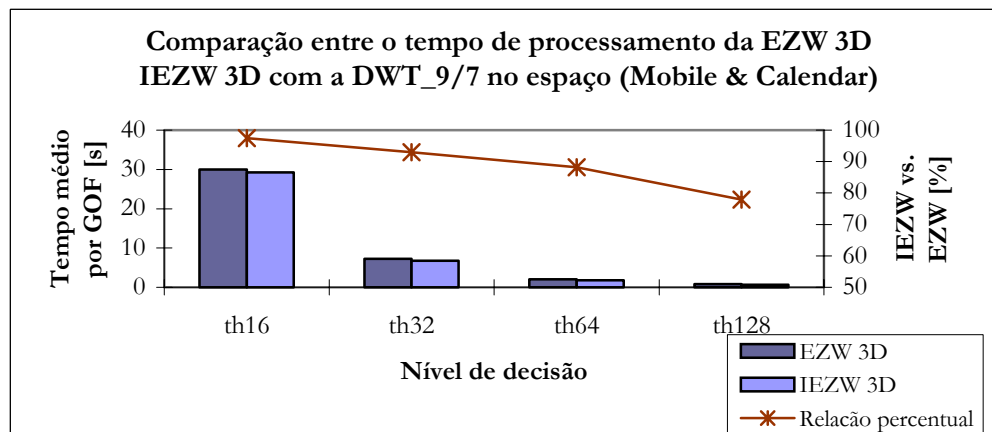


Figura 5.31: Tempo de processamento: EZW⁻¹ 3D vs. EZW 3D (9/7), no caso mais desfavorável.

Para a comparação entre a codificação EZW 3D com a DWT 9/7 e a DWT DB2 apresentam-se os gráficos das figuras 5.32 a 5.34.

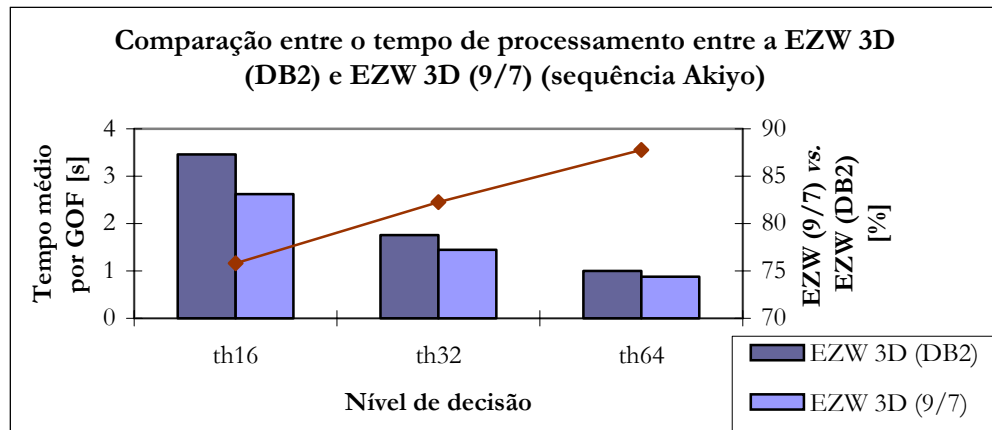


Figura 5.32: Tempo de processamento da EZW 3D: (9/7) vs. (DB2), no caso mais favorável.

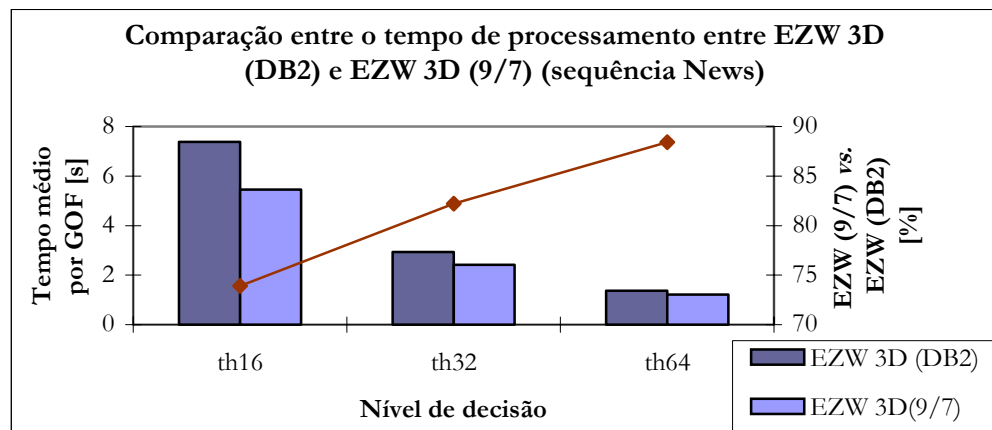


Figura 5.33: Tempo de processamento da EZW 3D: (9/7) vs. (DB2), no caso intermédio.

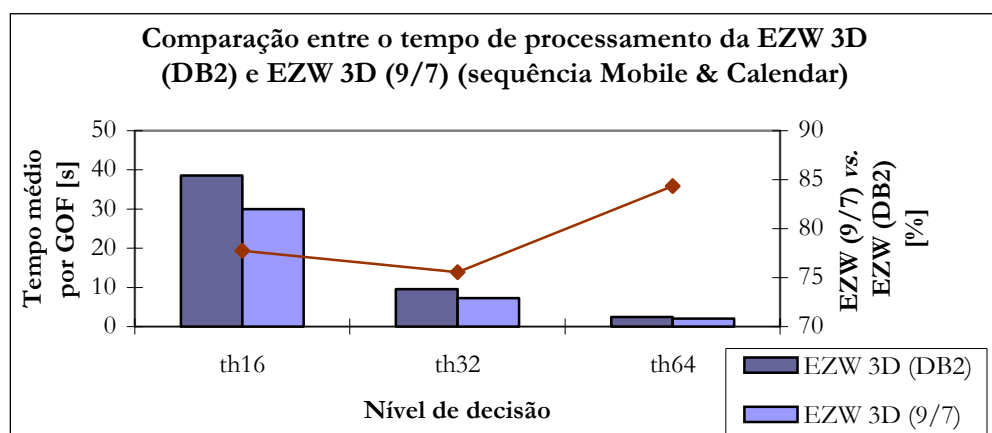


Figura 5.34: Tempo de processamento da EZW 3D: (9/7) vs. (DB2), no caso mais desfavorável.

Da análise dos gráficos das figuras 5.32 a 5.34, para os casos apresentados, verifica-se que o peso do processamento associado à codificação dos coeficientes da estrutura é maior para a ôndula DB2 para a DWT no espaço, do que para a DWT 9/7. Verifica-se também que a relação entre os tempos relativos às duas situações, apresenta um andamento quase linear na situação mais favorável e na situação intermédia, o que não se verifica na situação mais desfavorável. A relação entre os tempos de processamento varia entre 73% (“News” com nível de decisão 16) e 88% (“News” com nível de decisão 64). Esta variação, maior na situação de uso da ôndula DB2 no espaço, deve-se ao maior número de coeficientes a codificar, pelas razões já anteriormente referidas.

5.6 Conclusões

Neste capítulo apresentou-se a estrutura do CoDec de vídeo desenvolvido, e fez-se a descrição do funcionamento dos seus principais blocos constituintes, nomeadamente a formação da “árvore” 3D de coeficientes da transformada. O CoDec é baseado na organização dos coeficientes numa estrutura em árvore 3D, obtidos com base na transformada de ôndulas no domínio do tempo, com dois níveis de decomposição, e no domínio do espaço, com 3 níveis de decomposição. Consideram-se as ôndulas ortogonais Daubechies 2 e biortogonais (4,4). Na formação da estrutura 3D consideram-se grupos de 2^n ($n \in N_0$) imagens. No caso particular do CoDec realizado, consideram-se grupos de 4 imagens.

Foi referido o modo como se faz a extensão da codificação EZW (para imagem) a sequências de vídeo, e descreveram-se os algoritmos da codificação EZW 3D e da descodificação EZW⁻¹ 3D. Para avaliar o desempenho do CoDec realizado, consideraram-se sequências de vídeo de teste em formato QCIF, definidas para as normas H.263 e MPEG.

Obteve-se um conjunto de resultados, para os vários parâmetros a considerar na avaliação do desempenho, tais como: o factor de compressão, a relação sinal ruído entre as imagens recuperadas e as imagens originais, a qualidade subjectiva das imagens e os tempos de processamento.

Os factores de compressão obtidos com a codificação EZW 3D e a codificação de Huffman variam, aproximadamente, entre 10 e 70. Os factores de compressão, quando se usa a DWT no espaço, com os filtros 9/7, são quase sempre maiores do que quando se usa a ôndula DB2. os

poucos casos em que se verifica o inverso, estão relacionados com o facto das tabelas de Huffman usadas não se encontrarem totalmente adaptadas à codificação em causa. É possível obter ainda maiores factores de compressão, através da utilização de técnicas de estimação de movimento, que se indica como possível trabalho futuro.

A avaliação da qualidade das imagens recuperadas fez-se através do cálculo da relação sinal ruído de pico (PSNR), e da qualidade subjectiva por comparação com as imagens originais. Para as seqüências consideradas, a degradação das imagens recuperadas é, em geral, menor quando se usam os filtros 9/7 para o cálculo da DWT no domínio do espaço.

Os valores obtidos para o PSNR nem sempre indicam este facto, dado que não se consideram as componentes de crominância e não se contabiliza a degradação nos limites da imagem.

A qualidade subjectiva das imagens indica, também, que a degradação visual é mais acentuada com o uso da DWT DB2 no domínio do espaço, em particular para os níveis de decisão mais altos. Verifica-se também uma degradação mais acentuada na primeira trama do GOF, e que se deve aos erros introduzidos na codificação, em particular devido às características da ôndula DB2 na DWT⁻¹ no domínio do tempo.

Os tempos de processamento obtidos, levam a concluir que só é possível obter codificadores do tipo proposto em tempo real através do desenvolvimento de processadores especializados para a construção e pesquisa em árvore de coeficientes 3D.

Capítulo 6

CONCLUSÕES

Os principais objectivos do trabalho desta tese são o desenvolvimento de um CoDec de vídeo de ritmo binário variável, baseado na transformada de ôndulas, a avaliação dos requisitos computacionais e dos tempos de processamento associados a este tipo de codificação de vídeo, e a avaliação do desempenho de diferentes tipos de ôndulas para a codificação de vídeo.

As ôndulas ortogonais de Daubechies e as ôndulas biortogonais foram seleccionadas neste trabalho, porque conduzem à realização de filtros do tipo FIR com fase linear no caso das ôndulas biortogonais, e de fase quase linear nos casos das ôndulas ortogonais de Daubechies. Os filtros FIR de fase linear são sempre desejáveis em processamento de imagem e de vídeo, porque minimizam os efeitos de distorção de fase. Além disso, a transformada de ôndulas associada a este tipo filtros pode ser factorizada através do esquema progressivo.

O esquema progressivo, que foi aplicado no cálculo da transformada de ôndulas, permite a redução do custo computacional devido principalmente a duas razões: porque permite realizar operações de uma forma recursiva, e porque a factorização das matrizes polifásicas origina a decomposição triangular (superior e inferior) das matrizes intermédias, sendo os restantes elementos nulos. Por outro lado, o esquema progressivo não requer memória adicional, efectuando o que se designa, geralmente, por “in-place computation”.

O CoDec desenvolvido baseia-se na formação de estrutura 3D, obtida por decomposição através da transformada de ôndulas no domínio do tempo, com dois níveis de decomposição, e no domínio do espaço, com 3 níveis de decomposição, considerando-se as ôndulas ortogonais Daubechies 2 e biortogonais (4,4). Na formação da estrutura 3D consideram-se grupos de 2^n ($n \in N_0$) imagens. No caso particular do CoDec realizado consideram-se grupos de 4 imagens,

subdivididos em 3 sub-grupos assim constituídos: um subgrupo de matrizes relativo às componentes de luminância, e os restantes dois subgrupos relativos às componentes de crominância.

Os resultados obtidos na avaliação do desempenho do CoDec, comparando o uso das ôndulas Daubechies 2 e biortogonais (4,4) para o cálculo da DWT no domínio do espaço, indicam que com as ôndulas biortogonais (4,4) se obtêm, em geral, maiores factores de compressão, maior relação sinal/ruído, e que a qualidade subjectiva é também melhor. Porém, nalguns casos, os resultados de PSNR não indicam este facto, devendo-se isto ao facto de no seu cálculo não se considerarem as componentes de crominância e os limites das imagens para o cálculo de PSNR.

Os factores de compressão obtidos para a codificação EZW 3D sofrem um aumento entre 3 e 5, aproximadamente, quando se usa a codificação de Huffman. Os factores de compressão, quando se usa a ondula biortogonal são, quase sempre, maiores do que com a ôndula DB2, variando entre 9% e 16% sem codificação de Huffman, e 12,5% e 63% com codificação de Huffman. Todavia, existem casos em que tal não se verifica, o que se julga ser devido às tabelas de Huffman usadas não estarem completamente adaptadas às características deste tipo de codificação. A obtenção de maiores factores de compressão é possível, através de técnicas de estimação de movimento, que se deixa para trabalho futuro.

Com base nos factores de compressão para os vários níveis de decisão considerados para a quantificação dos coeficientes, e para as duas ôndulas no domínio do espaço, e tendo em atenção o factor de compressão e a degradação das imagens, foi possível estabelecer o limite superior e inferior do nível de decisão, para se obter imagens codificadas com uma qualidade aceitável. Assim, concluiu-se que o nível de decisão mínimo é 16, tanto para a DWT 9/7, como para a DWT DB2, e que o nível de decisão máximo é 64 para a ôndula DB2 e 128 para a ôndula DB9/7. Com base nos factores de compressão e na qualidade das imagens recuperadas, para níveis de decisão entre 16 e 64, conclui-se que o CoDec pode produzir ritmos binários variáveis para todas as sequências de vídeo consideradas.

Por último, os tempos de processamento que se obtêm para a codificação e para a descodificação EZW 3D, tendo como referência um processador de uso geral (Intel Pentium III a 550 MHz), indicam a impossibilidade de obter processamento em tempo real. A utilização de processadores digitais de sinais programáveis de alto desempenho, tal como o processador com arquitectura VLIW usado neste trabalho, permitem acelerar o cálculo das transformadas.

Adoptando-se o esquema progressivo foi usado no cálculo das transformadas de ôndulas, implementaram-se as rotinas de cálculo da DWT e da DWT^{-1} , no tempo e no espaço no DSP TMS320C6201 a funcionar a 167 MHz, obtendo-se uma aceleração de processamento que varia entre 1,12 e 15, em relação à implementação num processador de uso geral (Intel Pentium III a 550 MHz). A aceleração de processamento é maior para o cálculo das transformadas no tempo do que no espaço. Para as transformadas no espaço, melhores resultados foram alcançados para a ôndula DB2 em relação a DB9/7. Isto deve-se ao maior paralelismo que se obtém para o algoritmo da DWT e DWT^{-1} para a ôndula DB2, devido às características do algoritmo e às dependências dos dados.

No entanto, não é possível realizar codificação/descodificação em tempo real, mesmo utilizando o processador de sinal para codificar sequências de baixa resolução (formato QCIF). Grande parte do tempo de processamento corresponde à construção e pesquisa de valores na árvore 3D, operação para a qual o processador de sinal também não é adequado. Pode-se alcançar processamento em tempo real, com a tecnologia actual, desenvolvendo um processador especializado para a construção e pesquisa de dados em estruturas hierárquicas.

6.1 Perspectivas de Trabalho Futuro

Face aos resultados obtidos, apontam-se as seguintes direcções para desenvolvimento do trabalho apresentado:

A implementação de esquemas de estimação de movimento, de forma a explorar, ainda mais, a redundância temporal.

Desenvolver um processador dedicado para a construção e pesquisa em estruturas de dados hierárquicas 3D, que se revelou ser o bloco de processamento do CoDec de vídeo computacionalmente mais exigente.

Codificação de símbolos mais eficiente, por adaptação das tabelas de codificação de Huffman, e considerando, também, a codificação aritmética.

Comparação do desempenho do codificador desenvolvido com os codificadores normalizados do tipo H.263 e MPEG.

Referências Bibliográficas

- Algaz+97 V. R. Algazi, R. R. Estes Jr, “Analysis Based Coding of Image Transform and Subband Coefficients”, *Proc. of the SPIE*, vol. 2564, pp. 11–21, 1997.
- Alon+99 M. Alonso, E. J. Finn, “Física”, *Addison-Wesley Espanha S.A.*, 1999.
- Anton+92 M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, “Image Coding using Wavelet Transform”, *IEEE Trans. on Image Processing*, vol. 1, N° 2, pp. 205–220, Abril de 1992.
- Bhask+96 V. Bhaskaran, K. Konstantinides, “Image and Video Compression Standards”, *Kluwer Academic Press*, 2ª Edição, 1996.
- Bliu+93 B. Liu, A. Zaccarin, “New Fast Algorithms for the Estimation of Block Motion Vectors”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, N° 2, pp. 148–157, Abril de 1993.
- Brisl_96 C. M. Brislawn, “Classification of Nonexpansive Symmetric Extension Transforms for Multirate Filter Banks”, *Applied and Computational Harmonic Analysis*, vol. 3, pp. 337–357, Março de 1996.
- Caer_97 J. J. Caeiro, “Técnica de Processamento de Sinais com Transformadas Baseadas em Ondulas”, *Electricidade*, N° 348, pp. 299–305, Outubro de 1997.
- Cald+97 A. R. Calderbank, I. Daubechies, W. Sweldens, B. L. Yeo, “Lossless Image Compression Using Integer To Integer Wavelet Transform”, <http://cm.bell-labs/who/wim/papers/papers.html>
- Cald+98 A. R. Calderbank, I. Daubechies, W. Sweldens, B. L. Yeo, “Wavelet Transforms that Map Integers to Integers”, *Applied and Computacional Harmonic Analysis Journal*, vol.5, N°3, pp.332–369, 1998. <http://cm.bell-labs/who/wim/papers/papers.html>
- Chao+96 H. Chao, P. Fisher, “An Approach to Fast Integer Reversible Wavelet Transforms for Image Compression”, *Tech. Report, Computer & Information Science, Inc.*, 1996, <http://www.kaist.ac.kr/~cggeju/RESOURCE/image.html>
- Chen+96 Y. Chen, W. A. Pearlman, “Three-Dimensional Subband Coding of Video Using the Zero-Tree Method”, *Proc. SPIE’s 1996 Symposium on Visual Communications and Image Processing*, vol. 2727, Março de 1996.
- Chrys+00 C. Chrysafis, A. Ortega, “Minimum Memory Implementations of the Lifting Scheme”, *Proc. SPIE’s Internacional Symposium on Optical Science and Technology*, San-Diego, Agosto2000, <http://citeseer.nj.nec.com/332385.html>.
- Daub_92 I. Daubechies, “Ten Lectures on Wavelets”, *SIAM – Society for Industrial and Applied Mathematics*, 1992.
- Daub+98 I. Daubechies, W. Sweldens, “Factoring Wavelet Transforms into Lifting Steps”, *Joseph Fourier Analysis and Applications*, vol. 4, N° 3 pp. 245–267, 1998.

- DSP_Hbk V. K. Madiseti, D. B. Williams, “The Digital Signal Processing Handbook”, *CRC Press/IEEE Press*, 1998.
- Dudg+84 D. E. Dudgeon, R. M. Mersereau, “Multidimensional Digital Signal Processing”, Prentice-Hall, 1984.
- Gall_91 D. Le Gall, “MPEG: A Video Compression Standard for Multimedia Applications”, *Communications of the ACM*, vol. 34, N° 4, pp. 47–58, Abril de 1991.
- Gosw+99 J. Goswami, A. K. Chan, “Fundamentals of Wavelets – Theory, Algorithms and Applications”, *John Wiley & Sons*, 1999.
- Graps_95 A. Graps, “An Introduction to Wavelets”, *IEEE Computacional Science and Engineering*, vol. 2, N°2, pp. 50–61, 1995.
- Hilt+94 M. L. Hilton, B. D. Jawerth, A. Sengupta, “Compressing Still and Moving Images With Wavelets”, *Multimedia Systems*, vol. 2, N° 3, Abril 1994.
- Holl+00 J. Hollingworth, D. Butterfield, B. Swart, J. Allsop, *et. al.* “C++ Builder™ 5.0 Developer’s Guide”, *SAMS*, 2000.
- Ifch+93 E. C. Ifeachor, B. W. Jervis, “Digital Signal Processing: A Practical Approach”, pp. 614–678, *Addison-Wesley*, 1993.
- ITU.263 ITU-T (International Telecommunication Union – Telecommunication Standardisation Sector), “Video Coding for Low Bitrate Communication”, Draft ITU-T Recommendation H.263, Maio de 1996.
- JiLu+97 J. Lu, M. L. Liou, “A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, N° 2, pp. 429–433, Abril de 1997.
- Jlim_90 J. S. Lim, “Two Dimensional Signal and Image Processing”, *Prentice-Hall*, 1990
- Kung_88 S. Y. Kung, “VLSI Array Processors”, *Prentice-Hall*, 1998
- LaiPo+96 L. M. Po, W. C. Ma, “A Novell Four-Step Search Algorithm for Fast Block Motion Estimation”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, N° 3, pp. 313–317, Junho de 1996.
- Mand+96 M. K. Mandal, E. Chan, X. Wang, S. Panchanatham “Multiresolution Motion Estimation Techniques for Video Compression”, *Optical Enginnering*, vol. 35, N° 1, pp. 128–136, Janeiro de 1996.
- Malat_89 S. G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”, *IEEE Trans. on Pattern Analisys and Machine Inteligence*, vol. 7, pp. 674–693, 1989.
- Marc+00 M. Marcellin, M. Gormish, A. Bilgin, M. Poliek, “An Overview of JPEG-2000”, *Proc. of IEEE Data Compression Conference*, pp. 523–541, 2000.
- Mart+97 S. A. Martucci, I. Sodagar, T. Chiang, Y. Q. Zhang, “A Zerotree Wavelet Video Coder”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, N° 1, pp. 109–118, Fevereiro de 1997.
- MPEG_01 MPEG, “MPEG-4 Video Verification Model version 18.0”, documento ISO/MPEG N3908, Pisa, Janeiro de 2001, pp.15, 16. <http://www.mpeg.org>

- MPEG_02 MPEG, “Report of Ad-Hoc Group on Exploration of Interframe Wavelet Technology in Video”, documento ISO/MPEG02 M8205, Jeju, Março de 2002. <http://www.mpeg.org>.
- Ngan+96 K. N. Ngan, W. L. Choi, A. Millin, “Very Low Bit Rate Coding using H.263 Coder”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, N° 3, pp. 308–312, Junho de 1996.
- Niels_98 O. M. Nielsen, “Wavelets in Scientific Computing”, Department of Math. Modelling, Technical University of Denmark, PhD Thesis, Lingby, Dinamarca, Março de 1998, Parte 1, Cap. 1 e 2, <http://www.imn.dtu.dk/~omni/>
- Ohm_94 J. R. Ohm, “Three-Dimensional Subband Coding with Motion-Compensation”, *IEEE Trans. on Image Processing*, vol. IP-3, N° 5, pp. 559–571, Setembro de 1994. http://www.ient.rwth-aachen.de/engl/index_ient.html
- Pearl+98 W. A. Pearlman, A. Said, “A Survey of the State-of-the-Art and Utilization of Embedded Tree-Based Coding”, *Proc. of IEEE Int. Symp. On Circuits and Systems, Invited Paper*, Monterey, Canada, Junho de 1998.
- Pirsh_99 P. Pirsch, “Architectures for Digital Signal Processing”, *John Wiley & Sons*, 1999.
- ReLi+94 R. Li, B. Zeng, M. L. Liou, “A New Three-Step Search Algorithm for the Block Motion Estimation”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, N° 4, pp. 438–442, Agosto de 1994.
- Said1+96 A. Said, W. A. Pearlman, “A New, Fast, and Efficient Image CoDec Based on Set Partitioning in Hierarchical Trees”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, N° 3, pp. 243–249, Junho de 1996.
- Said2+96 A. Said, W. A. Pearlman, “An Image Multiresolution Representation for Lossless and Lossy Compression”, *IEEE Trans. on Image Processing*, vol. 5, pp. 1303–1310, Setembro de 1996.
- Seck+01 A. Secker, D. Taubman, “Motion-Compensated Highly Scalable Video Compression Using Adaptive 3D Wavelet Transform Based on Lifting”, *Proc. of ICIP 2001, Int. Conf. on Image Processing*, pp.1029–1032, Outubro de 2001.
- Shapr_93 J. M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients”, *IEEE Trans. on Signal Processing*, vol. 41, N° 12, pp. 3445–3462, Dezembro de 1993.
- Sousa_98 L. Sousa, “Processadores Digitais Especializados”, *DEEC – Instituto Superior Técnico*, 1998.
- Sweld_95 W. Sweldens, “The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Construction”, *Wavelet Applications in Signal and Image Processing III*, pp. 68–79, 1995. <http://cm.bell-labs/who/wim/papers/papers.html>
- Sweld_96 W. Sweldens, “The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets”, *Applied Comput. & Harmonic Analysis*, Vol 3, pp. 186–200, 1996. <http://cm.bell-labs/who/wim/papers/papers.html>
- Sweld_97 W. Sweldens, “The Lifting Scheme: A Construction of Second Generation Wavelets”, *SIAM Journal Math. Analysis*, N° 2, vol. 29, pp. 511–546, 1997. <http://cm.bell-labs/who/wim/papers/papers.html>

- Skod+01 A. Skodras, C. Christopoulos, T. Ebrahimi “The JPEG2000 Still Image Compression Standard”, *IEEE Signal Processing Magazine*, N°5, vol.18, pp.36 – 58, Setembro de 2001.
- Tex_186 “TMS320C6x Assembly Language Tools User’s Guide”, *Texas Instruments*, SPRU186E, Março de 1999.
- Tex_187 “TMS320C6x Optimizing C Compiler User’s Guide”, *Texas Instruments*, SPRU187E, Março de 1999.
- Tex_189 “TMS320C62x/C67x CPU and Instruction Set Reference Guide”, *Texas Instruments*, SPRU189D, Março de 1999.
- Tex_197 “TMS320C62x/C67x Technical Brief”, *Texas Instruments*, SPRU197D, Abril de 1999.
- Tex_198 “TMS320C62x/C67x Programmers Guide”, *Texas Instruments*, SPRU198C, Fevereiro de 1999.
- Tex_328 “Code Composer Studio User’s Guide”, *Texas Instruments*, SPRU328, Maio de 1999.
- Tham_95 J. Y. Tham, “Detail Preserving Image Compression using Wavelet Transform”, *IEEE Student Paper Book*, pp.1–9, 1995.
- Tham+96 J. Y. Tham, S. Ranganath, A. A. Kassim, “Scalable Very Low Bit-Rate Video Compression Using Motion Compensated 3-D Wavelet Decomposition”, *IEEE ISPACS Workshop*, vol. 3, pp.38.7.1 – 38.7.5, Novembro de 1996.
- Tham1+98 J. Y. Tham, S. Ranganath, A. A. Kassim, “Highly Scalable Wavelet-Based Video Codec for Very Low Bit-Rate Environment”, *IEEE Journal on Selected Areas in Communications – Special Issue in Very Low Bit-Rate Coding*, vol. 16, N° 1, Junho de 1998.
- Tham2+98 J. Y. Tham., S. Ranganath, M. Ranganath, A. A. Kassim, “A Novell Unrestricted Center-Based Diamond Search Algorithm for Block Motion Estimation”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, N° 4, pp. 369–377, Agosto de 1998.
- Usevt_01 Bryan E. Usevitch, “A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000”, *IEEE Signal Processing Magazine*, vol. 18, N° 5, pp. 22–35, Setembro de 2001.
- Vette+92 M. Vetterli, C. Herley, “Wavelets and Filter Banks: Theory and Design”, *IEEE Transactions on Signal Processing*, vol. 40, pp. 2207–2232, 1992.
- Villa+95 J. D. Villasenor, B. Belzer, J. Liao, “Wavelets Filter Evaluation for Image Compression”, *IEEE Transactions on Image Processing*, vol. 4, N° 8, Agosto de 1995.
- Zhan+92 Y. Q. Zhang, S. Zafar, “Motion-Compensated Wavelet Transform Coding for Color Video Compensation”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, N° 3, pp. 285–296, Setembro de 1992.

Apêndice A

A – Coeficientes dos Filtros de Ôndulas Ortogonais e Biortogonais

Os coeficientes dos filtros da família de ôndulas ortogonais de Daubechies, e da família de ôndulas biortogonais do tipo *spline*, são referidos em [Daub_92], na tabela 6.1 (pp.195) e na tabela 8.3 (pp. 279), respectivamente. Porém, com o objectivo de facilitar a interpretação deste texto, em particular no que se refere à obtenção dos coeficientes da DWT e DWT⁻¹ pelo esquema progressivo, apresentam-se os coeficientes dos filtros das ôndulas ortogonais de Daubechies, até 5 momentos nulos, e das ôndulas biortogonais do tipo *spline* (4,4) e (5,5). Apresentam-se também as propriedades mais importantes das ôndulas ortogonais de suporte compacto de Daubechies.

A1) Propriedades da ôndulas de Suporte Compacto de Daubechies

Ortonormalidade

Usando a equação 3.29 é possível representar a condição de ortonormalidade (3.29) dos deslocamentos da função de escala, ϕ , (3.13) em função dos coeficientes do filtro, a_k .

$$\delta_{0,n} = \int_{-\infty}^{\infty} \phi(x) \cdot \phi(x-n) \cdot dx$$

$$\delta_{0,n} = \int_{-\infty}^{\infty} \left(\sqrt{2} \sum_{k=0}^{D-1} a_k \cdot \phi(2x-k) \right) \cdot \left(\sqrt{2} \sum_{l=0}^{D-1} a_l \cdot \phi(2x-2n-l) \right) dx$$

$$\delta_{0,n} = \sum_{k=0}^{D-1} \sum_{l=0}^{D-1} a_k \cdot a_l \cdot \delta_{k-2n,l}$$

$$\delta_{0,n} = \sum_{k=k_1(n)}^{k_2(n)} a_k \cdot a_{k-2n} \quad n \in \mathbb{Z}$$

em que $k_1(n) = \max(0, 2n)$ e $k_2(n) = \min(D-1, D-1+2n)$. Da equação anterior apenas se obtêm $\frac{D}{2}$ equações diferentes, donde,

$$\sum_{k=k_1(n)}^{k_2(n)} a_k \cdot a_{k-2n} = \delta_{0,n} \quad n = 0, 1, \dots, \frac{D}{2}-1 \quad (\text{A.1})$$

$$\sum_{k=k_1(n)}^{k_2(n)} b_k \cdot b_{k-2n} = \delta_{0,n} \quad n = 0, 1, \dots, \frac{D}{2}-1 \quad (\text{A.2})$$

Conservação da Área

Recorda-se que pela equação 3.15, $\int_{-\infty}^{\infty} \phi(x) \cdot dx = 1$. Integrando ambos os membros da equação 3.25 tem-se,

$$\begin{aligned} \int_{-\infty}^{\infty} \phi(x) \cdot dx &= \sqrt{2} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} \phi(2x-k) \cdot dx \\ &= \frac{1}{\sqrt{2}} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} \phi(y) \cdot dy \quad y = 2x - k \end{aligned}$$

pelo que,

$$\sum_{k=0}^{D-1} a_k = \sqrt{2} \quad (\text{A.3})$$

O factor $\sqrt{2}$ é o factor de conservação da área. É um factor de normalização dos coeficientes dos filtros das ôndulas ortogonais e biortogonais, tal que $\sum_k h_k = \sqrt{2}$, em que h_k são os coeficientes dos filtros indicados na tabelas 6.1 (pp. 195) e 8.3 (pp. 279) em [Daub_92].

Momentos nulos

Uma propriedade importante da função de escala ϕ é a possibilidade de representação exacta de polinómios até um determinado grau, $P-1$. Mais concretamente,

$$x^p = \sum_{k=0}^{D-1} M_k^p \cdot \phi(x-k), \quad x \in \mathbb{R}, p = 0, 1, \dots, P-1 \quad (\text{A.4})$$

em que M_k^p é o momento nulo de ordem p , definido por,

$$M_k^p = \int_{-\infty}^{\infty} x^p \cdot \phi(x-k) dx, \quad k \in \mathbb{Z}, p = 0, 1, \dots, P-1 \quad (\text{A.5})$$

Como ϕ e ψ são ortonormais, através do produto interno com a ôndula $\psi(x)$, pode-se modificar a equação A.4, tal que,

$$\int_{-\infty}^{\infty} x^p \cdot \psi(x) \cdot dx = \sum_{k=0}^{D-1} M_k^p \int_{-\infty}^{\infty} \phi(x-k) \cdot \psi(x) \cdot dx = 0$$

e obtém-se,

$$\int_{-\infty}^{\infty} x^p \cdot \psi(x) \cdot dx = 0, \quad x \in \mathbb{R}, p = 0, 1, \dots, P-1 \quad (\text{A.6})$$

Substituindo a equação das bases da ôndula (3.26) na equação anterior (A.6) tem-se,

$$\int_{-\infty}^{\infty} x^p \cdot \psi(x) \cdot dx = \sqrt{2} \sum_{k=0}^{D-1} b_k \int_{-\infty}^{\infty} x^p \cdot \phi(2x-k) \cdot dx = 0$$

$$\frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} b_k \int_{-\infty}^{\infty} (y+k)^p \cdot \phi(y) \cdot dy = 0 \quad y = 2x-k$$

$$\frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} b_k \sum_{l=0}^p \binom{p}{l} k^l \int_{-\infty}^{\infty} y^{p-l} \cdot \phi(y) \cdot dy = 0 \quad (y+k)^p = \sum_{l=0}^p \binom{p}{l} y^{p-l} k^l$$

Nesta última expressão, substituindo o integral pela expressão A.4 obtém-se,

$$\frac{\sqrt{2}}{2^{p+1}} \sum_{l=0}^p \binom{p}{l} M_0^{p-l} \sum_{k=0}^{D-1} b_k k^l = 0 \quad (\text{A.7})$$

Para $p = 0$, usando o princípio da indução matemática, e simplificando a equação, usando novamente a fórmula binomial, obtém-se,

$$\sum_{n=0}^{D-1} (-1)^n a_n n^p = 0, \quad p = 0, 1, \dots, P-1 \quad (\text{A.8})$$

Para este tipo de ôndulas, o número máximo de momentos nulos é $D/2$, em que D é o número de coeficientes dos filtros.

A2) Coeficientes do Filtros da Ôndulas Ortogonais de Daubechies

	n	${}_N h_n$
$N = 2$	0	0,482962913144
	1	0,836516303737
	2	0,224143868042
	3	-0,129409522551
$N = 3$	0	0,332670552950
	1	0,806891509311
	2	0,459877502118
	3	-0,135011020010
	4	-0,085441273882
	5	0,035226291885
$N = 4$	0	0,230377813308
	1	0,714846570553
	2	0,630880767939
	3	-0,027983769416
	4	-0,187034811719
	5	0,030841381835
	6	0,032883011666
	7	-0,010597401785
$N = 5$	0	0,160102397974
	1	0,603829269797
	2	0,724308528437
	3	0,138428145901
	4	-0,242294887066
	5	-0,032244869584
	6	0,077571493840
	7	-0,006241490212
	8	-0,012580751999
	9	0,003335725285

Tabela A.1: Coeficientes dos filtros passa-baixo das ôndulas ortogonais de suporte compacto de Daubechies.

Os coeficientes dos filtros das ôndulas ortogonais de Daubechies são normalizados de modo

que $\sum_n h_n = \sqrt{2}$.

A3) Coeficientes das Ôndulas Biortogonais de Daubechies do tipo *spline*

	n	$\tilde{h}_n = \sqrt{2} \times (\text{coef. } e^{-in\xi} \text{ em } \tilde{m}_0)$	$h_n = \sqrt{2} \times (\text{coef. } e^{-in\xi} \text{ em } m_0)$
$\tilde{N} = 4$ $N = 4$	0	0,852698679007	0,788485616405
	1, -1	0,377402855612	0,418092273221
	2, -2	-0,110624404418	-0,040689417609
	3, -3	-0,023849465019	-0,064538882628
	4, -4	0,037828455507	0
$\tilde{N} = 5$ $N = 5$	0	0,736660181427	0,899506109747
	1, -1	0,345605281955	0,476803265797
	2, -2	-0,054463788468	-0,093504697401
	3, -3	0,007948108637	-0,136706584664
	4, -4	0,039687088347	-0,002694966880
5, -5	0	0,013456709459	
$\tilde{N} = 5$ $N = 5$	0	1,3270252857	0,54113273169
	1, -1	0,471986933790	0,343351739217
	2, -2	-0,363786090098	0,061156453413
	3, -3	-0,118433543197	0,000279879288
	4, -4	0,053826837837	0,021830571333
5, -5	0	0,009921772086	

Tabela A.2: Coeficientes dos filtros passa-baixo de análise e de síntese, para ôndulas biortogonais do tipo *spline*.

Os coeficientes dos filtros passa baixo das ôndulas biortogonais de Daubechies são normalizados de modo que $\sum_n h_n = \sqrt{2}$.

Apêndice B

B – Exemplos do Cálculo da DWT e da DWT^{-1} pelo Esquema Progressivo: Ôndulas DB2 e 9/7

Apresentam-se de seguida alguns exemplos simples da DWT e DWT^{-1} pelo esquema progressivo, aplicados a sequências finitas, considerando os filtros da ôndula DB2 e o par de filtros 9/7.

DWT e DWT^{-1} usando a ôndula DB2:

Considere-se a sequência $x[n]=[2 \ 4 \ 6 \ 8] \ n=0,1,2,3$. Por razões práticas de implementação, consideram-se preenchidos com ‘zero’ os valores anteriores e posteriores, nas regiões limite da sequência. Assim, em termos práticos considera-se a sequência: $x[n]=[0 \ 0 \ 2 \ 4 \ 6 \ 8 \ 0 \ 0] \ n=-2,-1,\dots,4,5$.

Para a DWT:

1ª iteração

Operandos: **Anteriores:** $x_{2l-2} = 0 \ x_{2l-1} = 0$;

Próximos: $x_{2l+2} = 6 \ x_{2l+3} = 8$;

Actuais: $x_{2l} = 2 \ x_{2l+1} = 4$;

$$d_i^{(1)} = x_{2l+1} - \sqrt{3}x_{2l} \approx 4 - (1,732 \times 2) \approx 0,5359$$

$$d_{l+1}^{(1)} \approx 8 - 1,732 \times 6 \approx -2,3923 \rightarrow \text{para a próxima iteração}$$

$$a_i^{(1)} = x_{2l} + \frac{\sqrt{3}}{4}d_i^{(1)} + \frac{\sqrt{3}-2}{4}d_{l+1}^{(1)} \approx 2 + (0,433 \times 0,5359) + [(-0,0669) \times (-2,3923)] \approx 2,3920$$

$$a_{l-1}^{(1)} \approx 0 + 0,433 \times (0) - 0,0669 \times (0,5359) \approx -0,0358$$

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}}a_{l-1}^{(1)} \approx 1,9318 \cdot a_{l-1}^{(1)} \approx 1,9318 \times (-0,0358) \approx -0,069$$

$$d_i^{(2)} = d_i^{(1)} + a_{l-1}^{(1)} \approx 0,5359 - 0,0359 \approx 0,5$$

$$a_i = \frac{\sqrt{3}+1}{\sqrt{2}}a_i^{(1)} \approx 1,9318 \times 2,3920 \approx 4,6208$$

$$d_i = \frac{\sqrt{3}-1}{\sqrt{2}}d_i^{(2)} \approx 0,5176 \times 0,5 \approx 0,2588$$

2ª iteração

Operandos: **Anteriores:** $a_{l-1}^{(1)} = 2,3923$ $d_l^{(1)} = -2,3923$;

Próximos: $x_{2l+2} = 0$ $x_{2l+3} = 0$

Actuais: $x_{2l} = 6$ $x_{2l+1} = 8$;

$$d_i^{(1)} \approx 8 - 1,732 \times 6 \approx -2,3923 \rightarrow \text{da iteração anterior}$$

$$d_{l+1}^{(1)} = 0 - 1,732 \times 0 = 0$$

$$a_i^{(1)} \approx 4 + 0,433 \times (-2,3923) \approx 4,964$$

$$a_{l-1}^{(1)} = 2,392 \rightarrow \text{da iteração anterior}$$

$$d_i^{(2)} = d_i^{(1)} + a_{l-1}^{(1)} \approx -2,3923 + 2,392 = 0$$

$$d_{l+1}^{(2)} = d_{l+1}^{(1)} + a_i^{(1)} \approx 0 + 4,964$$

$$a_l = 1,9318 \cdot a_l^{(1)} \approx 1,9318 \times 4,964 \approx 9,589$$

$$d_l = 0,5176 \cdot d_l^{(2)} = 0$$

$$d_{l+1} \approx 0,5176 \cdot d_{l+1}^{(2)} \approx 0,5176 \times 4,964 \approx 2,569$$

Após cumprimento de todos os passos do esquema progressivo para a DWT obtém-se a sequência: $\hat{X}[k] = [-0,069 \quad 0 \quad 4,6208 \quad 0,2588 \quad 9,589 \quad 0 \quad 0 \quad 2,569]$.

Agrupando os coeficientes de aproximação à esquerda e os coeficientes de detalhe à direita, obtém-se a sequência: $\hat{X}[k] = [-0,069 \quad 4,6208 \quad 9,589 \quad 0 \quad 0 \quad 0,2588 \quad 0 \quad 2,569]$.

Note-se, porém, que uma das características do esquema progressivo, é o facto de não necessitar de memória adicional para o cálculo da transformada, Os valores originais são substituídos pelos coeficientes da transformada em cada iteração. Como tal, desprezam-se os coeficientes relativos às condições limite, inicialmente preenchidas com ‘zero’.

Assim, a sequência válida a considerar como resultado da DWT, resulta do aproveitamento dos valores centrais, tanto para a subsequência formada pelos coeficientes de aproximação, como para a subsequência dos coeficientes de detalhe, isto é, $\hat{X}[k] = [4,6208 \quad 9,589 \quad 0,2588 \quad 0]$.

Para a DWT⁻¹:

Reorganizando a sequência final obtida para a DWT intercalando os coeficientes de aproximação e de detalhe, obtém-se a sequência $\hat{X}[k] = [4,6208 \quad 0,2588 \quad 9,589 \quad 0]$.

Pelas mesmas razões referidas para a DWT, consideram-se preenchidos com ‘zero’ os valores anteriores e posteriores, nas regiões limite da sequência., isto é, considera-se a sequência: $\hat{X}[k] = [0 \quad 0 \quad 4,6208 \quad 0,2588 \quad 9,589 \quad 0 \quad 0 \quad 0]$.

De modo a garantir reconstrução perfeita, convém considerar os coeficientes obtidos na DWT para os casos limite, e que foram desprezados na sequência final. Porém, o facto de terem sido desprezados não constitui um problema, uma vez que podem ser obtidos a partir dos valores da sequência final (útil) por regressão.

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} \times a_{l-1}^{(1)}, \text{ em que } a_{l-1}^{(1)} = d_l^{(2)} - d_l^{(1)}$$

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} (d_l^{(2)} - d_l^{(1)}) = \frac{\sqrt{3}+1}{\sqrt{2}} \left(d_l^{(2)} - \frac{4 \cdot d_l^{(2)}}{4 + \sqrt{3} - 2} \right) = \frac{\sqrt{3}+1}{\sqrt{2}} \cdot d_l^{(2)} \cdot \left(1 - \frac{4}{4 + \sqrt{3} - 2} \right)$$

$$a_{l-1} = \frac{\sqrt{3}+1}{\sqrt{2}} \cdot \frac{\sqrt{2}}{\sqrt{3}-1} \cdot d_l \cdot \left(1 - \frac{4}{4 + \sqrt{3} - 2} \right) \approx -0,2676 \times d_l$$

$$d_{l+1} = \frac{\sqrt{3}-1}{\sqrt{2}} \times d_{l+1}^{(2)}, \text{ em que } d_{l+1}^{(2)} = a_l^{(1)} + d_{l+1}^{(1)}$$

$$d_{l+1} = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot (a_l^{(1)} + d_{l+1}^{(1)}) = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot a_l^{(1)} = \frac{\sqrt{3}-1}{\sqrt{2}} \cdot \frac{\sqrt{2}}{\sqrt{3}+1} \cdot a_l \approx 0,2679 \cdot a_l$$

Assim, tem-se:

$$a_{l-1} \approx -0,2676 \times d_l \approx -0,2676 \times 0,2588 \approx -0,069$$

$$d_{l+1} \approx 0,2679 \cdot a_l \approx 0,2679 \times 9,589 \approx 2,569$$

donde se recupera a sequência $\hat{X}[k] = [-0,069 \quad 0 \quad 4,62 \quad 0,2588 \quad 9,589 \quad 0 \quad 0 \quad 2,569]$.

1ª iteração

Operandos: **Actuais:** $a_l = 4,62 \quad d_l = 0,2588$;

Anteriores: $a_{l-1} = -0,069 \quad d_{l-1} = 0$;

Próximos: $a_{l+1} = 9,589 \quad d_{l+1} = 0$;

$$d_l^{(2)} \approx 1,9318 \cdot d_l \approx 1,9318 \times 0,2588 \approx 0,5$$

$$d_{l+1}^{(2)} \approx 1,9318 \cdot d_{l+1} \approx 1,9318 \times (-1,4498) \approx -2,7989 \rightarrow \text{para a próxima iteração}$$

$$a_l^{(1)} \approx 0,5176 \cdot a_l \approx 0,5176 \times 4,122 \approx 2,1335$$

$$d_l^{(1)} = d_l^{(2)} - a_{l-1}^{(1)} \approx -0,4327$$

$$d_{l+1}^{(1)} = d_{l+1}^{(2)} - a_l^{(1)} \approx -2,7989 - 2,1335 \approx -4,9344 \rightarrow \text{para a próxima iteração}$$

$$x_{2l} \approx a_l^{(1)} - 0,433 \cdot d_l^{(1)} + 0,0669 \cdot d_{l+1}^{(1)}$$

$$x_{2l} \approx 2,1335 - 0,433 \times (-0,4327) + 0,0669 \times (-4,9344) \approx 1,649$$

$$x_{2l+1} \approx d_l^{(1)} + 1,732 \cdot x_{2l} \approx -0,4327 + 1,732 \times 1,649 \approx 2,433$$

2ª iteração

Operandos: **Actuais:** $a_l = 9,589$ $d_l = 0$;

Anteriores: $a_{l-1} = -0,069$ $d_l^{(2)} = -2,7989$;

Próximos: $a_{l+1} = 0$ $d_{l+1} = 2,569$;

$$d_l^{(2)} \approx 1,9318 \cdot d_l \approx 1,9318 \times (-1,4489) \approx -2,7989 \rightarrow \text{da iteração anterior}$$

$$d_{l+1}^{(2)} \approx 1,9318 \cdot d_{l+1} = 0$$

$$a_l^{(1)} \approx 0,5176 \cdot a_l \approx 0,5176 \times 3,605 \approx 1,866$$

$$d_l^{(1)} = d_l^{(2)} - a_{l-1}^{(1)} \approx -2,7989 - 2,1335 \approx -0,6654$$

$$d_{l+1}^{(1)} = d_{l+1}^{(2)} - a_l^{(1)} \approx 0 - 1,866 \approx -1,866$$

$$x_{2l} \approx a_l^{(1)} - 0,433 \cdot d_l^{(1)} + 0,0669 \cdot d_{l+1}^{(1)}$$

$$x_{2l} \approx 1,866 - 0,433 \times (-0,6654) + 0,0669 \times (-1,866) \approx 2,0293$$

$$x_{2l+1} \approx d_l^{(1)} + 1,732 \cdot x_{2l} \approx -0,6654 + 1,732 \times 2,0293 \approx 2,849$$

Obtém-se assim a sequência recuperada $x'[n] = [2,03 \quad 3,98 \quad 5,998 \quad 7,998]$, cujos valores são aproximadamente iguais aos da sequência original ($x[n] = [2 \quad 4 \quad 6 \quad 8]$).

DWT e DWT⁻¹ usando o par de filtros 9/7:

Considere-se a sequência $x[n] = [2 \ 3 \ 4 \ 2]$ $n = 0, 1, 2, 3$. Pelas razões práticas referidas anteriormente, consideram-se preenchidos com ‘zero’ os valores anteriores e posteriores, nas regiões limite da sequência. Note-se que o algoritmo para a DWT com o par de filtros 9/7 é recursivo, com profundidade 2 à esquerda e à direita. Assim, em termos práticos considera-se a sequência: $x[n] = [0 \ 0 \ 0 \ 0 \ 2 \ 3 \ 4 \ 2 \ 0 \ 0 \ 0 \ 0]$ $n = -4, -3, \dots, 6, 7$.

Para a DWT:

1ª iteração

Operandos: **Actuais:** $x_{2l} = a_l^{(0)} = 2$ $x_{2l+1} = d_l^{(0)} = 3$;

Anteriores: $x_{2l-4} = a_{l-2}^{(0)} = 0$ $x_{2l-3} = d_{l-2}^{(0)} = 0$ $x_{2l-2} = a_{l-1}^{(0)} = 0$ $x_{2l-1} = d_{l-1}^{(0)} = 0$;

Próximos: $x_{2l+2} = a_{l+1}^{(0)} = 4$ $x_{2l+3} = d_{l+1}^{(0)} = 2$ $x_{2l+4} = a_{l+2}^{(0)} = 0$ $x_{2l+5} = d_{l+2}^{(0)} = 0$;

$$d_l^{(1)} = d_l^{(0)} + \alpha \cdot (a_l^{(0)} + a_{l+1}^{(0)}) \approx 3 - 1,58613 \times (2 + 4) \approx -6,5168$$

$$a_l^{(1)} = a_l^{(0)} + \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \approx 2 - 0,0529 \times (-6,5168 + d_{l-1}^{(1)})$$



calc. aux: $d_{l-1}^{(1)} = d_{l-1}^{(0)} + \alpha \cdot (a_{l-1}^{(0)} + a_l^{(0)}) \approx -1,5861 \times (2) \approx -3,1722$

$$a_l^{(1)} = a_l^{(0)} + \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \approx 2 - 0,0529 \times (-6,5168 - 3,1722) \approx 2,5133$$

$$d_l^{(2)} = d_l^{(1)} + \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \approx -6,5168 + 0,8829 \times (2,5133 + a_{l+1}^{(1)})$$



calc. aux: $a_{l+1}^{(1)} = a_{l+1}^{(0)} + \beta \cdot (d_{l+1}^{(1)} + d_l^{(1)}) \approx 4 - 0,0529 \times (d_{l+1}^{(1)} - 6,5168)$



calc. aux: $d_{l+1}^{(1)} = d_{l+1}^{(0)} + \alpha \cdot (a_{l+1}^{(0)} + a_{l+2}^{(0)}) \approx 2 - 1,5864 \times (4) \approx -4,3445 \rightarrow$ próx. iter.

$a_{l+1}^{(1)} = a_{l+1}^{(0)} + \beta \cdot (d_{l+1}^{(1)} + d_l^{(1)}) = 4 - 0,0529 \times (-4,3445 - 6,5168) \approx 4,5754 \rightarrow$ próx. iter.

$$d_l^{(2)} = d_l^{(1)} + \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \approx -6,5168 + 0,8829 \times (2,5133 + 4,5754) \approx -0,258$$

$$a_i^{(2)} = a_i^{(1)} + \delta \cdot (d_i^{(2)} + d_{i-1}^{(2)}) \approx 2,5133 + 0,4435 \times (-0,258 + d_{i-1}^{(2)})$$

$$\text{calc. aux: } d_{i-1}^{(2)} = d_{i-1}^{(1)} + \gamma \cdot (a_{i-1}^{(1)} + a_i^{(1)}) \approx -3,1722 + 0,8829 \times (a_{i-1}^{(1)} + 2,5133)$$

$$\text{calc. aux: } a_{i-1}^{(1)} = a_{i-1}^{(0)} + \beta \cdot (d_{i-1}^{(1)} + d_{i-2}^{(1)}) = \beta \cdot (d_{i-1}^{(1)}) \approx -0,0529 \times (-3,1722) \approx 0,168$$

$$d_{i-1}^{(2)} = d_{i-1}^{(1)} + \gamma \cdot (a_{i-1}^{(1)} + a_i^{(1)}) \approx -3,1722 + 0,8829 \times (0,168 + 2,5133) \approx -0,8048$$

$$a_i^{(2)} = a_i^{(1)} + \delta \cdot (d_i^{(2)} + d_{i-1}^{(2)}) \approx 2,5133 + 0,4435 \times (-0,258 - 0,8048) \approx 2,0419$$

$$a_i = \varphi \cdot a_i^{(2)} \approx 1,1496 \times 2,0419 \approx 2,3474$$

$$d_i = \frac{1}{\varphi} \cdot d_i^{(2)} \approx 0,8698 \times (-0,258) \approx -0,2244$$

Obtém-se ainda os valores finais das posições anteriores, em resultado das condições limite.

$$a_{i-1}^{(1)} = 0,168 \text{ (já calculado)}$$

$$a_{i-1}^{(2)} = a_{i-1}^{(1)} + \delta \cdot (d_{i-1}^{(1)} + d_{i-2}^{(1)}) \approx 0,168 + 0,4435 \times (-0,8048) \approx -0,1888$$

$$a_{i-1} = \varphi \cdot a_{i-1}^{(2)} \approx 1,1496 \times (-0,1888) \approx -0,2171$$

$$d_{i-1}^{(1)} \approx -3,1722$$

$$d_{i-1}^{(2)} = d_{i-1}^{(1)} + \gamma \cdot (a_{i-1}^{(1)} + a_i^{(1)}) \approx -3,1722 + 0,8829 \times (0,168 + 2,5133) \approx -0,8048$$

$$d_{i-1} = \frac{1}{\varphi} \cdot d_{i-1}^{(2)} \approx 0,86986 \times (-0,8048) \approx -0,7$$

2ª iteração

$$\text{Operandos: } \underline{\text{Actuais:}} \quad x_{2l+2} = a_l^{(0)} = 4 \quad x_{2l+3} = d_l^{(0)} = 2;$$

$$\underline{\text{Anteriores:}} \quad x_{2l-2} = a_{l-2}^{(0)} = 0 \quad x_{2l-1} = d_{l-2}^{(0)} = 0 \quad x_{2l} = a_{l-1}^{(0)} = 0 \quad x_{2l+1} = d_{l-1}^{(0)} = 0;$$

$$\underline{\text{Próximos:}} \quad x_{2l+4} = a_{l+1}^{(0)} = 0 \quad x_{2l+5} = d_{l+1}^{(0)} = 0 \quad x_{2l+6} = a_{l+2}^{(0)} = 0 \quad x_{2l+7} = d_{l+2}^{(0)} = 0;$$

$$d_l^{(1)} = -4,34457 \rightarrow \underline{\text{da iteração anterior}}$$

$$a_i^{(1)} = 4,5754 \rightarrow \text{da iteração anterior}$$

$$d_i^{(2)} = d_i^{(1)} + \gamma \cdot (a_i^{(1)} + a_{i+1}^{(1)}) \approx -4,3445 + 0,8829 \times (4,5754 + a_{i+1}^{(1)})$$



$$\text{calc.aux: } a_{i+1}^{(1)} = a_{i+1}^{(0)} + \beta \cdot (d_{i+1}^{(1)} + d_i^{(1)}) \approx -0,0529 \times (-4,3445) \approx 0,2301 \rightarrow \text{próx. iteração}$$

$$d_i^{(2)} = d_i^{(1)} + \gamma \cdot (a_i^{(1)} + a_{i+1}^{(1)}) \approx -4,3445 + 0,8829 \times (4,5754 + 0,23017) \approx -0,1016$$

$$a_i^{(2)} = a_i^{(1)} + \delta \cdot (d_i^{(2)} + d_{i-1}^{(2)}) \approx 4,5754 + 0,4435 \times (-0,1016 - 0,258) \approx 4,4159$$

$$a_i = \varphi \cdot a_i^{(2)} \approx 1,1496 \times 4,4159 \approx 5,0765$$

$$d_i = \frac{1}{\varphi} \cdot d_i^{(2)} \approx 0,8698 \times (-0,1016) \approx -0,0884$$

Obtém-se ainda os valores finais das posições seguintes, em resultado das condições limite.

$$a_{i+1}^{(0)} \approx 0,2301 \rightarrow \text{calculado nesta iteração}$$

$$a_{i+1}^{(2)} = a_{i+1}^{(1)} + \delta \cdot (d_{i+1}^{(1)} + d_i^{(1)}) \approx 0,2301 + 0,4435 \times (-0,1016) \approx 0,1851$$

$$a_{i-1} = \varphi \cdot a_{i-1}^{(2)} \approx 1,149 \times (0,1851) \approx 0,2127$$

$$d_{i+1}^{(1)} = 0 \rightarrow \text{calculado nesta iteração}$$

$$d_{i+1}^{(2)} = d_{i+1}^{(1)} + \gamma \cdot (a_{i+1}^{(1)} + a_{i+2}^{(1)}) = 0,8829 \times (0) = 0$$

$$d_{i-1} = \frac{1}{\varphi} \cdot d_{i-1}^{(2)} = 0,8698 \times (0) = 0$$

Após cumprimento de todos os passos do esquema progressivo para a DWT obtém-se a sequência:

$$\hat{X}[k] = [0 \quad 0 \quad -0,2171 \quad -0,7 \quad 2,3474 \quad -0,2244 \quad 5,0765 \quad -0,0884 \quad 0,2127 \quad 0 \quad 0 \quad 0]$$

Tal como no caso anterior, os coeficientes de aproximação e detalhe apresentam-se intercalados. Agrupando todos os coeficientes de aproximação à esquerda e todos os coeficientes de detalhe à direita, obtém-se a sequência:

$$\hat{X}[k] = [0 \quad -0,2171 \quad 2,3474 \quad 5,0765 \quad 0,2127 \quad 0 \quad 0 \quad -0,7 \quad -0,2244 \quad -0,0884 \quad 0 \quad 0]$$

Pelas mesmas razões referidas no caso anterior, a sequência válida a considerar na DWT resulta do aproveitamento dos valores centrais, tanto para a subsequência formada pelos coeficientes de aproximação, como para a subsequência formada pelos coeficientes de detalhe, isto é, $\hat{X}[k] = [2,3474 \quad 5,0765 \quad -0,2244 \quad -0,0884]$.

Para a DWT⁻¹:

Reorganizando a sequência da DWT, intercalando os coeficientes de aproximação e de detalhe, obtém-se a sequência $\hat{X}[k] = [2,3474 \quad -0,2244 \quad 5,0765 \quad 0,0884]$.

De modo a garantir reconstrução perfeita, convém considerar os valores limite obtidos na DWT. Porém, no caso destas ondas em particular, devido à forma encadeada do algoritmo pelo esquema progressivo, não é simples encontrar os coeficientes nas regiões limite por regressão. Caso não se considerem os coeficientes das regiões limite surgem erros, que no caso da DWT de imagens em vários níveis se propagam, e se manifestam de forma significativa nas regiões limite. Este problema manifesta-se sobretudo em esquemas que não recorram a memória adicional, como é o caso do esquema progressivo.

Uma das possibilidades para solucionar este problema, consiste em considerar a extensão dos sinais nas regiões limite, tanto para a DWT, como para a DWT⁻¹, tal como foi realizado nesta tese, e referido no capítulo 4. Porém, neste caso em particular, como o objectivo deste apêndice é apenas demonstrar o cálculo da DWT e da DWT⁻¹, considera-se que os coeficientes relativos aos casos limite são de algum modo “conhecidos” pelo decodificador e que na primeira iteração são: $d_{l-1}^{(2)} = -0,8048$. Nos casos em que se realiza expansão da sequência, como é o caso nesta tese, este problema apenas se coloca nos coeficientes nas regiões limite (na extensão), e não nas regiões da sequência, ficando, portanto, solucionado.

Pelas mesmas razões referidas anteriormente, consideram-se preenchidos com ‘zero’ os valores anteriores e posteriores, nas regiões limite da sequência., isto é, considera-se a sequência:

$$\hat{X}[k] = [0 \quad 0 \quad 0 \quad 0 \quad 2,3474 \quad -0,2244 \quad 5,0765 \quad -0,0884 \quad 0 \quad 0 \quad 0 \quad 0] \quad k = -4, -3, \dots, 0, 1$$

1ª iteração

Operandos: Actuais: $\hat{X}[0] = a_l = 2,3474 \quad \hat{X}[1] = d_l = -0,2244$;

Anteriores: $\hat{X}[-4] = a_{l-2} = 0$ $\hat{X}[-3] = d_{l-2} = 0$ $\hat{X}[-2] = a_{l-1} = 0$ $\hat{X}[-1] = d_{l-1} = 0$;

Próximos: $\hat{X}[2] = a_{l+1} = 5,0765$ $\hat{X}[3] = d_{l+1} = -0,0884$ $\hat{X}[4] = a_{l+2} = 0$ $\hat{X}[5] = d_{l+2} = 0$;

$$d_l^{(2)} = \varphi \cdot d_l \approx 1,1496 \times (-0,2244) \approx -0,258$$

$$a_l^{(2)} = \frac{1}{\varphi} \cdot a_l \approx 0,8698 \times (2,3474) \approx 2,0419$$

$$d_{l-1}^{(2)} = -0,8048$$

$$a_l^{(1)} = a_l^{(1)} - \delta \cdot (d_l^{(2)} + d_{l-1}^{(2)}) \approx 2,0419 - 0,4435 \times (-0,258 - 0,8048) \approx 2,5133$$

$$d_l^{(1)} = d_l^{(2)} - \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \approx -0,258 - 0,8829 \times (2,5133 + a_{l+1}^{(1)})$$

$$\text{c. aux: } a_{l+1}^{(1)} = a_{l+1}^{(2)} - \delta \cdot (d_{l+1}^{(2)} + d_l^{(2)}) \approx a_{l+1}^{(2)} - 0,4435 \times (d_{l+1}^{(2)} - 0,258)$$

$$\text{c. aux: } a_{l+1}^{(2)} = \frac{1}{\varphi} \cdot a_{l+1} \approx 0,8698 \times (5,0765) \approx 4,4159 \quad \text{próx. iteração}$$

$$d_{l+1}^{(2)} = \varphi \cdot d_{l+1} \approx 1,1496 \times (-0,0884) \approx -0,1016 \quad \text{próx. iteração}$$

$$a_{l+1}^{(1)} = a_{l+1}^{(1)} - \delta \cdot (d_{l+1}^{(2)} + d_l^{(2)}) \approx 4,4159 - 0,4435 \times (-0,1016 - 0,258) \approx 4,5754$$

$$d_l^{(1)} = d_l^{(2)} - \gamma \cdot (a_l^{(1)} + a_{l+1}^{(1)}) \approx -0,258 - 0,8829 \times (2,5133 + 4,5754) \approx -6,5167$$

$$a_l^{(0)} = a_l^{(1)} - \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \approx 2,2844 + 0,0529 \times (-6,5167 + d_{l-1}^{(1)})$$

$$a_{l-1}^{(2)} = -0,1888$$

$$\text{c. aux: } d_{l-1}^{(1)} = d_{l-1}^{(2)} - \gamma \cdot (a_{l-1}^{(1)} + a_l^{(1)}) \approx 0,8048 - 0,8829 \times (a_{l-1}^{(1)} + 2,5133)$$

$$\text{c. aux: } a_{l-1}^{(1)} = a_{l-1}^{(2)} - \gamma \cdot (d_{l-1}^{(2)} + d_{l-2}^{(2)}) = -0,18886 - 0,4435 \times (-0,8048) = 0,16807$$

$$d_{l-1}^{(1)} = d_{l-1}^{(2)} - \gamma \cdot (a_{l-1}^{(1)} + a_l^{(1)}) \approx 0,8048 - 0,8829 \times (0,168 + 2,5133) \approx -3,1722$$

$$a_l^{(0)} = a_l^{(1)} - \beta \cdot (d_l^{(1)} + d_{l-1}^{(1)}) \approx 2,2844 + 0,0529 \times (-6,5167 - 3,1722) \approx 1,9999 \approx 2$$

$$d_i^{(0)} = d_i^{(1)} - \alpha \cdot (a_i^{(0)} + a_{i+1}^{(0)}) \approx -6,5167 + 1,5861 \times (2 + a_{i+1}^{(0)})$$

$$\begin{array}{l} \downarrow \\ \text{c. aux: } a_{i+1}^{(0)} = a_{i+1}^{(1)} - \beta \cdot (d_{i+1}^{(1)} + d_i^{(1)}) \approx a_{i+1}^{(1)} + 0,0529 \times (d_{i+1}^{(1)} + d_i^{(1)}) \\ \downarrow \\ \text{c. aux: } d_{i+1}^{(1)} = d_{i+1}^{(2)} - \gamma \cdot (a_{i+1}^{(1)} + a_{i+2}^{(1)}) \approx -0,1016 - \gamma \cdot (a_{i+1}^{(1)} + a_{i+2}^{(1)}) \\ \downarrow \\ \text{c. aux: } a_{i+2}^{(1)} = a_{i+2}^{(2)} - \delta \cdot (d_{i+2}^{(2)} + d_{i+1}^{(2)}) \approx -\delta \cdot (d_{i+1}^{(2)}) \approx 0,045 \\ \text{dado que } a_{i+2}^{(2)} = 0 \quad d_{i+2}^{(2)} = 0, \text{ devido às condições limite.} \\ \downarrow \\ d_{i+1}^{(1)} \approx -0,10164 - 0,8829 \times (4,5754 + 0,045) \approx -4,1811 \\ \downarrow \\ a_{i+1}^{(0)} = a_{i+1}^{(1)} - \beta \cdot (d_{i+1}^{(1)} + d_i^{(1)}) \approx 4,5754 + 0,0529 \times (-4,1811 - 6,5167) \approx 4,0086 \approx 4 \\ \downarrow \\ d_i^{(0)} = d_i^{(1)} - \alpha \cdot (a_i^{(0)} + a_{i+1}^{(0)}) \approx -6,5167 + 1,5861 \times (2 + 4) \approx 3 \end{array}$$

2ª iteração

Operandos: Actuais: $\hat{X}[0] = a_i = 5,0765$ $\hat{X}[1] = d_i = -0,0884$;

Próximos: $\hat{X}[2] = a_{i+1} = 0$ $\hat{X}[3] = d_{i+1} = 0$ $\hat{X}[4] = a_{i+2} = 0$ $\hat{X}[5] = d_{i+2} = 0$;

$$d_i^{(0)} = d_i^{(1)} - \alpha \cdot (a_i^{(0)} + a_{i+1}^{(0)}) \approx -4,1811 + 1,5861 \times (4 + a_{i+1}^{(0)})$$

$$\text{c. aux. } a_{i+1}^{(0)} = a_{i+1}^{(1)} - \beta \cdot (d_{i+1}^{(1)} + d_i^{(1)}) \approx 0 + 0,0529 \times (0 - 4,1811) \approx -0,2215$$

$$d_i^{(0)} = d_i^{(1)} - \alpha \cdot (a_i^{(0)} + a_{i+1}^{(0)}) \approx -4,1811 + 1,5861 \times (4 - 0,2215) \approx 2$$

Após cumprimento de todos os passos do esquema progressivo para a DWT obtém-se a sequência, $\tilde{X}[k] \approx [0 \ 0 \ 0 \ 0 \ 2 \ 3 \ 4 \ 2 \ 0 \ 0 \ 0 \ 0]$. Considerando apenas os valores centrais, isto é, ignorando os valores das condições limite, recupera-se a sequência $\tilde{X}[k] \approx [2 \ 3 \ 4 \ 2]$, que corresponde a uma réplica da sequência original.

Apêndice C

C – Tabelas Usadas na Codificação de Huffman

Apresentam-se de seguida as tabelas usadas na codificação de Huffman, no CoDec desenvolvido neste trabalho.

Na primeira tabela estabelece-se a dimensão mínima da palavra binária para representação do valor do coeficiente. A dimensão obtida é usada para endereçar as posições da tabela seguinte, para a obtenção da representação final do código VLC. As duas tabelas são usadas na codificação de Huffman na norma JPEG *baseline*, sendo a segunda tabela referente à codificação dos coeficientes relativos às componentes de luminância.

Valor do coeficiente	Dimensão (<i>SIZE</i>)
0	0
$[-1] \cup [1]$	1
$[-3, -2] \cup [2, 3]$	2
$[-7, -4] \cup [4, 7]$	3
$[-15, -8] \cup [8, 15]$	4
$[-31, -16] \cup [16, 31]$	5
$[-63, -32] \cup [32, 63]$	6
$[-127, -64] \cup [64, 127]$	7
$[-255, -128] \cup [128, 255]$	8
$[-511, -256] \cup [256, 511]$	9
$[-1023, -512] \cup [512, 1023]$	10
$[-2047, -1024] \cup [1024, 2047]$	11
$[-4095, -2048] \cup [2048, 4095]$	12
$[-8191, -4096] \cup [4096, 8191]$	13
$[-16383, -8192] \cup [8192, 16383]$	14
$[-32767, -16384] \cup [16384, 32767]$	15
32768	16

Run/Size	Comp	Código VLC
0/0	4	1010 (EOB)
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	111111110000010
0/A	16	111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	111111110000100
1/7	16	111111110000101
1/8	16	111111110000110
1/9	16	111111110000111
1/A	16	111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	111111110001001
2/6	16	111111110001010
2/7	16	111111110001011
2/8	16	111111110001100
2/9	16	111111110001101
2/A	16	111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	111111110001111
3/5	16	111111110010000
3/6	16	111111110010001
3/7	16	111111110010010
3/8	16	111111110010011
3/9	16	111111110010100
3/A	16	111111110010101
4/1	6	111011
4/2	10	1111111000
4/3	16	111111110010110
4/4	16	111111110010111
4/5	16	111111110011000
4/6	16	111111110011001
4/7	16	111111110011010
4/8	16	111111110011011
4/9	16	111111110011100
4/A	16	111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	111111110011110
5/4	16	111111110011111
5/5	16	111111110100000
5/6	16	111111110100001
5/7	16	111111110100010
5/8	16	111111110100011
5/9	16	111111110100100
5/A	16	111111110100101

Run/Size	Comp	Código VLC
6/1	7	1111011
6/2	11	111111110110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	111111110111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011
7/9	16	1111111110110100
7/A	16	1111111110110101
8/1	9	111111000
8/2	15	11111111000000
8/3	16	1111111110110110
8/4	16	1111111110110111
8/5	16	1111111110111000
8/6	16	1111111110111001
8/7	16	1111111110111010
8/8	16	1111111110111011
8/9	16	1111111110111100
8/A	16	1111111110111101
9/1	9	111111001
9/2	16	1111111110111110
9/3	16	1111111110111111
9/4	16	111111111000000
9/5	16	111111111000001
9/6	16	111111111000010
9/7	16	111111111000011
9/8	16	111111111000100
9/9	16	111111111000101
9/A	16	111111111000110
A/1	9	111111010
A/2	16	111111111000111
A/3	16	111111111001000
A/4	16	111111111001001
A/5	16	111111111001010
A/6	16	111111111001011
A/7	16	111111111001100
A/8	16	111111111001101
A/9	16	111111111001110
A/A	16	111111111001111
B/1	10	1111111001
B/2	16	111111111010000
B/3	16	111111111010001
B/4	16	111111111010010
B/5	16	111111111010011
B/6	16	111111111010100
B/7	16	111111111010101
B/8	16	111111111010110
B/9	16	111111111010111
B/A	16	111111111011000

Run/Size	Comp	Código VLC
C/1	10	1111111010
C/2	16	111111111011001
C/3	16	111111111011010
C/4	16	111111111011011
C/5	16	111111111011100
C/6	16	111111111011101
C/7	16	111111111011110
C/8	16	111111111011111
C/9	16	111111111100000
C/A	16	111111111100001
D/1	11	11111111000
D/2	16	111111111100010
D/3	16	111111111100011
D/5	16	111111111100101
D/6	16	111111111100110
D/7	16	111111111100111
D/8	16	111111111101000
D/9	16	111111111101001
D/A	16	111111111101010
E/1	16	111111111101011
E/2	16	111111111101100
E/3	16	111111111101101
E/4	16	111111111101110
E/5	16	111111111101111
E/6	16	111111111110000
E/7	16	111111111110001
E/8	16	111111111110100
E/9	16	111111111110101
E/A	16	111111111110100
F/0	11	11111111001
F/1	16	111111111110101
F/2	16	111111111110110
F/3	16	111111111110111
F/4	16	111111111111000
F/5	16	111111111111001
F/6	16	111111111111010
F/7	16	111111111111011
F/8	16	111111111111100
F/9	16	111111111111101
F/A	16	111111111111110

Nota. Valores apresentados em hexadecimal.

