



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações



**CONTRIBUIÇÕES PARA PÓS PROCESSAMENTO DA TRANSFORMADA
WAVELET NA CODIFICAÇÃO ROI E SPIHT COM APLICAÇÃO NA
TRANSMISSÃO DE IMAGENS.**

Roger Fredy Larico Chávez
Orientador: Prof. Dr. Yuzo Iano

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.
Área de concentração: Telecomunicações e Telemática.

Banca Examinadora

Yuzo Iano..... Decom/Feec/Unicamp
Evaldo Gonçalves Pelaes UFPA
Antônio Cláudio Paschoarelli Veiga..... UFU
Andre Leon Sampaio Gradvohl FT/Unicamp
Luiz César Martini Decom/Feec/Unicamp

Campinas, SP
Janeiro/2012

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

L324c	<p>Larico Chávez, Roger Fredy</p> <p>Contribuições para pós processamento da transformada wavelet na codificação roi e spiht com aplicação na transmissão de imagens / Roger Fredy Larico Chávez. --Campinas, SP: [s.n.], 2012.</p> <p>Orientador: Yuzo Iano.</p> <p>Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Compressão de imagens. 2. Wavelet (Matemática). 3. Processamento de imagens. I). Iano, Yuzo. II). Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III). Título.</p>
-------	--

Título em Inglês: *Contributions for post processing of wavelet transform with SPIHT ROI coding and application in the transmission of images.*

Palavras-chave em Inglês: *image compression, wavelet (mathematical), image processing.*

Área de concentração: Telecomunicações e Telemática

Titulação: Doutor em Engenharia Elétrica.

Banca examinadora: Evaldo Gonçalves Pelaes, Antônio Cláudio Paschoarelli
Veiga, Andre Leon Sampaio Gradvohl, Luiz César Martini.

Data da defesa: 31-01-2012

Programa de Pós-Graduação: Engenharia Elétrica.

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Roger Fredy Larico Chávez

Data da Defesa: 31 de janeiro de 2012

Título da Tese: "Contribuições para pós processamento da transformada wavelet na codificação ROI e SPIHT com aplicação na transmissão de imagens"

Prof. Dr. Yuzo Iano (Presidente): Yuzo Iano

Prof. Dr. Antônio Cláudio Paschoarelli Veiga: Antonio C. Veiga

Prof. Dr. Evaldo Gonçalves Pelaes: Evaldo Gonçalves Pelaes

Prof. Dr. Andre Leon Sampaio Gradvohl: A. L. S. Gradvohl

Prof. Dr. Luiz César Martini: Luiz Cesar Martini

Resumo

A área que trata de compressão de imagem com perdas é, atualmente, de grande importância. Isso se deve ao fato de que as técnicas de compressão permitem representar de uma forma eficiente uma imagem reduzindo assim, o espaço necessário para armazenamento ou um posterior envio da imagem através de um canal de comunicações. Em particular, o algoritmo SPIHT (*Set Partitioning of Hierarchical Trees*) muito usado em compressão de imagens é de implementação simples e pode ser aproveitado em aplicações onde se requer uma baixa complexidade. Este trabalho propõe um esquema de compressão de imagens utilizando uma forma personalizada de armazenamento da transformada DWT (*Discrete Wavelet Transform*), codificação flexível da ROI (*Region Of Interest*) e a compressão de imagens usando o algoritmo SPIHT. A aplicação consiste na transmissão dos dados correspondentes usando-se codificação turbo. A forma personalizada de armazenamento da DWT visa um melhor aproveitamento da memória por meio do uso de algoritmo SPIHT. A codificação ROI genérica é aplicada em um nível alto da decomposição DWT. Nesse ponto, o algoritmo SPIHT serve para ressaltar e transmitir com prioridade as regiões de interesse. Os dados a serem transmitidos, visando o menor custo de processamento, são codificados com um esquema turbo convolucional. Isso porque esse esquema é de implementação simples no que concerne à codificação. A simulação é implementada em módulos separados e reutilizáveis para esta pesquisa. Os resultados das simulações mostram que o esquema proposto é uma solução que diminui a quantidade de memória utilizada bem como o custo computacional para aplicações de envio de imagens em aplicações como transmissão de imagens via satélite, radiodifusão e outras mídias.

Palavras-chave: compressão de imagens, transformada *wavelet*, redução de memória, algoritmo SPIHT, região de interesse, codificação ROI, processamento de imagens.

Abstract

Nowadays, the area that comes to lossy image compression is really important. This is due to the fact that compression techniques allow an efficient way to represent an image thereby reducing the space required for storage or subsequent submission of an image through a communications channel. In particular, the algorithm SPIHT (Set Partitioning of Hierarchical Trees) widely used in image compression is simple to implement and can be used in applications where a low complexity is required. This study proposes an image compression scheme using a personalized storage transform DWT (Discrete Wavelet Transform), encoding flexible ROI (Region Of Interest) and image compression algorithm using SPIHT. The application consists in a transmission of the corresponding data using turbo coding. The shape of the custom storage DWT aims to make better use of memory by reducing the amount of memory through the use of SPIHT algorithm. ROI coding is applied in a generic high-level DWT decomposition. At this point, the algorithm serves to highlight SPITH and transmit the priority areas of interest. The data to be transmitted in order to lower the cost of processing are encoded with a turbo convolutional scheme. This is due this scheme is simple to implement with regard to coding. The simulation is implemented in separate modules and reusable for this research. The simulations and analysis show that the proposed scheme is a solution that decreases the amount of memory used and the computational cost for applications to send images in applications such as image transmission via satellite, broadcasting and others medias.

Keywords: *image compression, wavelet transform, reduced memory, algorithm SPIHT, region-of-interest, ROI code, image processing.*

*Ao Deus **Jeová** por tudo que me permitiu fazer.*

Agradecimentos

Ao meu orientador, Prof. Yuzo Iano sou grato pela orientação, conselhos e compreensão.

Aos meus pais, minha família e meus amigos pelo apoio durante esta jornada.

Aos integrantes dos grupos de pesquisa formados durante a minha vida acadêmica dentro e fora do Laboratório de Comunicações Visuais (LCV). Aos demais colegas de pós-graduação, pelas sugestões, ajudas, conselhos e amizades.

Ao convênio ITA-Unicamp agradeço especialmente. Agradeço ao Prof. Dr. Osamu Saotome, Prof. Dr. David Fernandes e a todas as instituições e pessoas envolvidas no projeto ITASAT, pela estrutura física e suporte financeiro disponibilizado. Também sou grato aos coordenadores e integrantes do grupo Unicamp-ITASAT.

Meus agradecimentos às agências financiadoras do grupo de pesquisa do LCV como Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - "*Coordination for the Improvement of Higher Level Personnel*"), Capes RH-TVD (Programa de Formação de Recursos Humanos em Televisão Digital - "*Human Resources in Digital Television*"), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - "*National Counsel of Technological and Scientific Development*"), Fapesp (Fundação de Amparo à Pesquisa do Estado de São Paulo - "*São Paulo State Research Foundation*"), Faepex/Unicamp (Fundo de Apoio ao Ensino, à Pesquisa e Extensão - "*Fund for the Support of Teaching, Research and Extension*"), RNP/CTIC (Rede Nacional de Pesquisas/Programa Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação - "*National Research Network/The Program Research and Development Center in Digital Technologies for Information and Communication*").

Agradecimentos

Agradecimentos especiais ao programa CAPES RH-TVD da **Coordenação de Aperfeiçoamento de Pessoal de Nível Superior** tanto pelo apoio financeiro quanto pelo incentivo acadêmico para que este trabalho pudesse ser realizado.

Agradecimentos

Agradecimentos especiais ao programa **ITASAT** e a **AEB** (Agência Espacial Brasileira) tanto pelo apoio financeiro quanto pelo incentivo acadêmico para que este trabalho pudesse ser realizado.

Sumário

1. Introdução.....	1
2. Compressão de imagens utilizando transformada <i>wavelet</i>.....	7
2.1 Introdução.....	7
2.2 Transformada <i>wavelet</i>	8
2.2.1 Transformada <i>wavelet</i> contínua TWC.....	8
2.2.2 Transformada <i>wavelet</i> discreta TWD.....	10
2.3 Algoritmos de compressão de imagens	13
2.4 CCSDS: Recomendação para compressão de imagens	13
2.4.1 Transformada <i>wavelet</i> discreta no CCSDS	14
2.4.2 Codificador de planos de <i>bits</i> (BPE)	15
2.4.3 Algoritmo Rice	16
2.4.4 TER - Recomendação para compressão de imagens CCSDS	17
2.5 JPEG-2000.....	18
2.5.1 Processo de codificação.....	19
2.6 SPIHT (<i>Set Partitioning In Hierarchical Trees</i>).....	22
2.6.1 Algoritmo SPIHT	23
2.7 Simulações e resultados.....	24
2.8 Considerações finais.....	27
3. Pós-processamento da transformada <i>wavelet</i> para diminuir a memória da codificação SPIHT.....	29
3.1 Introdução.....	29
3.2 Transformada <i>wavelet</i> com pós-processamento.....	31
3.3 Codificação SPIHT.....	36
3.4 Simulações e resultados.....	38
4. Codificação de região de interesse generalizada aplicada ao SPIHT	41
4.1 Introdução.....	41
4.2 Codificação ROI no JPEG-2000	44
4.3 Codificação da ROI no SPIHT	45
4.4 Codificação generalizada da ROI.....	46
4.5 Geração da máscara da ROI	47
4.5.1 Algoritmo proposto	49
4.6 Simulações e resultados.....	51
5. Comparações turbo produto TPC, turbo convolucional TCC e códigos LDPC para aplicações de baixa potência	59
5.1 Introdução.....	59
5.2 Código turbo convolucional TCC	61

5.3	Código turbo produto TPC	63
5.4	Códigos LDPC (<i>Low Density Parity Check Codes</i>).....	66
5.5	Simulações e resultados.....	67
5.5.1	Configuração de simulações turbo código convolucional (TCC)	67
5.5.2	Configuração das simulações código turbo produto (TPC)	68
5.5.3	Configuração das simulações do código LDPC	70
5.5.4	Comparação entre diversas configurações dos códigos	74
6.	Conclusões e trabalhos futuros	77
6.1	Conclusões.....	77
6.2	Sugestões para trabalhos futuros	78
7.	Referências bibliográficas.....	81
Apêndice A: Visão geral de aplicações de imagens em pequenos satélites e o projeto ITASAT.....		97
A.1	Introdução.....	97
A.2	Projeto ITASAT	98
A.3	Aplicações de imagens em pequenos satélites	98
A.4	Contexto: Aplicações comerciais de pequenos satélites	99
	Pequenos satélites no Brasil.....	100
Apêndice B: Criação automática de vetores binários.....		103
Apêndice C: Aplicações da pesquisa na TV digital.....		105
C.1	TV digital: visão geral.....	105
C.2	Aplicação das pesquisas na TV digital.....	109

Lista de Figuras

Fig. 1.1. Esquema de transmissão de imagens.....	5
Fig. 2.1. Interpretação da TWC por banco de filtros.	9
Fig. 2.2. Quadrantes de frequência da TWD aplicada a uma imagem; a) os quatro quadrantes <i>LL</i> , <i>HL</i> , <i>LH</i> e <i>HH</i> ; b) imagem original; c) TWD da imagem em um nível.....	11
Fig. 2.3. Intensidades dos coeficientes da TWD com 1, 2 e 3 níveis.	12
Fig. 2.4. Dois blocos básicos nos algoritmos de compressão recomendado pela CCSDS.	14
Fig. 2.5. Decomposição TWD 2D em três níveis de uma imagem [51].	14
Fig. 2.6. Esquema da TWD de uma imagem para o uso do codificador de planos de <i>bits</i> BPE [51].	16
Fig. 2.7. Visão geral do processo de compressão JPEG-2000 [42].	19
Fig. 2.8. Diagrama de blocos e de documentos próprios da especificação JPEG-2000 [63].....	20
Fig. 2.9. a) Diagrama de fluxo do JPEG-2000 (<i>Intel IPP - Codec Process Flow</i> [58]); b) Um algoritmo do JPEG-2000 visualizado em [59].	21
Fig. 2.10. Desempenho em PSNR e SSIM das imagens reconstruídas utilizando transformada <i>wavelet</i> com número limitado de <i>bits</i> para representar cada coeficiente inteiro e com uma casa decimal.....	25
Fig. 2.11. Desempenho dos esquemas de compressão revisados em PSNR para imagens: a) <i>Airplane</i> , b) <i>Lena</i> , c) <i>Peppers</i>	26
Fig. 2.12. Desempenho do algoritmo SPIHT utilizando um número limitado, (a) Utilizando somente a parte inteira dos coeficientes <i>wavelet</i> ; (b) Utilizando a parte inteira e uma casa decimal.....	27
Fig. 3.1. Planos de <i>bits</i> utilizados na TWD para cada nível.	30
Fig. 3.2. Esquema de codificação SPIHT utilizando o pós-processamento proposto onde diminui-se o número de <i>bits</i> requerido para cada plano de <i>bits</i> de <i>M</i> até <i>N bits</i>	31
Fig. 3.3. Vetor que estabelece o MSB de cada subbanda.	33
Fig. 3.4. Fluxograma da proposta da transformada <i>wavelet</i> com pós-processamento.....	34
Fig. 3.5. Passo de reordenamento utilizando o vetor <i>vMSB</i> do método de pós-processamento proposto para a TWD.	34
Fig. 3.6. Passo de utilização do último plano necessário (nível 1) para armazenar planos de <i>bits</i> dos outros níveis da TWD.	35
Fig. 3.7. Representação da memória física utilizada: <i>N</i> de <i>M bits</i> , depois do pós-processamento.	35
Fig. 3.8. Memória interpretada pela interface no quarto passo da proposta.	36
Fig. 3.9. Diagrama de blocos do algoritmo SPIHT enfatizando o acesso à memória.	37
Fig. 3.10. Teste para executar a TWD, o pós-processamento proposto e o SPIHT.....	38

Fig. 3.11. Desempenho da proposta de codificação SPIHT usando TWD com pós-processamento em termos de relação sinal ruído (PSNR).	39
Fig. 3.12. Análise de alguns pontos sobre o resultado do desempenho.	40
Fig. 4.1. Desempenho do JPEG-2000, SPIHT e JPG.	42
Fig. 4.2. Métodos de escalonamento dos planos de <i>bits</i> da ROI a) Não ROI; b) Método <i>ScaleShift</i> de escalonamento geral $s=6$; c) Método <i>Maxshift</i> $s=10$; d) Método híbrido <i>HBSHift</i> $s_1=3$ e $s_2=4$; e) Método <i>BbBSHift</i> $s_1=6$ e $s_2=4$; f) Método <i>GBbBSHift</i>	44
Fig. 4.3. Parâmetro ‘vetor de escalonamento’ da região de interesse: $vsROI$; do fundo: $vsBG$. ..	46
Fig. 4.4. Diagrama de codificação para compressão utilizando ROI a partir de uma TWD.	48
Fig. 4.5. Região de interesse vista na imagem original e na região da <i>LL</i> da TWD.	49
Fig. 4.6. Forma de aplicar a ROI desejada para uma ROI somente na <i>LL</i> (aproximada) e depois para outros níveis de decomposição.	50
Fig. 4.7. Procedimento teste para executar a compressão de imagens com ROI.	51
Fig. 4.8. Desempenho do algoritmo SPIHT personalizado na parte da TWD do conjunto de imagens teste (TWD bior4.4, somente parte inteira e com 15 <i>bits</i> para representar cada coeficiente).	52
Fig. 4.9. Máscara da ROI na <i>LL</i> m_ROI_LL de tamanho 8×8 <i>pixels</i> ; máscara gerada de toda a matriz de coeficientes m_ROI 512×512 <i>pixels</i>	52
Fig. 4.10. Resultados utilizando configuração híbrida que teve semente nos métodos anteriores à exceção do <i>scaling-based</i>	54
Fig. 4.11. Resultados utilizando configuração equivalente ao <i>scaling-based</i>	55
Fig. 4.12. Resultado desejável encontrado utilizando configuração híbrida que teve semente no <i>scaling-based</i>	56
Fig. 4.13. Resultados utilizando configuração híbrida que teve semente no <i>scaling-based</i>	57
Fig. 4.14. Resultados não desejáveis utilizando configuração híbrida que teve semente nos métodos mencionados à exceção do <i>scaling-based</i>	58
Fig. 5.1. Codificador turbo convolucional a) serie e b) paralelo.	62
Fig. 5.2. Construção de código turbo produto série em 2D e 3D.	64
Fig. 5.3. Curvas com número de iterações do código turbo convolucional com <i>frame</i> tamanho $N = 1024$ e taxas $R = 1/3$, $R = 1/2$ e $R = 2/3$	68
Fig. 5.4. Desempenho do turbo produto 2D, 3D, 4D e 5D para $R \approx 0,41$	69
Fig. 5.5. Curvas da taxa de distorção do LDPC. (a) $R = 1 / 3$; (b) $R = 1 / 2$, (c) $R = 2 / 3$	71
Fig. 5.6. Resultados do código convolucional + LDPC. (a) 5ª iteração, (b) 10ª iteração.	72
Fig. 5.7. LPDC com e sem BCH comparação a taxas diversas. (a) $R = 2/3$, (b) $R = 1/2$, (c) $R = 1/3$; (d) $R = 1/4$	73
Fig. 5.8. Curvas da taxa de distorção de código de várias taxas. (a) $R = 2/3$, (b) $R = 1/2$, (c) $R = 1/3$	74
Fig. 7.1. Fluxograma do algoritmo de criação de vetores binários $vROI$ e vBG	103
Fig. 7.2. Exemplo de como aplicar uma iteração do gerador de vetores binários no $vsROI$ e $vsBG$ e guardar os melhores resultados.	104
Fig. 7.3. Configuração do sistema ISDB-T, ressaltando as partes onde as pesquisas atuais podem ser aplicadas [159].	105
Fig. 7.4. Desempenho dos codificadores em PSNR e taxa de <i>bits</i> para uma sequência de vídeo em dimensão SD (720×480 <i>pixels</i> , 4:3) [162].	106

Lista de Tabelas

Tabela 3.1: Quantidade de <i>bits</i> necessária para representar cada coeficiente nos níveis da TWD.	32
Tabela 5.1: Resultado de simulação para código turbo convolucional serial.	67
Tabela 5.2: Resultado de simulação para código turbo produto 4D (3-7 dB).	69
Tabela 5.3: Complexidade computacional de códigos de correção de erros [151].	75

Abreviaturas

AVC	- <i>Advanced Video Coding</i>
ATSC	- <i>Advanced Television Systems Committee</i>
AWGN	- <i>Additive White Gaussian Noise</i>
BCH	- <i>Bose Chaudhuri Hocquenghem</i>
BCJR	- <i>Bahl-Cocke-Jelinek-Raviv</i>
BER	- <i>Bit Error Rate</i>
Bior4.4	- <i>Biorthogonal wavelets (filtro wavelet de ordem 4.4)</i>
BPE	- <i>Bit Plane Encoder</i>
BPP	- <i>Bit Per Pixel</i>
BTC	- <i>Block Turbo Code</i>
CDF	- <i>Cohen-Daubechies-Feauveau (filtro wavelet)</i>
CCSDS	- <i>Consultative Committee for Space Data Systems</i>
DVB-S	- <i>Digital Video Broadcasting for Satellite</i>
DVB-T	- <i>Digital Video Broadcasting Terrestrial</i>
EBCOT	- <i>Embedded Block Coding with Optimized Truncation</i>
EZW	- <i>Embedded Zerotree Wavelet coding</i>
GPL	- <i>General Public License</i>
HDTV	- <i>High Definition Television</i>
ISDB-T	- <i>Integrated Services Digital Broadcasting Terrestrial</i>
ISDB-T _B	- <i>Integrated Services Digital Broadcasting Terrestrial option B</i>
JPEG	- <i>Joint Photographic Experts Group</i>
JP2	- <i>Formato de imagem JPEG-2000</i>
JPIP	- <i>JPEG 2000 Interactive Protocol</i>
JPL	- <i>Jet Propulsion Laboratory</i>
JPX	- <i>Formato de imagem JPEG extended com metadados</i>

LCD	- <i>Liquid Crystal Display</i>
LDPC	- <i>Low Density Parity Check Code</i>
LLR	- <i>Log likelihood Ratio</i>
LIP	- <i>List of Insignificant Pixels</i>
LIS	- <i>List of Insignificant Sets</i>
LSP	- <i>List of Significant Pixels</i>
LSB	- <i>Least Significant Bit</i>
MAP	- <i>Maximum a Posteriori Probability</i>
MPEG	- <i>Moving Picture Experts Group</i>
MSB	- <i>Most Significant Bit</i>
PSNR	- <i>Peak Signal to Noise Ratio</i>
SDTV	- <i>Standard Definition Television</i>
SISO	- <i>Soft-Input Soft-Output</i>
SPC	- <i>Single Parity Check</i>
SPECK	- <i>Set Partitioned Embedded bloCK</i>
SPIHT	- <i>Set Partitioning in Hierarchical Trees</i>
SSIM	- <i>Structural Similarity</i>
TCC	- <i>Turbo Convolutional Code</i>
TPC	- <i>Turbo Product Code</i>
TWC	- <i>Transformada Wavelet Contínua</i>
TWD	- <i>Transformada Wavelet Discreta</i>
ROI	- <i>Region Of Interest</i>

Publicações

Listagem das publicações e participação em projetos e eventos acadêmicos durante o curso de doutorado na Faculdade de Engenharia Elétrica e de Computação da Unicamp (2006-2011).

1. LARICO, R. F.; IANO, Y.; HIGA, R.; SAOTOME, O.; A Coding Scheme for Storage Surveillance in Timeline using SPIHT and ROI Coding. International Workshop on Telecommunications (IWT 2011), Brazil, May, 2011.
2. LARICO, R. F.; IANO, Y.; Projeto SPACES4D: Sistema Participativo de Gestão e Monitoramento de Cidades e Serviços Públicos usando rastreamento com câmeras 4D. I Workshop Spaces4D, 2011.
3. IANO, Y.; SAOTOME, O.; LARICO, R. F.; CRUZ, A. L.; SILVA, F. S.; LEITE, R. B.; Turbo convolutional, turbo product and LDPC codes comparisons for low power applications. In: 13th International Symposium on Wireless Personal Multimedia Communications, 2010, Recife. Proceedings of the 13th International Symposium on Wireless Personal Multimedia Communications. Recife : Iecom Edition, v. 1, p. 30, 2010.
4. ARAUJO, A.F.; CRUZ, A. L.; SILVA, F. S.; LARICO, R. F.; SAOTOME, O.; Uma Análise de Desempenho de Códigos Concatenados Utilizando Códigos Convolucionais e LDPC. Revista Ciência e Tecnologia, v. 11, p. 19, 2009.
5. LARICO, R. F.; IANO, Y.; LEITE, R. B.; SILVA, F. S.; CRUZ, A. L.; SAOTOME, O.; Image Compression for Storage and Transmission of Satellite Digitized Images: source and channel coding. In: Workshop ITASAT 2009: III Workshop do Programa Satélite Universitário ITASAT, 2009, São Jose dos Campos. Caderno do ITA 2009, 2009.
6. LARICO, R. F.; IANO, Y.; SABLON, V.; Recognition Process of Human Iris: Fast Segmentation Approach. In: IV Iberoamerican Symposium in Computer Graphics, 2009, Margarita Island. Proceedings of SIACG 2009 IV Iberoamerican Symposium in Computer Graphics. Caracas, 2009. v. 1, p. 227.
7. LARICO, R. F.; IANO, Y.; Projeto STB-Scan: Ferramenta de coleta de sinais, estatística de acesso, auxílio à predição de cobertura do sinal digital terrestre e diagnóstico para instalações de TV digital como API para middleware Ginga, (I-VI) Workshop STB-Scan, 2009-2011.
8. ALVAREZ, V. H.; LARICO, R. F.; AZNAR, M.; IANO, Y.; Parameter Estimation for VLE Calculation by Global Minimization: Genetic Algorithm. Brazilian Journal of

Chemical Engineering, v. 1, p. 1-2, 2008.

9. CRUZ, A. L.; IANO, Y.; SILVA, F. S.; LARICO, R. F.; Compressão Fractal de Imagens. Telecomunicações (Santa Rita do Sapucaí), v. 11, p. 14-21, 2008.
10. OLIVEIRA, L. CRUZ, A. L.; SILVA, F. S.; IANO, Y.; LARICO, R. F.; SABLON, V.; Uma visão sobre Codificação *Wavelet* de Imagens. Revista Ciência e Tecnologia, v. 1, p. 9-28, 2008.
11. ARTHUR, R.; IANO, Y.; CARVALHO, S.; LARICO, R. F.; Planificación de la Expansión del Servicio de Retransmisión de TV Digital en Brasil usando redes SFN. Revista IEEE América Latina, v. 5, p. 573-578, 2007.
12. LARICO, R. F.; IANO, Y.; SILVA, F. S.; CRUZ, A. L.; SAOTOME, O.; Desenvolvimento de simuladores para codificação de um canal via satélite e aplicação dos códigos turbos e LDPC. In: Workshop ITASAT 2007: II Workshop do Programa Satélite Universitário ITASAT, 2007, São Jose dos Campos. Caderno do ITA 2007, 2007.
13. LARICO, R. F.; IANO, Y.; SABLON, Vicente. Processo de Reconhecimento de Íris Humana: Localização Rápida de Íris. In: 5th Latin American and Caribbean Consortium of Engineering Institution, Tampico, 2007.
14. LARICO, R. F.; CRUZ, A. L.; SILVA, F. S.; MILAGRES, D.; IANO, Y.; SAOTOME, O. . Compressão de dados sem perdas / com perdas e códigos corretores de erro. In: Workshop ITASAT 2006: I Workshop do Programa Satélite Universitário ITASAT, 2006, São Jose dos Campos. Caderno_WSH_ITASAT_2006, 2006.
15. ARTHUR, A. M.; IANO, Y.; LARICO, R. F.; TURUDA, P.; ARTHUR, R.; Técnicas de análise da retina humana: revisão breve sobre os recursos e as renovações propedêuticas disponíveis à prevenção e o diagnóstico precoce das complicações vasculares da retina associadas às doenças sistêmicas. Revista Ensaios e Ciência. Dez. 2011. (Aceito)
16. LARICO, R. F.; IANO, Y.; ARTHUR, R.; HIGA, R.; SAOTOME, O.; New approach for reduced memory wavelet transform using post-processing and SPIHT. Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT). 2012. (Submetido)
17. LARICO, R. F.; IANO, Y.; HIGA, R.; ARTHUR, R.; SAOTOME, O.; Generalized ROI applied to SPIHT. Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT). 2012. (Submetido)

Capítulo 1

Introdução

A compressão de imagens digitais é uma técnica que reduz as redundâncias. As imagens são representadas de outra forma, tendo-se como alvo a diminuição do espaço de armazenamento e a redução dos custos de envio por algum canal de comunicações.

O desenvolvimento de métodos inovadores em tratamento de imagens tem um custo relacionado (como por exemplo, para aumentar a compressão alguns esquemas aumentam a complexidade, outros têm baixa complexidade mas utilizam mais memória). O campo de sistemas embarcados atualmente, encontra-se em franca expansão e tem atraído a atenção de muitos pesquisadores. Além disso, cada vez mais se exige uma redução do tempo de desenvolvimento de novas versões de produtos [1]. Em geral, sistemas embarcados móveis, como por exemplo, celulares com câmera ou sistemas autônomos e satélites têm capacidades limitadas de processamento, comunicação e decisão [2]. Na construção dos equipamentos devem-se seguir diretrizes de acordo com a própria natureza dos mesmos [3]. Então se faz necessária a utilização de ferramentas e pesquisas que ajudem em cada estágio, desde a concepção da ideia até o desenvolvimento em *hardware*. Isso inclui a diminuição do custo relacionado à *aplicação-taxa*.

O presente trabalho de tese está em parte orientado para projetos que têm como objetivos, o desenvolvimento e lançamento de satélites tecnológicos de pequeno porte e promover a capacitação de recursos humanos nas áreas relacionadas à transmissão de imagens via satélite [4] (ver Anexo A). Isso porque as propostas e comparações sobre ferramentas utilizando a TWD (Transformada *Wavelet* Discreta) para redução de memória têm sido pesquisadas com o propósito de melhorar a utilização de recursos no processo de compressão de imagens [5, 6, 7, 8]. A codificação ROI (Região de interesse) é usada para priorizar a informação visual a ser enviada. Realizam-se também estudos e comparações sobre códigos corretores de erro que garantam um envio correto de dados para uma determinada taxa. Também o trabalho não é uma implementação em *hardware*, mas contém contribuições que podem ser aplicadas nesse campo de implementação física.

Aplicações da transformada *wavelet* discreta como ferramenta

As transformadas *wavelets* discretas têm sido fortemente adotadas como ferramentas em muitas áreas de concentração. Em especial, para a compressão de sinais, dada a sua habilidade em representar de forma eficiente transientes tais como sons de áudio ou componentes de alta frequência em imagens, atingindo altas taxas de compressão quando combinadas a esquemas sofisticados.

Os algoritmos rápidos devem ter boa capacidade de concentrar a energia do sinal em poucos coeficientes, transmissão progressiva (multiresolução), além de não introduzir artefatos de blocagem. Entre os algoritmos tem-se, por exemplo o EZW (*Embedded Zerotree Wavelet Coding*) [9] e o SPIHT (*Set Partitioning in Hierarchical Trees*) [10, 11].

A TWD hoje faz parte de um importante padrão de compressão de imagem, o JPEG2000 [12, 13]. Figuram entre aplicações do JPEG2000, por exemplo, dispositivos tais como câmeras digitais, assistentes digitais pessoais, telefones 3G móveis, fac-símile, impressoras, varredores ou escâner, etc. Em aplicações de detecção de região de interesse, de formas, de faces, íris, e outras, são também utilizados de forma satisfatória e com mais frequência [14, 15, 16]. Nessa transformada a região de interesse ROI pode ser incorporada e aplica-se separadamente uma codificação que permita dar graus de importância a essas regiões.

Estudos mais recentes também têm sido realizados para otimizar a TWD [17] e para realizar a compressão *wavelet* de imagens estáticas explorando características geométricas das imagens, obtendo melhores resultados em termos de qualidade visual com menor complexidade [18]. Diversos codificadores de vídeo tais como codificadores o Rududu que utiliza o esquema *Libdwic Directional Wavelet Image Compression* (<http://rududu.berlios.de/>), o codificador Dirac (<http://www.bbc.co.uk/rd/projects/dirac/>), Pixlet (*codec intraframe para Apple*), padrão ECW (*Enhanced Compression Wavelet*), MrSID (*Multiresolution Seamless Image Database*), entre outros, foram também desenvolvidos utilizando essa transformada.

Imagens codificadas com *wavelet* também apresentam vantagens em um ambiente de transmissão, ao serem reconstruídas no caso de perdas de pacotes na rede [19]. Em análise de dados científicos, a TWD encontra aplicações entre esquemas de mineração de dados (*data mining*) [20, 21], análise de bases de dados de meteorologia [22]. Em imagens médicas as *wavelets* são usadas, por exemplo, em especificações de intercâmbio médico dos dados, codificação e compressão de sinais ECG [23], detecção de câncer por meio de análise espectroscópica [24], classificação em mapeamento de genes [25], entre outros. É comum utilizar *wavelet* 3D em vídeo para aplicações médicas, onde a alta complexidade inerente é contornada com diversas técnicas, tais como, por exemplo, a exploração das características da tecnologia *Hyper-threading* de alguns processadores [26].

Muitos hospitais fazem uso de compressões de imagens com perdas com taxas de

compressão de 10:1 e até 40:1 operando sem perdas de informações com relevância clínica [27, 28]. Adicionalmente, *wavelets* são utilizadas em análise de sinais de voz para tratamentos de algumas doenças [29], classificação de tomografias [30], diagnósticos de problemas no coração [31] e em restauração de imagens de ultrassom [32]. Na área de inteligência artificial, é adotada em filtragem e compressão de sinais de um olfato artificial [33] e reconhecimento de objetos utilizando características locais e globais [34]. Em outras áreas de concentração as *wavelets* com esquemas de *lifting* são aplicadas em codificação de imagens de radar [35]. Em algoritmos para inserir a marca de água *watermarking* [36], em sinais musicais [37] e até em diversas aplicações como controle e proteção de linhas de transmissão (em conjunto a redes neurais) [38]. Diversas áreas além das referências anteriores utilizam as *wavelets*. Assim, pode-se dizer que são de grande interesse nas pesquisas atuais e futuras como ferramenta na solução de problemas.

Compressão de imagens utilizando TWD

A compressão de imagens utilizando *wavelets* resultou no padrão JPEG-2000 e variações, algoritmos rápidos como o EZW, SPIHT e variações. Neste trabalho, utilizam-se esses esquemas que usam a TWD por estar no estado de arte e por ser de fácil manipulação. De fato, a facilidade de se trabalhar com matriz que contem os coeficientes, serve para diversos propósitos, como por exemplo, para codificação ROI. Neste trabalho, aborda-se o desempenho dos algoritmos JPEG-2000, SPIHT, e a recomendação da CCSDS. No entanto, a ênfase será dada para o cálculo de necessidade de *bits* para o algoritmo SPITH por notar-se nele uma menor complexidade (ver Cap. 2).

O algoritmo SPIHT é quem tem o segundo melhor desempenho, somente superado pelo JPEG-2000 (diferença de $\sim 0,5\text{dB}$). No caso de uma implementação embarcada, o JPEG-2000 apresenta uma complexidade computacional superior em relação ao algoritmo SPIHT.

Redução de memória usando-se TWD na compressão de imagens

Em esquemas de compressão de imagens baseados em TWD, o módulo que gera esses coeficientes *wavelets* se junta sequencialmente a algum codificador de planos de *bits*. À medida que aumenta o nível da decomposição TWD (Transformada *Wavelet* Discreta), o número de *bits* necessários para representar os coeficientes *wavelets* também aumenta. Uma quantidade significativa de memória é necessária para armazenar esses coeficientes especialmente quando o nível de decomposição da TWD é elevado. Neste trabalho, um método de pós-processamento é proposto para definir a amplitude dos coeficientes variáveis. Isso é realizado, dependendo do nível do coeficiente, sendo que os planos de *bits* mais significativos dos últimos níveis podem ser aproveitados para armazenar outros planos de *bits* de outros níveis. Os resultados alcançados mostram uma significativa redução de consumo de memória para o processamento do algoritmo SPIHT utilizando as

características da decomposição *wavelet* e um pós-processamento (ver Cap. 3).

Codificação da região de interesse

Essa ideia é usada em muitas aplicações que utilizam codificação de imagens como navegadores de rede, imagens de base de dados, telemedicina e vídeo vigilância. É necessário reconstruir a imagem com prioridade em uma região de interesse ROI antes do restante da imagem. O presente trabalho fornece uma forma de fazer essa codificação em conjunto com o SPIHT, procurando-se utilizar diversos esquemas usados por outros codificadores. A proposta utiliza um esquema flexível para determinar a codificação ROI e também aceita múltiplas regiões. Além disso, essa proposta mantém as propriedades do SPIHT que apresenta baixa complexidade, é progressivo e tem bom desempenho com relação ao PSNR. Os resultados mostram que alguns vetores de configuração apresentam um melhor desempenho para um determinado conjunto de imagens de teste.

Codificação de canal para aplicações de baixa potência

Com uma codificação eficiente de canal, é possível se atingir uma redução significativa na potência de transmissão bem como aumentar a área de cobertura do sinal, mantendo uma perfeita recuperação de dados. Esses fatores são significativos nas transmissões seja via satélite ou via outros meios. Nesse âmbito, as codificações tipo turbo convolucional (TCC), produto (TPC) e o LDPC estão recebendo destaque no mundo das telecomunicações, dado o seu desempenho próximo ao limite teórico de Shannon. Para comparação, foram realizadas simulações, análises e tecidas conclusões quando se usam combinações entre esses códigos com um canal AWGN, variando-se parâmetros como tamanho de *frame*, número de iterações e taxas de código (puncionamento). Os resultados apresentados neste trabalho de tese permitem obter informações e conclusões referentes aos codificadores em questão. Os resultados mostram que cada codificador tem vantagens para cada taxa/aplicação e a complexidade deve ser levada em conta conforme a aplicação seja para dispositivos móveis ou satélite (ver Cap. 5).

O esquema de compressão de imagens utilizando codificação ROI pode ser utilizado em muitas aplicações específicas na área de imagens médicas, de satélite e de vídeo vigilância. Em específico na área de vídeo vigilância o esquema proposto foi testado e simulado para um esquema de armazenamento agendado, com vantagens no aumento de tempo de vida de imagens armazenadas [39]. Nesse contexto, apresenta-se um esquema geral para codificação de imagens a serem enviadas por um canal de comunicações. Logo, na recepção as imagens devem ser decodificadas e eventualmente armazenadas. A Fig. 1.1 mostra um esquema geral, onde as imagens são processadas com TWD, ROI e o algoritmo SPIHT. As imagens codificadas, após uma codificação de canal, são transmitidas através de um canal de comunicações. Tais aspectos são temas abordados na presente pesquisa.

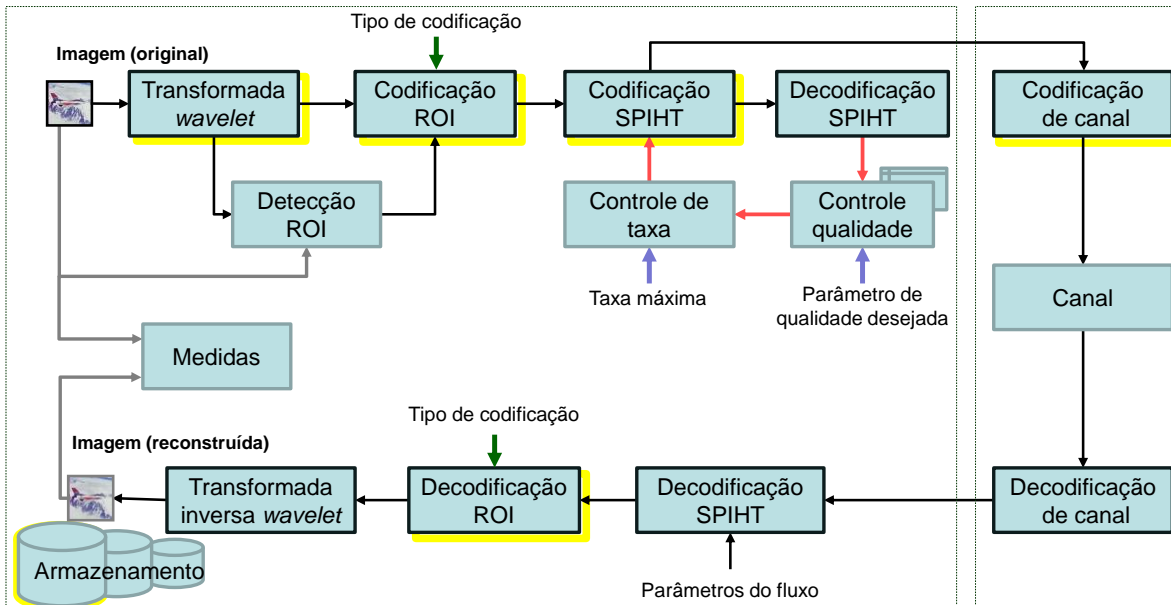


Fig. 1.1. Esquema de transmissão de imagens.

As comparações neste trabalho de tese utilizam o PSNR (*Peak Signal to Noise Ratio*) que é uma medida de relação entre a máxima energia possível de um sinal e o ruído que afeta a sua representação fidedigna. Essa medida é utilizada na comunidade científica em muitas pesquisas [40] e é uma referência utilizada em artigos antigos e recentes. Existem outras medidas de qualidade objetivas como o SSIM (em alguns caso é utilizado). Porém, para se fazer comparações, serão utilizadas as mesmas medidas que normalmente aparecem nos artigos antigos e recentes. Na referência [40] conclui-se que a PSNR ou SSIM são medidas muito relacionadas e que mostram graus de sensibilidade de degradação diferente. Em [16] e [14], conclui-se que o PSNR é um bom indicador de qualidade para aplicações em reconhecimento de face, digitais e íris quando se usam os esquemas de compressão de imagens estáticas no estado de arte, como o padrão JPEG2000 e o algoritmo SPIHT.

Objetivos

Os objetivos neste trabalho de tese estão voltados principalmente para melhorias de um sistema de transmissão de imagens com prévio processamento para garantir o adequado recebimento a um custo aceitável.

Dessa forma, são enfocados os seguintes itens:

- Melhor forma de armazenamento da TWD para o funcionamento em dispositivos com menor utilização de memória.
- Proposta de um esquema de codificação de ROI para se utilizar em um esquema de compressão usando o algoritmo SPIHT.
- Testes e esclarecimentos sobre o funcionamento de esquema de codificação

turbo. Trata-se de uma alternativa viável para aplicações em dispositivos para comunicação via satélite.

Contribuições deste trabalho de tese

Todas as contribuições deste trabalho de tese estão ressaltadas em amarelo na Fig. 1.1. e as propostas e comparações estão descritas em cada um dos seguintes capítulos:

- Capítulo 2: apresenta comparações gerais sobre compressão de imagens utilizando esquemas de compressão que utilizam a TWD como o JPEG-2000, o algoritmo SPIHT e o algoritmo recomendado pela CCSDS. Também foi implementado um módulo da TWD com a opção de controle de quantidade de *bits* que é utilizada.
- Capítulo 3: um módulo de pós-processamento da TWD para diminuir a memória da TWD na codificação SPIHT é proposto e implementado de forma flexível a fim de servir para outros propósitos além deste trabalho.
- Capítulo 4: um módulo de uma codificação de região de interesse generalizada aplicada ao SPIHT é proposto, e implementado com características de flexibilidade ao se codificar. Para testar várias formas de codificação foi implementado um algoritmo de geração de vetores de forma automática.
- Capítulo 5: comparações gerais entre codificadores de canal para aplicações de baixa potência são realizadas em ambiente de simulação. Isso serve para esclarecer que códigos de canal podem ser utilizados adequadamente.

Estrutura da tese

Todas as informações envolvidas no desenvolvimento desta tese estão descritas em cada um dos capítulos que compõem a presente pesquisa.

- Capítulo 2: definições e comparações gerais sobre compressão de imagens que são utilizadas nas pesquisas propostas.
- Capítulo 3: proposta de um esquema de pós-processamento da TWD para diminuir a memória da TWD na codificação SPIHT.
- Capítulo 4: proposta de uma codificação de região de interesse generalizada aplicada ao SPIHT, com características de flexibilidade ao se codificar.
- Capítulo 5: comparações entre codificadores de canal para aplicações de baixa potência para esclarecer que códigos de canal podem ser utilizados adequadamente.
- Capítulo 6: conclusões e trabalhos futuros.

Capítulo 2

Compressão de imagens utilizando transformada *wavelet*

2.1 Introdução

As técnicas de codificação de fonte são necessárias para reduzir a quantidade de dados que será transmitida. Os dados são submetidos a distorções normalmente destrutivas [41], por parte do canal de transmissão. Com a codificação de canal é possível atingir uma redução significativa na potência de transmissão, aumentar a área de cobertura do sinal, mantendo uma excelente recuperação dos dados. Porém, a codificação de canal possibilita proteger os dados utilizando redundâncias que podem até duplicar ou triplicar a quantidade de dados.

A codificação de fonte, para compressão de imagens, é um método utilizado para reduzir o volume de dados digitais, a fim de alcançar alguns benefícios, tais como:

- Redução da largura de banda do canal de transmissão;
- Redução do requisito de armazenamento temporário e memória;
- Redução de tempo de transmissão de dados em uma dada taxa.

Há duas classes de métodos de compressão de dados: sem perda (*lossless*) e com perda (*lossy*). Na compressão sem perdas, os dados originais podem ser reproduzidos exatamente, enquanto que na compressão com perdas, a quantização ou outras aproximações usadas no processo de compressão resultam na incapacidade para reproduzir o conjunto de dados originais sem alguma distorção. O aumento da quantidade de informação dos dados sujeitos a compressão sem perdas resulta em um grande volume de dados para um determinado conjunto de dados fonte.

Os esquemas de compressão (como por exemplo o JPEG-2000, SPIHT e EZW [12, 10, 9]) comumente, incluem duas partes: uma transformada das imagens (*wavelet*) e um codificador de planos de *bits*. A constante melhoria de sistemas tem levado a requisitos

básicos que incluem boa qualidade da imagem e transmissão progressiva, entre outros, que serão discutidos neste trabalho de tese.

O objetivo deste capítulo é apresentar os métodos e aplicações possíveis em sistemas de baixa complexidade. Essa baixa complexidade decorre da necessidade de baixo consumo de energia, como ocorre em satélites. Com isso, é possível, por exemplo, escolher um algoritmo mais adequado para um sistema de imagens para transmissão via satélite, por exemplo.

Nas seções 2.3, 2.4, 2.5 serão abordados os principais algoritmos e processos de compressão de imagens usados atualmente. Também serão mostrados algoritmos rápidos que servirão de base para as propostas deste trabalho, na seção 2.6.

2.2 Transformada *wavelet*

As bases *wavelet*, assim como as bases de Fourier, revelam a regularidade do sinal através da amplitude do coeficiente. Além disso, no caso da transformada *wavelet*, a sua estrutura conduz a um algoritmo rápido computacionalmente. No entanto, as *wavelets* são localizadas e poucos coeficientes são necessários para representar estruturas transitórias locais. Em oposição a uma base de Fourier, uma base *wavelet* define uma representação esparsa de funções com sinais regulares, que podem incluir transientes e singularidades. Em imagens, a maior parte dos coeficientes *wavelets* de alta amplitude estão localizados na vizinhança das bordas e nas texturas irregulares [42].

2.2.1 Transformada *wavelet* contínua TWC

A transformada *wavelet* é uma transformada pontual e proporcional à escala. A análise do sinal é feita em escalas e deslocamentos diferentes (dilatações e translações). Isso acontece a partir de uma função protótipo $\psi(t) \in L^2(\mathbf{R})$ (*wavelet* mãe) por meio de dilatações e translações como se observa na equação (2.1). O fator de escalonamento a e as translações por meio de uma variável b definem a forma geral de uma família *wavelet*.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbf{R}, a \neq 0 \quad (2.1)$$

onde $1/\sqrt{|a|}$ é o termo de normalização de energia.

Essas versões da *wavelet* básica $\psi(t)$ são capazes de gerar todo um espaço $L^2(\mathbf{R})$. Para que essas *wavelets* sejam aplicadas na prática em processamento de sinais devem ser satisfeitos os requisitos [43]:

- Suporte compacto: é dito ser de suporte compacto se a maioria da energia dessa *wavelet* estiver restrita a um intervalo finito, ou seja, se a função é exatamente zero

fora desse intervalo.

- Suavidade ou regularidade: a *wavelet* $\psi(t)$ tem p momentos nulos se $\int_{-\infty}^{+\infty} t^k \psi(t) dt = 0$, para $0 \leq k < p$. Assim, quanto maior o p mais suave será ψ , exceto para alguns casos onde é possível se obter *wavelets* com a mesma regularidade e com um número menor de momentos nulos [44].

Seja a *wavelet* $\psi(t)$ uma função oscilatória e de curta duração. Logo, a TWC é uma transformada real e é definida pela seguinte expressão na equação (2.2) [45].

$$W(a, b) = \langle f(t), \psi_{a,b}(t) \rangle = \int_{-\infty}^{+\infty} f(t) \cdot \psi_{a,b}(t) dt \quad (2.2)$$

Na equação (2.1), na definição sobre a *wavelet* $\psi_{a,b}$, observa-se que o parâmetro b é responsável pelo deslocamento de $\psi(t)$ ao longo do eixo t , e a é o responsável pela dilatação. A constante $1/\sqrt{a}$ é o termo de normalização da energia. Para $b = 0$ define-se o complexo conjugado da *wavelet* escalonada de a . Assim, tem-se:

$$\tilde{\psi}_a(t) = \psi_a^*(-t) = \frac{1}{\sqrt{a}} \psi_a^*\left(\frac{-t}{a}\right) \quad a \in \mathbf{R}, a \neq 0 \quad (2.3)$$

Pode-se reescrever como:

$$W(a, b) = \int_{-\infty}^{+\infty} f(t) \cdot \psi_a(t - b) dt = f * \tilde{\psi}_a \quad (2.4)$$

Nota-se que para a fixo, $W(a, b)$ é a convolução de $f(t)$ com o complexo conjugado refletido da *wavelet*-básica escalonada. A TWC pode ser interpretada como um banco de filtros lineares atuando sobre $f(t)$, conforme ilustrado na Fig. 2.1. Todas as saídas dos filtros juntas compõem a TWC.

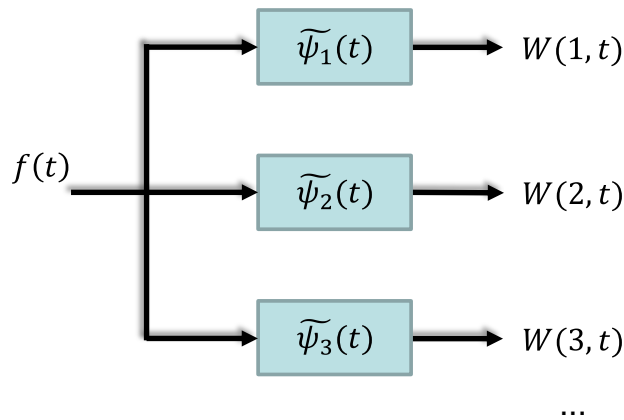


Fig. 2.1. Interpretação da TWC por banco de filtros.

Nota-se que $\widetilde{\psi}_1(t), \widetilde{\psi}_2(t), \widetilde{\psi}_3(t), \widetilde{\psi}_4(t)$ são, cada uma, uma *wavelet*-básica escalonada com uma frequência diferente e, portanto, cada uma é uma família de *wavelet* graças ao deslocamento b . Cada uma é ainda um conjunto de funções-base de mesma frequência [42, 46, 47].

2.2.2 Transformada *wavelet* discreta TWD

Para poder utilizar a transformada *wavelet* como ferramenta em computadores ou dispositivos dedicados, é necessário obter *wavelets* discretas. Para tanto, deve-se discretizar os parâmetros a (escala) e b (deslocamento), ou seja, as variáveis da função $W(a, b)$.

Para a toma-se valores inteiros de potências de um parâmetro fixo a_0 ,

$$a = a_0^{-j}, \quad a_0 > 1 \quad e \quad j \in \mathbf{Z} \quad (2.5)$$

Na equação (2.6) observa-se que a discretização de b deve depender de j tal que *wavelets* estreitas sejam deslocadas por passos pequenos, e *wavelet* largas sejam deslocadas por passos maiores.

$$b = kb_0 a_0^{-j}, \quad b_0 > 0, \quad j, k \in \mathbf{Z} \quad (2.6)$$

Pode-se escrever a *wavelet* discreta,

$$\psi_{j,k}(t) = a_0^{j/2} \psi(a_0^j t - kb_0) \quad j, k \in \mathbf{Z} \quad (2.7)$$

No caso discreto, devem-se cumprir condições necessárias sobre suporte compacto e regularidade. Existem várias formas de se implementar transformadas *wavelet* discretas como o algoritmo de Mallat, utilizando banco de filtros organizados em um esquema piramidal [42, 46].

As transformadas *wavelets* discretas TWD direta e inversa podem ser expressas da seguinte forma:

$$W(j, k) = d_{j,k} = \langle f(t), \psi_{j,k}(t) \rangle = \int f(t) \cdot \psi_{j,k}(t) dt \quad (2.8)$$

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \psi_{j,k}(t) d_{j,k} \quad (2.9)$$

onde $d_{j,k}$ são os coeficientes *wavelets*, e correspondem a $W(a, b)$ da transformada integral.

A reconstrução do sinal por meio dos coeficientes *wavelets* exige uma escolha muito especial de $\psi(t)$. Daubechies provou que a condição necessária e suficiente para a

reconstrução do sinal dos coeficientes *wavelet* é que a energia deve estar entre dois limites positivos [44]:

$$A\|f(t)\|^2 \leq \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} |\langle f(t), \psi_{j,k}(t) \rangle|^2 \leq B\|f(t)\|^2 \quad (2.10)$$

Se $A = B$, então a energia da TWD é proporcional à energia do sinal e as *wavelet* se comportam como bases ortogonais. Nesse caso específico a reconstrução é proporcional a $A + B$.

Uma maneira eficiente de aplicar essa transformada é por meio de filtros, técnica desenvolvida por Mallat [42, 48], onde se tem a decomposição da *wavelet* implementando a análise multiresolução. Essa é uma técnica que permite analisar sinais em múltiplas bandas de frequências [49, 50].

A transformada *wavelet* multiresolução pode ser considerada como um filtro passa-baixas e outro filtro passa-altas. A função *wavelet* $\psi(t)$ está relacionada a um filtro passa-altas, a qual produz os coeficientes de detalhes da decomposição *wavelet* (ver Fig. 2.2.c). Na análise multiresolução, há uma função adicional que está relacionada ao filtro passa-baixas, chamada de função de escalonamento $\phi(t)$ e está associada com os coeficientes de aproximação da decomposição da *wavelet*.

Para aplicar a TWD em 2D, a imagem terá que passar do domínio espacial para o domínio da *wavelet* (isso pode ser em vários níveis). Quando é aplicada a TWD bidimensional obtém-se como resultado a matriz de coeficientes em quatro partes ou grupos de coeficientes (ver Fig. 2.2 e Fig. 2.3). *L* significa frequência baixa (*Low*), *H* significa frequência alta (*High*).



Fig. 2.2. Quadrantes de frequência da TWD aplicada a uma imagem; a) os quatro quadrantes *LL*, *HL*, *LH* e *HH*; b) imagem original; c) TWD da imagem em um nível.

Na Fig. 2.2 observa-se o quadrante *LL* chamado de passa-baixas. Aqui se inicia uma iteração e o quadrante *LL* se divide em outros quatro quadrantes. Usualmente, para análises ou compressão de imagens é usual realizar três, quatro ou cinco iterações (nível da TWD).

Note-se na Fig. 2.3, o comportamento da intensidade diferenciada dos coeficientes da TWD (utilizando filtro CDF 9/7). Ao aumentar o número de níveis observa-se que a intensidade é concentrada no passa-baixas *LL*, em cada nível.

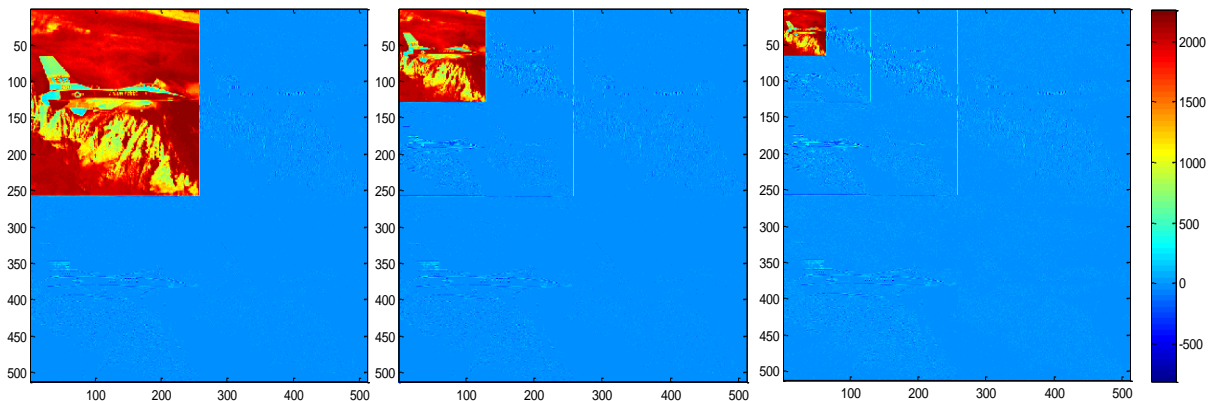


Fig. 2.3. Intensidades dos coeficientes da TWD com 1, 2 e 3 níveis.

A TWD tem propriedades que são exploradas estatisticamente pelos codificadores. Tais propriedades são:

- Localização espaço-frequência. Cada coeficiente *wavelet* representa informações de certa banda de frequências em uma determinada localização espacial.
- Compactação de energia. A decomposição piramidal da construção da TWD resulta que as passa baixas (*LL*) compactam efetivamente a energia em poucos coeficientes. Isso se deve ao comportamento da energia contida em regiões homogêneas (típico em imagens naturais), tender a concentrar-se em uma dessas subbandas (*LL*).
- Similaridade entre subbandas em diferentes escalas. Essa similaridade entre os coeficientes insignificantes (zeros) pode ser representada em uma árvore de *zerotree*.
- Decaimento da magnitude dos coeficientes *wavelet* entre as subbandas. Estatisticamente pode-se comprovar que a magnitude decai de pai para o filho.
- Agrupamento de coeficientes significativos em uma subbanda (clusterização).

Até agora, foram brevemente revisados alguns conceitos sobre a TWD que servirão de apoio para o entendimento das seções e capítulos seguintes. A importância da TWD como ferramenta utilizável em diversas aplicações na presente área de concentração bem como em outras áreas é revisada na seção 1. Os conceitos teóricos e detalhes matemáticos podem ser encontrados em [42, 46, 47, 48].

2.3 Algoritmos de compressão de imagens

Nesta seção será estudada a parte geral de algoritmos de compressão de imagens que representa o estado da arte e também esquemas caracterizados por serem rápidos e de desenvolvimento (implementação) simples. A compressão de imagens, utilizando a TWD, tem sido eficiente devido às estratégias inovadoras para a organização e representação de dados da imagem transformada. Tais estratégias, de forma implícita ou explícita, exploram propriedades estáticas dos coeficientes transformados em uma pirâmide *wavelet* [47].

O padrão atual de maior destaque é o JPEG-2000 [12, 13], referenciado como um algoritmo de estado da arte. Também é analisada a recomendação CCSDS [51] que utiliza a transformada *wavelet* e um codificador de planos de *bits* que será explicado brevemente (ver seção 2.4). Também é analisado o algoritmo rápido de compressão de imagens SPIHT (*Set Partitioning In Hierarchical Trees*) e mencionam-se algumas de suas variantes (ver seção 2.6). O restante da tese é todo baseado em um módulo de transformação *wavelet*.

2.4 CCSDS: Recomendação para compressão de imagens

A finalidade dessa recomendação (*Image data compression* CCSDS 120.1-G-1) é para normalização dos algoritmos de compressão de dados aplicados a imagens espaciais digitais bidimensionais, obtidos de instrumentos da carga útil (*payload*). Serve para especificar como esses dados comprimidos são formatados em segmentos para permitir a descompressão no receptor final. Essa recomendação para normalização define um algoritmo específico para compressão de dados de imagem de carga útil que tem aplicabilidade difundida para vários tipos de instrumentos. A teoria e os detalhes sobre a operação são descritos em [51].

A técnica de compressão descrita pode ser usada para produzir ambas as formas de compressão: sem perda (*lossless*) e com perda (*lossy*). Esse algoritmo é baseado no padrão JPEG-2000 [12]. Entretanto, a recomendação CCSDS difere do padrão JPEG-2000 em vários aspectos:

- A técnica foi criada especificamente para instrumentos de taxa alta de transmissão, ou seja, instrumentos que são usados a bordo de espaçonaves;
- Um compromisso é mantido entre o desempenho de compressão e a complexidade com ênfase específica em aplicações de espaçonaves;
- A baixa complexidade desse algoritmo auxilia na implementação rápida e de baixa potência (*low-power*) do *hardware*;
- A técnica tem um conjunto limitado de opções, auxiliando na sua aplicação bem sucedida sem um conhecimento profundo do algoritmo.

Os detalhes importantes são descritos brevemente, a seguir. A recomendação da CCSDS está baseada em dois blocos funcionais: a transformada *wavelet* discreta (TWD) e a codificação de planos de *bits* (BPE - *Bit Plane Encoder*), como se observa na Fig. 2.4.

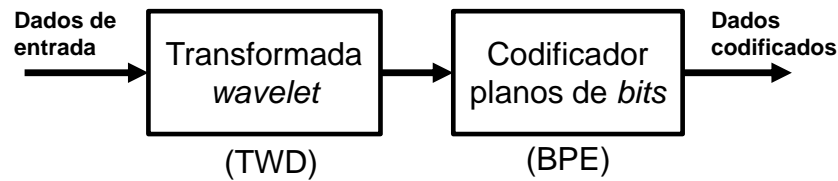


Fig. 2.4. Dois blocos básicos nos algoritmos de compressão recomendado pela CCSDS.

Existem especificações detalhadas para o fluxo (*bitstream*), formato, e outros itens. Neste trabalho somente são abordados o tema da codificação da imagem tanto o bloco TWD quanto o codificador de planos de *bits* e não os protocolos ou formas de envio de dados.

2.4.1 Transformada *wavelet* discreta no CCSDS

Essa recomendação para normalização, para o módulo de decorrelação, faz uso da transformada *wavelet* discreta (TWD) separável, bidimensional (2D), de três níveis, com nove e sete coeficientes (*taps*) para filtros passa-baixas e passa-altas, respectivamente. Essa decorrelação da imagem é efetuada usando uma TWD bidimensional, a qual é executada pela aplicação repetida da TWD unidimensional (ver Fig. 2.5).

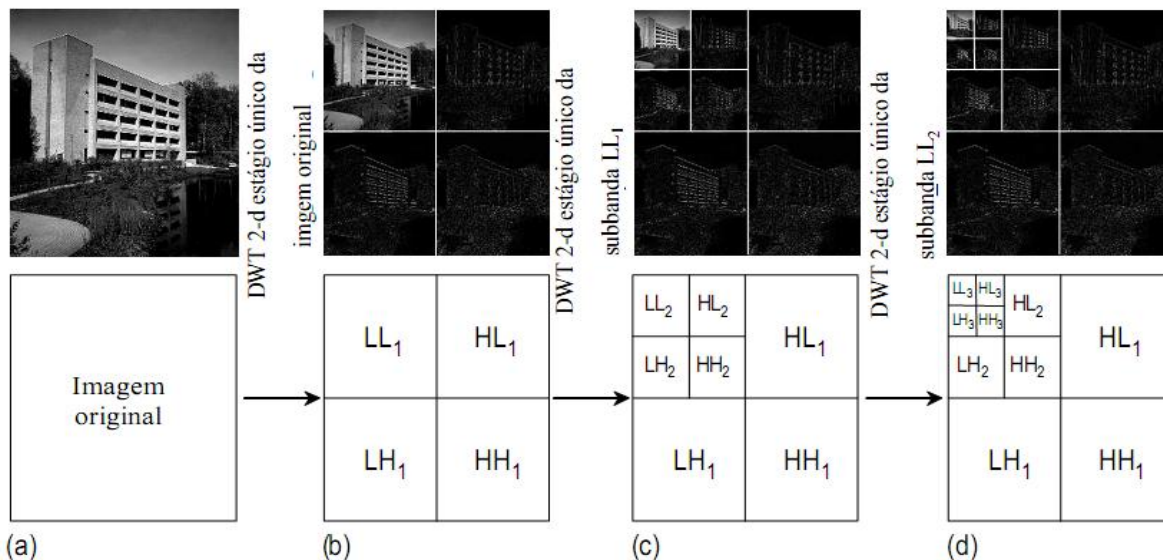


Fig. 2.5. Decomposição TWD 2D em três níveis de uma imagem [51].

Considerando-se a imagem como uma matriz de dados composta por linhas e colunas de vetores de sinal, uma TWD 2D de um nível deve ser implementada na imagem usando-se dois passos: 1) A TWD 1D deve ser calculada em cada linha da imagem, produzindo uma matriz (*array*) intermediária de dados filtrados; 2) A TWD 1D deve ser aplicada a cada coluna de ambas as matrizes intermediárias para produzir quatro subbandas, isso para um nível de decomposição (ver Fig. 2.5.b).

Para aumentar a eficiência da compressão, a correlação restante na subbanda *LL*, depois da decomposição TWD 2D, é explorada aplicando-se níveis adicionais de decomposição TWD para produzir uma TWD 2D multinível. Isso produz a decomposição piramidal descrita em [42, 48]. Essa recomendação CCSDS especifica três níveis de decomposição como se observa na Fig. 2.5.d.

2.4.2 Codificador de planos de *bits* (BPE)

Após a TWD, os coeficientes da *wavelet* têm duas opções:

- São aproximados para o inteiro mais próximo (quando a transformada de ponto flutuante é usada);
- Escalonados, utilizando os fatores de peso específicos (quando a transformada de inteiros é usada).

Quando o escalonamento é realizado, um ou mais dos *bits* menos significativos, em muitas subbandas, são necessariamente zerados. Na descrição nesta seção, os valores de coeficiente mencionados referem-se aos valores após qualquer operação de escalonamento ou de aproximação. A operação *bitshift* é usada para indicar o número de *bits* menos significativos em cada coeficiente da subbanda específica que são necessariamente zero como resultado da operação de escalonamento da subbanda. Quando a TWD 9/7 de ponto flutuante é usada, as subbandas não são escalonadas, e assim o *bitshift* é ajustado em zero para cada subbanda. O codificador de planos de *bits* (BPE) processa coeficientes *wavelet* (Fig. 2.6.a) em grupos de 64 coeficientes chamados de blocos. Como se observa na Fig. 2.6.b, um bloco corresponde a uma região localizada na imagem original.

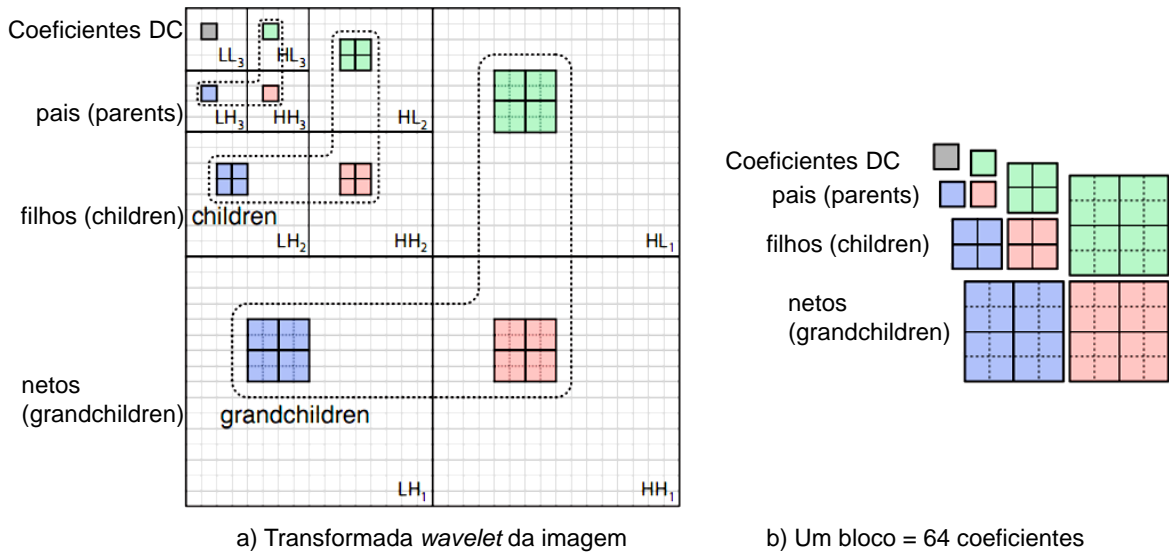


Fig. 2.6. Esquema da TWD de uma imagem para o uso do codificador de planos de *bits* BPE [51].

Essa estrutura é utilizada pelo algoritmo para tirar informação dos grupos de coeficientes dentro do grupo, dada a alta correlação estatística. O bloco é segmentado e se passa o algoritmo em uma ordem de leitura. O algoritmo faz uma seleção de *bits* de coeficientes significativos por meio do algoritmo Rice [52].

2.4.3 Algoritmo Rice

Esse algoritmo foi proposto por Rice [52] em 1991, como um candidato com excelente desempenho e baixa complexidade para implementação. A recomendação CCSDS está baseada nele e são adicionadas opções de entropia adaptativa. Esse algoritmo explora conjuntos de códigos de tamanho variável para poder comprimir os dados. Cada código está perto do ótimo para uma dada distribuição. Por isso, diferentes códigos são utilizados no algoritmo e devem ser adaptativos para muitas fontes de dados (passar de baixa entropia a alta). O algoritmo trabalha com blocos ou pacotes independentes.

Segundo uma análise matemática feita em [53], cada opção do algoritmo é essencialmente um código de *Huffman* com distribuição laplaciana (PDF) para algum determinado conjunto de valores de entropia [54].

Esse algoritmo funciona com alto desempenho somente se a origem de dados for adequada (devido à extensão limitada de entropia onde se comporta com melhor desempenho). Esse fato acontece na maioria dos codificadores entrópicos. No caso de imagens espaciais, os esquemas são previsíveis [53], e o algoritmo Rice pode ser bem aplicado. Também pode ser aplicado em outras fontes previsíveis de imagens.

2.4.4 TER - Recomendação para compressão de imagens CCSDS

TER é uma implementação da recomendação CCSDS para compressão de imagens (*Recommended Standard 122,0 CCSDS-B-1 Blue Book*). TER é projetado e programado com o objetivo de proporcionar uma boa base para se testar e desenvolver a recomendação CCSDS para compressão de imagens [55].

O aplicativo fornece diferentes funcionalidades que podem ser controladas através de seus parâmetros. Esses parâmetros podem ser passados como argumentos do programa, selecionando alguns dos algoritmos que o aplicativo incorpora. Mais de 30 parâmetros permitem a combinação de diferentes processos para produzir uma imagem com *codestream* comprimido [56]. Assim como em JPEG-2000, o fluxo codificado pode ser ordenado por:

- Qualidade (L);
- Posição (P);
- Resolução (R);
- Componente (C).

Obtêm-se, entre outros, os seguintes pedidos de progressão: LRCP, RLCP, RPCL, PCRL, e CPRL. Cabeçalhos embarcados (*embedded*) permitem decodificar qualquer ordem de progressão. Todas essas informações e os códigos fonte estão publicamente disponíveis no site *web* "<http://www.gici.uab.es/TER>" [55, 56], onde é possível adquirir as fontes e os manuais de forma livre, aceitando uma licença do tipo GPL (*General Public License*).

O aplicativo em *software* TER também incorpora algoritmos para torná-lo flexível o suficiente para gerenciar diferentes tipos de dados, para controlar e monitorar os estágios de compressão, para extrair informações estatísticas, etc. Assim, isso faz com que o TER seja uma plataforma para testar novas ideias e não para ser um produto comercial ou para ser integrado a outros aplicativos que precisam de suporte para arquivos de codificação/decodificação com a recomendação CCSDS para imagens. Basicamente é um *software* de uso acadêmico e multiplataforma modular, implementado usando orientação a objetos. Essas funcionalidades do aplicativo implicam num desempenho em velocidade não tão boa, e dessa forma, não sendo recomendável para uso comercial.

Neste trabalho será testada essa implementação somente para esclarecer o desempenho entre as implementações de referência do JPEG-2000 e do SPIHT.

2.5 JPEG-2000

O padrão de compressão JPEG-2000 é um padrão de compressão de imagens de alta definição baseado em bases *wavelet* [42, 13]. Pode compactar até 90% do arquivo original sem perder a qualidade de imagem. Em comparação ao JPEG que possuía uma distorção moderada utilizando uma taxa de 1,0 bpp, o JPEG-2000 degrada pouco uma imagem nessa taxa e, graças à característica de transmissão progressiva pode ser utilizado em baixas taxas. JPEG-2000 é um codificador que aplica o algoritmo EBCOT (*Embedded Block Coding with Optimized Truncation*) de Taubman e Marcellin [13], e com sucessivas melhorias, otimizações e pesquisas resultou em um elegante código, boa qualidade de visualização, em comparação ao padrão JPEG [42]. É considerado estado de arte, com bom desempenho em baixas taxas [57].

Características básicas

- Possui desempenho de compressão superior. A altas taxas de *bits*, quando artefatos se tornam quase imperceptíveis, o JPEG-2000 tem uma vantagem de fidelidade sobre o JPEG. Em baixas taxas de *bits*, o JPEG-2000 tem muito mais vantagens significativas sobre certos modos de JPEG, onde a ausência de efeito de blocos é o mais aparente. Os ganhos de compressão sobre JPEG são atribuídos ao uso da TWD (*Discrete Wavelet Transform*) e a um método mais sofisticado de codificação entrópica.
- Transmissão progressiva com precisão em *pixel*, resolução (comumente referenciada como decodificação progressiva) e taxa de escalabilidade sinal-ruído (SNR). A qualidade melhora progressivamente à medida que mais *bits* da fonte são lidos.
- Permite compressão com e sem perdas. O padrão JPEG-2000 provê compressão com e sem perdas em uma única arquitetura de compressão. A compressão sem perda é feita com o uso de uma transformada inteira de *wavelet* reversível no JPEG-2000.
- Acesso aleatório ao código *stream*, também referido como região de interesse (ROI). Os *streams* de código JPEG-2000 oferecem alguns mecanismos para suportar acesso espacial aleatório ou acesso à região de interesse a vários graus de granularidade. Dessa forma, é possível armazenar diferentes partes da mesma imagem usando-se diferentes qualidades.
- Possui resistência a erros. Como o JPEG (1991), o JPEG-2000 é tolerante a erros de *bits* introduzidos por canais de comunicações com ruído, devido à codificação de dados em pequenos blocos independentes.
- Tem formato de arquivo flexível. Os formatos JP2 e JPX permitem manusear a informação de cor-espaco, *metadata*, e interatividade em aplicações em rede como foi desenvolvido no protocolo JPEG Part 9 JPIP.

- Permite informação espacial de *side channel*. Suporta totalmente transparência e planos *alpha*.
- Permite representação múltipla de resolução. O JPEG-2000 decompõe a imagem em uma representação múltipla de resolução no processo de compressão. Essa representação pode ser usada para outros propósitos de apresentação de imagem.

2.5.1 Processo de codificação

O processo de codificação é feito em um codificador embarcado que tem a vantagem de explorar *inter* e *intra* dependências dos coeficientes *wavelet*. As operações básicas do algoritmo são:

- Codificação de significância;
- Codificação de sinal; e
- O refinamento.

Essas operações básicas são feitas com codificação aritmética adaptativa. Esse tipo de codificador explora a intensidade do coeficiente *wavelet* para criar um número de intensidade absoluta menor, dependendo da vizinhança do mesmo coeficiente dentro de um contexto probabilístico.

Como se observa na Fig. 2.7 os coeficientes *wavelet* são divididos em blocos independentes de tal forma que cada bloco S_k (de $L \times L$ coeficientes) é codificado segundo uma codificação aritmética em cada plano de *bits*. Isso gera o *stream* binário c^k . O fluxo de dados ou *stream* é composto por *substreams* $c^k = \{c_1^k, c_2^k, c_3^k, \dots\}$ onde cada c_n^k corresponde ao n -ésimo plano de *bits*.

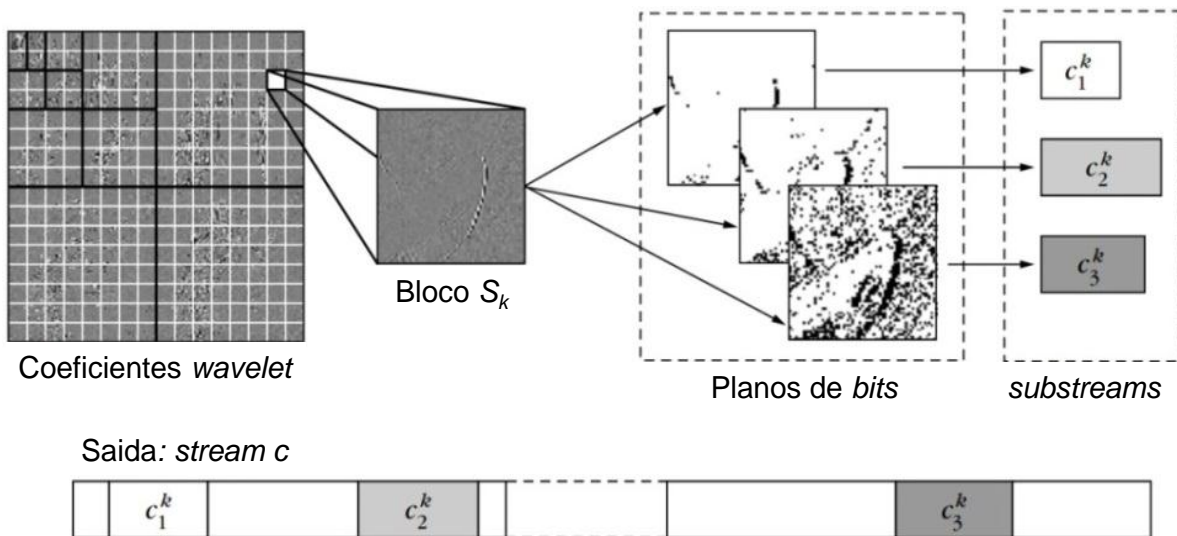


Fig. 2.7. Visão geral do processo de compressão JPEG-2000 [42].

O algoritmo de codificação tem uma alta complexidade, pois contém vários módulos sequenciais. Existem vários *softwares* que implementam o JPEG-2000. Como exemplo apresenta-se na Fig. 2.9.a, o digrama de fluxo utilizado na *Intel*® [58] que corresponde ao algoritmo somente de codificação da imagem. Na Fig. 2.9.b, apresenta-se o algoritmo geral utilizado que se encontra em [59] sobre multiprocessamento em *chip*.

O fator limitante para o JPEG-2000 é a memória que o formato demanda [60]. Isso representa uma desvantagem em comparação aos métodos rápidos (EZW, SPIHT, SPECK).

Atualmente, estão disponíveis *software* e *hardware* comerciais como um produto da *Analog Device* denominado ADV202 (*JPEG-2000 Video codec*) para aplicações de alta qualidade e resolução de imagem. Também tem-se, *IP-Cores* como da *intoPix* para inserir em diversas placas de desenvolvimento como a *Virtex (intoPIX America, Inc.)*.

Além das implementações privadas e específicas que existem no mundo comercial que cumprem essas especificações, existem versões livres (de acordo com a licença) para uso acadêmico, como: FFmpeg, JasPer [61], JJ2000¹, OpenJPEG². Versões proprietárias como o Kakadu (sendo uma opção recomendada pela rapidez de codificação) e a versão do Matlab. Segundo a comparação feita em [62], o JasPer é um pouco melhor do que o Kakadu, dado que tem uma leve vantagem em termos da métrica PSNR.

Os detalhes de cada parte ou de cada módulo poderão ser estudados na norma e outras publicações referentes ao *international JPEG-2000 part I final committee draft version 1.0* [12, 63, 61] como observado na Fig. 2.8. Cada anexo dessa especificação também está disponível nesse documento [63].

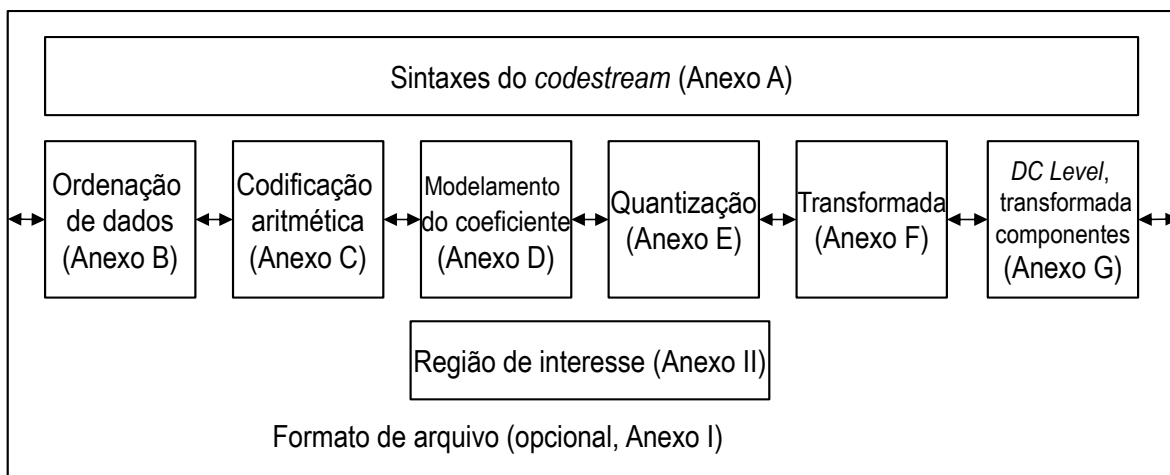
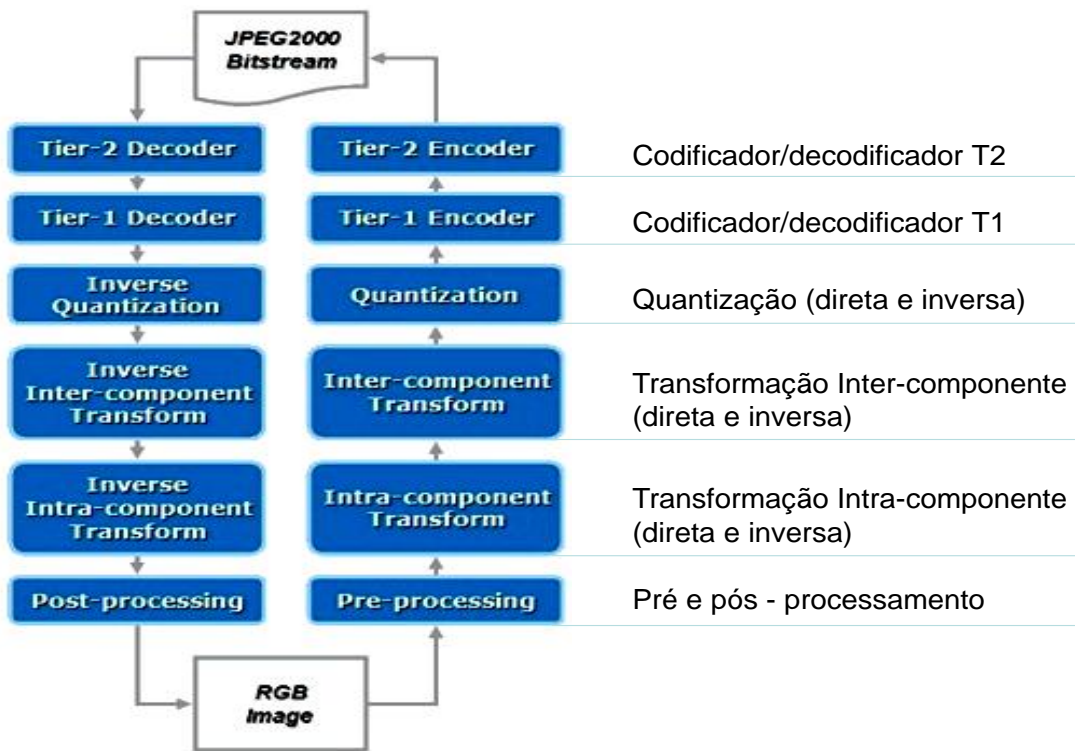


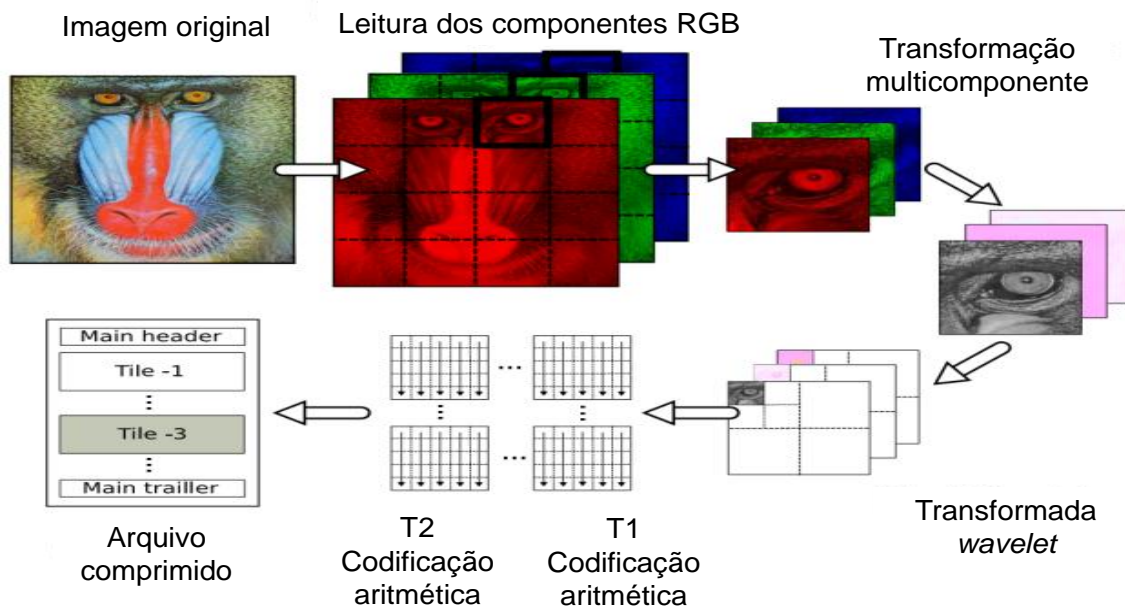
Fig. 2.8. Diagrama de blocos e de documentos próprios da especificação JPEG-2000 [63].

1 JJ2000, <http://code.google.com/p/jj2000/>

2 OpenJPEG homepage, <http://www.openjpeg.org/>



(a)



(b)

Fig. 2.9. a) Diagrama de fluxo do JPEG-2000 (*Intel IPP - Codec Process Flow* [58]); b) Um algoritmo do JPEG-2000 visualizado em [59].

2.6 SPIHT (*Set Partitioning In Hierarchical Trees*)

O algoritmo SPIHT foi introduzido por Said e Pearlman [11, 64] e se apresenta como uma versão bastante refinada do EZW (*Embedded Zerotree Wavelet coding*) [9, 65, 66]. Esse algoritmo apresenta resultados com PSNR elevados e taxas de *bits* aceitáveis (em relação ao EZW, por exemplo). Consequentemente, Rao e Yip [67] o apontam como sendo provavelmente o algoritmo de compressão de coeficientes *wavelet* mais utilizado com variações diversas, servindo de padrão básico de comparação para algoritmos subsequentes. Tanto o EZW e o SPECK quanto o SPIHT utilizam o conceito de árvores de particionamento. Como a decomposição concentra a energia nas subimagens de menor escala, em geral, os coeficientes possuem magnitudes maiores à medida que estão localizados mais próximos da raiz da árvore. Logo, se um dado coeficiente tiver magnitude menor que um dado limiar (*threshold*), sua descendência tende fortemente a apresentar magnitudes menores que esse mesmo limiar (propriedade de compactação da *wavelet*). Nesse contexto, o EZW, SPECK e o SPIHT (e variações) executam passo por passo o refinamento. Isso possibilita uma alta compressão e a transmissão progressiva dos dados.

No EZW, uma grande parcela de informação é transmitida a baixo custo quando uma subárvore inteira é declarada como não-significativa sendo representada com uma simples ‘etiqueta’ *zerotree*. O SPIHT usa um particionamento melhorado das árvores de forma a manter os coeficientes *wavelet* insignificantes, melhor agrupados em subconjuntos maiores [68].

As decisões binárias de particionamento são transmitidas ao decodificador. A eficiência da codificação do mapa de significância pelo SPIHT é tal que a codificação aritmética das decisões binárias gera muito pouco ganho adicional na codificação [67]. Já o SPECK utiliza pequenos blocos ao invés de coeficientes e trabalha da mesma forma que o SPIHT. Os limiares usados para testar a significância são potências de 2, de forma que em sua essência, o SPIHT termina por enviar a representação binária do valor inteiro dos coeficientes *wavelet*.

Outras implementações do algoritmo incluem melhorias e variações. Utiliza-se para melhorar o desempenho desse algoritmo SPIHT métodos híbridos [69, 70, 71, 72] que melhoram o desempenho ao comprimir imagens. Outros autores gerenciam menor quantidade de listas [73] com o propósito de utilizar a menor quantidade de recursos nos codificadores. Outros propõem redução de memória dos codificadores ou módulos em geral [8] e melhorias quanto à utilização de memória em sistemas embarcados [74, 75, 6].

2.6.1 Algoritmo SPIHT

Baseado na descrição do próprio autor [76], tem-se:

```
(1) Iniciar - Initialization:
    • iniciar  $n = \text{floor}(\log_2(\max(\text{abs}(c_{i,j})))$ ;
    • iniciar lista LSP vazia.
    • iniciar a lista LIP com as coordenadas  $(i,j)$  que pertencem H
    • iniciar a LIS igual que a LIP a exceção de quem não tem descendência, com
      etiqueta A
(2) Passo de significância - Sorting pass:
    (a) Para cada entrada  $(i,j)$  na LIP:
        (i) enviar saída correspondente  $G_n(i,j)$ ;
        (ii) cumprir a condição:
            Se  $G_n(i,j) = 1$  então
                mover  $(i,j)$  para a LSP e a saída é o sinal de  $c_{i,j}$ ;
    (b) Para cada entrada  $(i,j)$  na LIS fazer:
        (i) Se a entrada é de tipo A então
            • enviar saída correspondente  $G_n(D(i,j))$ ;
            • Se  $G_n(D(i,j)) = 1$  então
                — Para cada  $(k,l)$  pertence  $O(i,j)$  fazer:
                    * enviar saída correspondente  $G_n(k,l)$ ;
                    * Se  $G_n(D(k,l)) = 1$  então
                        adicionar  $(k,l)$  para a LSP e a saída é o sinal  $c_{k,l}$ ;
                    * Se  $G_n(D(k,l)) = 0$  então
                        adicionar  $(k,l)$  para o final da LIP;
                — Se  $L(i,j)$  é diferente de 0 então
                    mover  $(i,j)$  para o final da LIS, com etiqueta tipo B,
                    em outro caso remover a entrada  $(i,j)$  da LIS;
        (ii) Se a entrada é de tipo B então
            • enviar saída correspondente  $G_n(L(i,j))$ 
            • Se  $G_n(L(i,j)) = 1$  então
                — adicionar cada  $(k,l)$  que pertence  $O(i,j)$  para o final da
                  LIS com etiqueta tipo A;
                — remove  $(i,j)$  da LIS.
(3) Passo de refinamento - Refinement pass:
    Para cada entrada  $(i,j)$  na LSP, (a exceção os incluídos no ultimo 'sorting pass')
        * enviar saída correspondente ao  $n$ -ésimo bit significativo de  $\text{abs}(c_{i,j})$ ;
(4) Passo de atualização - Threshold update:
    Reduzir  $n$  em 1 e iniciar desde o passo (2). Para  $n \geq 0$ 
Onde:
H são as coordenadas das raízes das arvores (spatial orientation tree - SOT).
 $G_n(i,j) = \{ \begin{array}{l} 1, \max(\text{abs}(c_{i,j})) \text{ para } i,j \text{ pertence } T, \\ 0, \text{ outros casos} \end{array} \}$ 
```

Para entender melhor o algoritmo SPIHT encontram-se disponíveis exemplos no próprio livro do autor [76] e também um exemplo numérico passo a passo completo em [77].

O algoritmo SPIHT está em constante pesquisa e melhorias ainda têm sido propostas, mas em termos de simplicidade, para taxas baixas e codificação rápida, já é considerado um dos melhores, havendo outros algoritmos baseados nele com poucos ganhos em desempenho e qualidade. Esse é o motivo pelo qual é pesquisado como uma alternativa viável em imagens de satélite dado o pouco consumo de energia que deveria apresentar em relação à complexidade. Atualmente, existem implementações do algoritmo do autor [10], também implementações em linguagem C de Fowler [78]. A versão que é utilizada para teste neste trabalho está feita em Matlab (MathWorks®) e atende às especificações do algoritmo original básico.

SPECK (*Set Partitioned Embedded bloCK*)

Algoritmos, os quais utilizam quantização escalar em estruturas hierárquicas em codificação de imagens, são usados com bastante eficácia e com técnicas computacionais simples. Esses algoritmos são versões da proposta de Shapiro [9], melhorado muitas vezes, sendo uma ‘versão base’ para o início do desenvolvimento do SPIHT [76, 79]. Todos os esquemas de hierarquização e quantização utilizam a ideia de explorar os valores mais significativos dos *pixels* (coeficientes *wavelet*) de acordo com algum limiar conhecido. Características importantes em um codificador de imagens rápido e com eficácia são a propriedade de transmissão progressiva e a possibilidade de ser embarcado (além de ter um bom desempenho na compressão). O algoritmo SPECK [79] possui essas características desejáveis como as anteriormente observadas, mas com algumas particularidades. Uma característica é que ao invés de explorar as similaridades entre subbandas em árvore, o SPECK utiliza conjuntos de blocos dentro de uma subbanda.

As características do codificador são: 1) é completamente embarcável; 2) é de transmissão progressiva; 3) tem baixa complexidade computacional; 4) tem baixo consumo de memória dinâmica; 5) apresenta codificação e decodificação rápidas; 6) possui desempenho eficiente, e; 7) é utilizado para compressão com e sem perdas. O desempenho é parecido ao do SPIHT e de outras variantes que utilizam a TWD como mostram os resultados em [79]. O algoritmo detalhado pode ser encontrado em [80].

2.7 Simulações e resultados

O objetivo deste capítulo é fornecer uma base de simulações que auxiliem na determinação de qual algoritmo pode ser utilizado na codificação de fonte em satélite ou para outras pesquisas.

Os testes proveram a realização das curvas de PSNR vs bpp (*bit per pixel*). São mostradas medidas de cada imagem e da média do conjunto de imagens escolhidas. Os métodos revisados utilizam a decomposição *wavelet*. Para os testes utiliza-se um conjunto de imagens (*Airplane, Baboon, Lenna, Barbara, Goldhill, Peppers, Sailboat e Satellite*). Assim, determina-se o comportamento de uma TWD utilizando um filtro bior4.4 do Matlab. O filtro bior4.4 corresponde a um filtro *wavelet* biortogonal de ordem 4.4. Nesse teste se utiliza um número limitado de *bits* para representar cada coeficiente a fim de determinar o comportamento no módulo TWD. O parâmetro *shiftdec* indica o deslocamento do ponto decimal para inserir maior precisão.

Na Fig. 2.10, utiliza-se um conjunto de figuras e o gráfico corresponde à média das curvas. Observa-se que quanto maior a quantidade de *bits* obtem-se uma melhor qualidade na reconstrução. O PSNR e o SSIM apresentam um comportamento esperado, no entanto a

precisão finita ($shiftdec=0, 1$) faz com que se tenha um limite de qualidade na reconstrução. Por exemplo, para um coeficiente que possui somente a parte inteira, com 15 *bits*, já se obtém o máximo desempenho possível. Para melhorar esse desempenho é necessário aumentar o número de casas decimais e por conseguinte o número de *bits* também (a figura pode ser gerada como caso específico na seção anterior). A melhor configuração e a mais utilizada é a de 13 *bits* por coeficiente (para essa configuração), porque se utilizando mais decimais não se afeta muito a reconstrução em aplicações que requerem alta compressão (não são utilizados os planos de *bits* menos significativos). Para os testes seguintes, utilizam-se especificamente, as imagens: *Airplane*, *Lena* e *Peppers*, com tamanho 512×512 *pixels* para ver o comportamento dos diferentes esquemas.

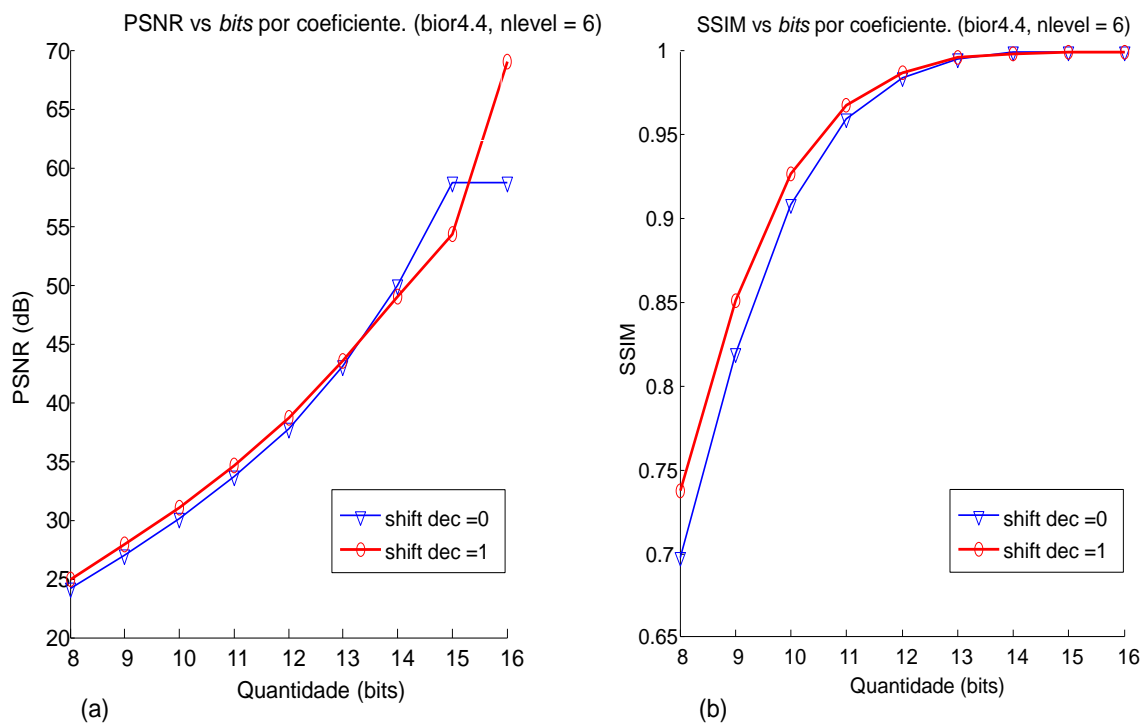


Fig. 2.10. Desempenho em PSNR e SSIM das imagens reconstruídas utilizando transformada *wavelet* com número limitado de *bits* para representar cada coeficiente inteiro e com uma casa decimal.

Na Fig. 2.11, o comportamento depende da imagem, mas a relação entre as diversas imagens fornece informação suficiente para concluir que o JPEG-2000 supera o algoritmo SPIHT em 1dB, aproximadamente. O esquema da recomendação CCSDS com a implementação TER (*TER Software*, <http://gici.uab.es/TER/>) sempre está muito próximo do algoritmo SPIHT, mas sempre abaixo em desempenho, insignificante em baixas taxas e piora um pouco à medida que a taxa é maior do que 0,8 *bpp*.

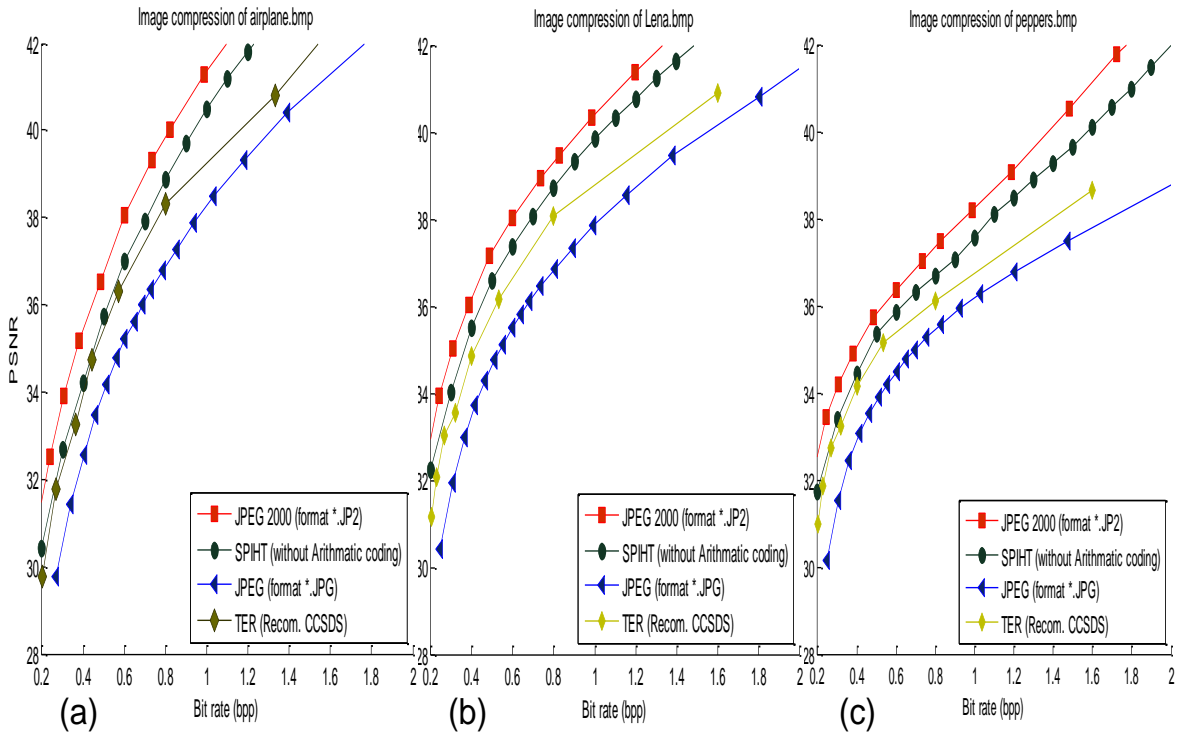


Fig. 2.11. Desempenho dos esquemas de compressão revisados em PSNR para imagens: a) *Airplane*, b) *Lena*, c) *Peppers*.

Para gerar a Fig. 2.12, é testado um conjunto de imagens completo e observa-se a relação que tem a utilização de quantidade de *bits* limite na TWD (ver Fig. 2.10) com a compressão utilizando o algoritmo SPIHT. Por exemplo, utilizando-se 10 *bits* para representar o coeficiente *wavelet*, o desempenho da compressão (30 dB) está limitado ao máximo desempenho da reconstrução *wavelet* (30 dB), ou seja, não melhorará. Novamente, a configuração de 13 *bits* para representar os coeficientes é muito parecida ao desempenho utilizando mais *bits*, nesse teste. Para baixas taxas (< 0.5 bpp), observa-se que são necessários somente 10 até 11 *bits*. Esse teste foi implementado em Matlab e pode ser de interesse do usuário para determinar a taxa de *bits* de acordo com as necessidades da qualidade desejada e os *bits* disponíveis para representar um coeficiente (diminuir custos).

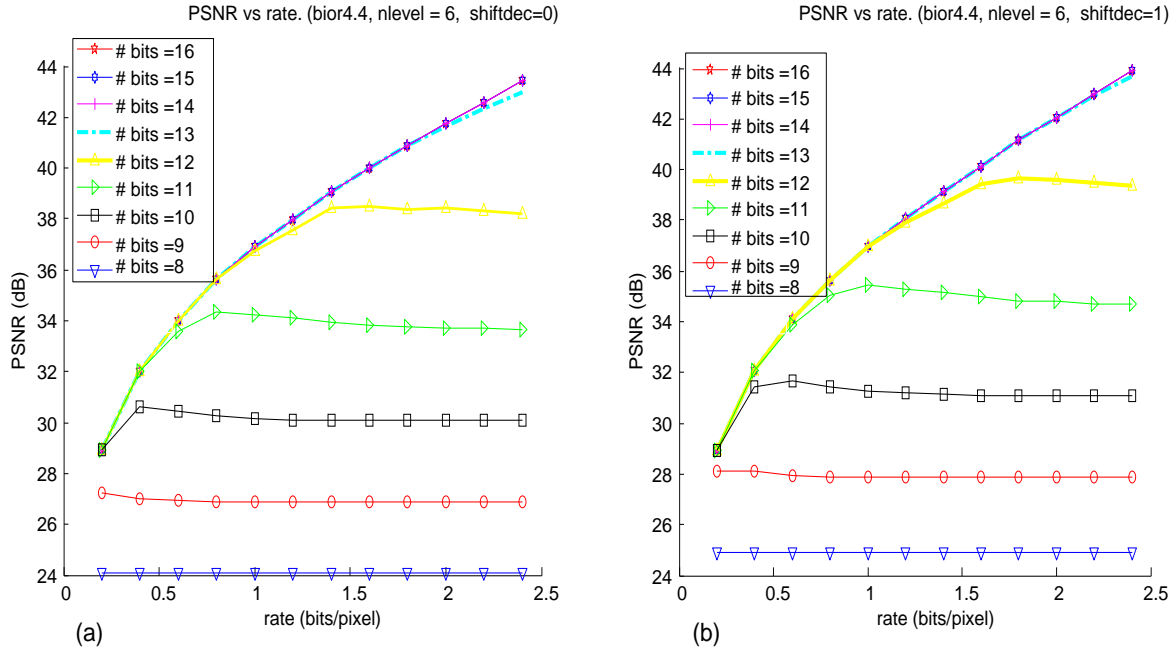


Fig. 2.12. Desempenho do algoritmo SPIHT utilizando um número limitado, (a) Utilizando somente a parte inteira dos coeficientes *wavelet*; (b) Utilizando a parte inteira e uma casa decimal.

2.8 Considerações finais

O objetivo primário neste capítulo é o de determinar qual algoritmo deve-se utilizar com sucesso para aplicações de transmissão de imagens visando um futuro uso para imagens de satélite. O algoritmo SPIHT consegue o melhor desempenho com uma complexidade menor. Sabe-se que o melhor desempenho em termos de compressão é o do JPEG-2000, pois se encontra maduro e tem sido melhorado até se tornar o mais sofisticado. A complexidade do JPEG-2000 de acordo com o número de módulos, torna esse padrão não desejável e dificilmente aplicável em um sistema embarcado em um satélite de pequeno porte. Por conta disso, a recomendação CCSDS indicou uma variação desse algoritmo, onde são retiradas partes, garantindo assim uma diminuição de complexidade. Essa recomendação está apresentada como padrão de compressão de imagens e tem a informação técnica pertinente, constantemente atualizada no *CCSDS Book Blue & Green* [81]. Por isso, neste trabalho, sugere-se aplicar um algoritmo rápido como SPIHT (ou variações) para sistemas de satélite de acordo com as necessidades *aplicação/taxa*. Para aplicar o SPIHT de forma a diminuir a utilização de recursos, recomenda-se a utilização dessa ferramenta para testar a necessidade suficiente para cada *aplicação/taxa*.

Capítulo 3

Pós-processamento da transformada *wavelet* para diminuir a memória da codificação SPIHT

3.1 Introdução

A transformada *wavelet* discreta (TWD) representa hoje uma importante ferramenta para a compressão de sinais multimídia. Isso se deve, especialmente, à sua habilidade em representar de forma eficiente componentes de alta frequência em imagens. Atingem-se altas taxas de compressão, ao utilizar alguns algoritmos sofisticados como o EZW (*Embedded Zerotree Wavelet coding*), SPIHT (*Set Partitioning in Hierarchical Trees*) [64], JPEG-2000 [82], e a recomendação de compressão de imagens do CCSDS (*The Consultative Committee for Space Data Systems*) [83, 84]. Aplicações como *fusion image* [85] utilizam os coeficientes da TWD como ferramenta para um processamento específico em sinais multimídia, em dispositivos de redes [86], de reconhecimento utilizando imagens [15], entre muitas outras aplicações [69, 87]. Todas essas aplicações são possíveis de serem implementadas em sistemas embarcados, incluída a compressão de imagens que requer um armazenamento considerável para o seu processamento. Uma TWD pode ser representada em M bits por coeficiente. Neste trabalho, o foco é reduzir a quantidade de memória durante o processamento.

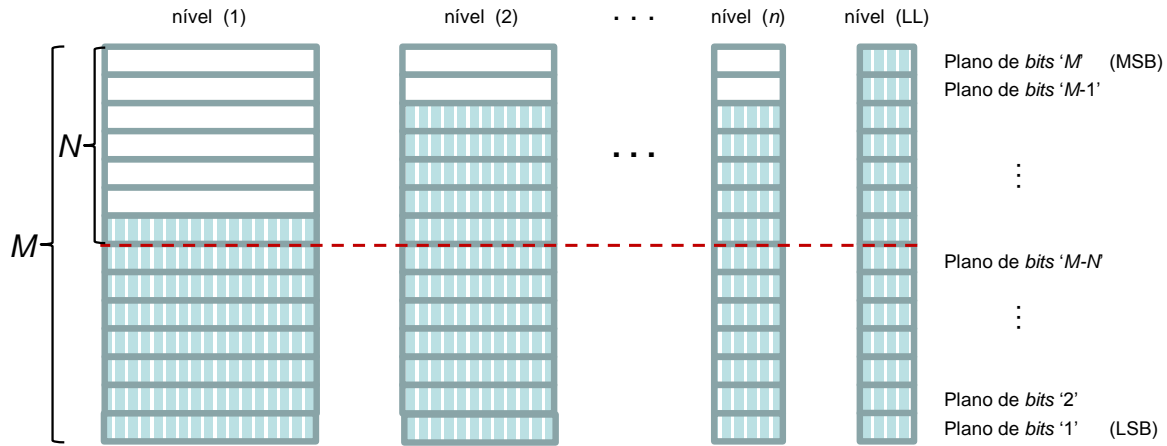


Fig. 3.1. Planos de *bits* utilizados na TWD para cada nível.

Na Fig. 3.1 pode-se observar a forma tradicional em que o valor de M pode ser reduzido, para uso de codificadores de planos de *bits*. Os *bits* mais significativos (MSB) de cada plano são codificados de acordo com um esquema de compressão (por critério de significância) até atingir uma taxa (com perdas $N < M$) ou fazer a varredura de todos os planos de *bits* (sem perdas $N = M$).

Como mostrado na Fig. 3.1, dos M *bits* é possível utilizar somente os primeiros N *bits* mais significativos (MSB) da TWD e, assim, processar os algoritmos de planos de *bits*, dependendo da aplicação. Utiliza-se, nesse caso, a característica da TWD de concentrar energia nos coeficientes da LL_i , que são uma aproximação da imagens. As outras subbandas, HL_i , LH_i , HH_i , representam os detalhes do sinal para cada nível.

No caso da compressão com o algoritmo SPIHT, é possível utilizar essa técnica [80] mostrada na Fig. 3.1. A TWD pode ser aplicada no SPIHT cortando-se os *bits* menos significativos, dado que, primeiramente, o algoritmo processa os planos mais significativos progressivamente.

Nesse capítulo, é proposto um esquema de codificação mostrado na Fig. 3.2, que é um método alternativo para a forma de se armazenar os coeficientes. Realiza-se um pós-processamento da TWD, que consiste em um reordenamento e uma interface (seção 3.2). Depois, usa-se um algoritmo de compressão de imagens SPIHT sem modificações (seção 3.3).

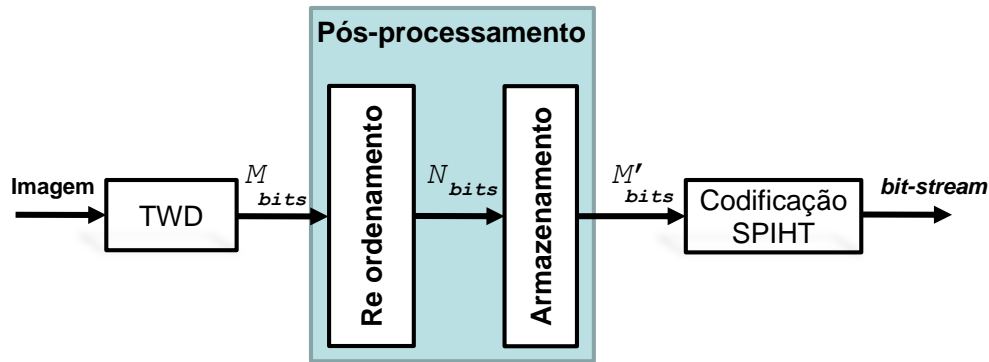


Fig. 3.2. Esquema de codificação SPIHT utilizando o pós-processamento proposto onde diminui-se o número de *bits* requerido para cada plano de *bits* de M até N *bits*.

3.2 Transformada *wavelet* com pós-processamento

A codificação que utiliza a metodologia de planos de *bits* como o algoritmo SPIHT proporciona eficiência e boa compressão. Permite ainda, a transmissão progressiva e baixa complexidade de processamento [64]. Alguns pesquisadores têm procurado, recentemente, melhores soluções para o problema de restrição de memória, isso no desenvolvimento de *hardware* do codificador de planos de *bits*. Uma das frentes consiste em diminuir a quantidade de memória, reduzindo o número de listas [5]. Outras frentes procuram modificar o codificador SPIHT [6], ou usar outros métodos. Outra alternativa para diminuir a utilização de memória envolve o módulo de TWD do algoritmo SPIHT.

Implementações de TWD com o objetivo de diminuir a memória usam esquemas que utilizam métodos para o cálculo da transformada de forma recursiva [88], ou métodos baseados em linha e também cálculo coeficiente a coeficiente (*line-based*) [7]. Tais métodos reduzem a utilização de memória no módulo da TWD dando ênfase ao método associado ao cálculo. Em [8] é utilizado um pós-processamento das amplitudes da TWD, reduzindo a quantidade de *bits* necessária para representar cada coeficiente.

Para a codificação SPIHT, os coeficientes da transformada *wavelet* precisam ser armazenados em um banco de memória para o processamento, pois os coeficientes são primeiramente verificados pela significância (acesso a planos de *bits* mais significantes) e, posteriormente, refinados em cada um dos passos de codificação. Isso exigiria uma grande quantidade de espaço de memória, especialmente, quando o tamanho da imagem é grande e a decomposição da TWD for de nível alto [76].

A Tabela 3.1 mostra o número de *bits* necessários para cada coeficiente em cada nível da TWD para um conjunto de oito imagens (*Airplane*, *Baboon*, *Lenna*, *Barbara*, *Goldhill*, *Peppers*, *Sailboat* e *Satellite*). A TWD utilizada é biortogonal (bior4.4) e somente é utilizada a parte inteira. O cálculo foi feito em um módulo Matlab (MathWork®).

Tabela 3.1: Quantidade de *bits* necessária para representar cada coeficiente nos níveis da TWD.

Nível TWD	Subbanda TWD	Tamanho	Média de <i>bits</i>	Máximo de <i>bits</i>
LL (nível 6)	LL_6	8×8	15	15
nível 6	HL_6, LH_6, HH_6	8×8	11	13
nível 5	HL_5, LH_5, HH_5	16×16	9	13
nível 4	HL_4, LH_4, HH_4	32×32	8	12
nível 3	HL_3, LH_3, HH_3	64×64	7	11
nível 2	HL_2, LH_2, HH_2	128×128	6	10
nível 1	HL_1, LH_1, HH_1	256×256	5	9

Na Tabela 3.1, a penúltima coluna que corresponde à ‘Média de *bits*’ representa a média de *bits* necessária para representar um coeficiente em uma determinada subbanda. Esse comportamento ocorre porque existem muitos coeficientes em cada nível, como o valor médio geral (muitos valores próximo de zero), e poucos valores altos ou picos que não afetam essa média. Porém, é necessário lembrar que essas informações são as mais importantes [76]. Na última coluna da Tabela 3.1 tem-se o máximo de *bits* necessário para representar a todos os coeficientes da respectiva subbanda. Observa-se que com o aumento dos níveis de decomposição TWD, existe um conseqüente aumento do número de *bits* necessário para representar os coeficientes [89]. Mostra-se que os maiores coeficientes situam-se na região *LL* do nível mais alto (característica de concentração de energia), e o número de *bits* necessário para representá-lo é de 15 para esse grupo de teste. Assim, a implementação do codificador SPIHT de planos de *bits* requer um mínimo de *bits* variável para cada subbanda. Nesse caso, a TWD precisa de 512×512 coeficientes de 15 *bits* cada. Isso exigiria uma grande quantidade de armazenamento de memória. Utilizando-se 15 *bits* para qualquer coeficiente, por exemplo, nas subbandas HL_1, LH_1, HH_1 do primeiro nível que consiste em 3/4 partes de toda a matriz *wavelet* somente são utilizados no máximo 9 *bits*. Assim, 6 *bits* por coeficiente referente a essas 3 regiões são desperdiçados, isso porque não são utilizados. Para os níveis seguintes, tem-se um comportamento parecido, totalizando cerca de 37,78% de alocação de memória não utilizada (última coluna da Tabela 3.1).

A Fig. 3.3 mostra a ideia de se obter as posições dos planos *bits* MSB_i para cada nível i (passo 1). Observa-se que para cada nível, tem-se uma posição diferente. Assim, se estabelece o vetor de posições $vMSB$ como na equação (3.1) (sinalizado pela seta vermelha). Também podem-se aplicar outras formas personalizadas para definir esse vetor.

$$vMSB = \{MSB_1, \dots, MSB_i, \dots, MSB_n, MSB_{LL_n}\} \quad (3.1)$$

Na representação de cada coeficiente, existe um *bit* de sinal (nessa representação é o mais significativo do coeficiente). Na Fig. 3.3, nota-se que na LL_n , o sinal é de valor conhecido (sempre positivo por ser aproximação de uma imagem). Assim, o mais significativo na LL_n é o plano de *bits* $MSB_{LL} - 1$ (seta azul).

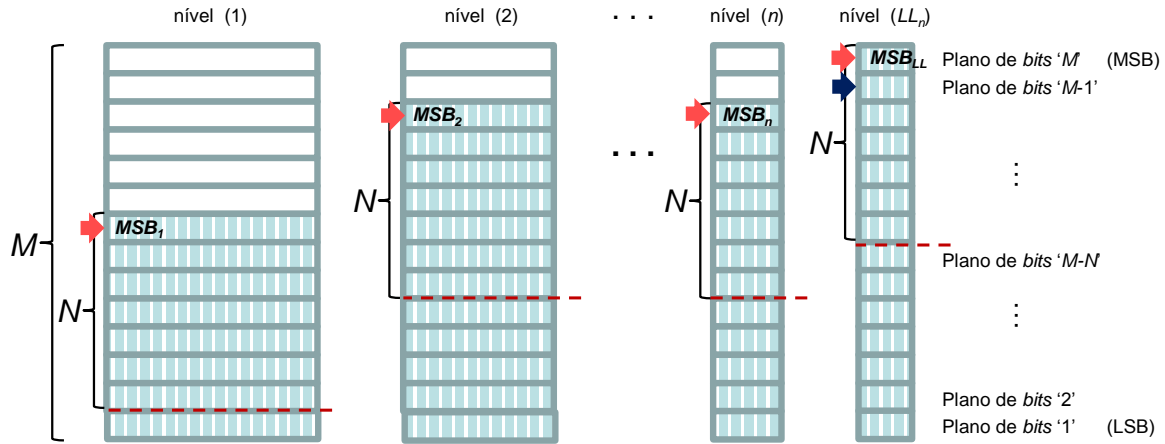


Fig. 3.3. Vetor que estabelece o MSB de cada subbanda.

A fim de reduzir a memória necessária para armazenar os coeficientes da TWD, é proposto um método de pós-processamento para redefinir a amplitude dos coeficientes (passo 2). Um reordenamento de um plano de *bits* do nível 1 (que utiliza menos *bits*) é utilizado para compensar os outros níveis superiores, para que o tamanho de armazenamento seja equivalente a N *bits* (passo 3). Isso é possível dado que a área de um nível ' n ' equivale a 3 vezes a área do nível $n+1$. Assim, um plano de *bits* do nível 1 (sem LL_1) equivale a 3 planos de *bits* para o restante dos níveis. Finalmente, a interface considera como zerada aquela parte que não se consegue salvar (passo 4).

Passos do esquema

O pós-processamento proposto é esquematizado na Fig. 3.4.

Os 4 passos que são detalhados e exemplificados a seguir.

O primeiro passo é obter o vetor de MSB_i . Tal vetor deve ser calculado com critérios de um subgrupo como se viu nas últimas duas colunas da Tabela 3.1, ou por treinamentos utilizando-se outras técnicas. Nessa proposta, foi criada uma ferramenta que calcula e fornece informações, como aquelas apresentadas na Tabela 3.1. Também, se houver um adequado controle de *overflow*, é possível utilizar uma média ponderada entre a média de *bits* (penúltima coluna) e o número máximo de *bits* (última coluna).

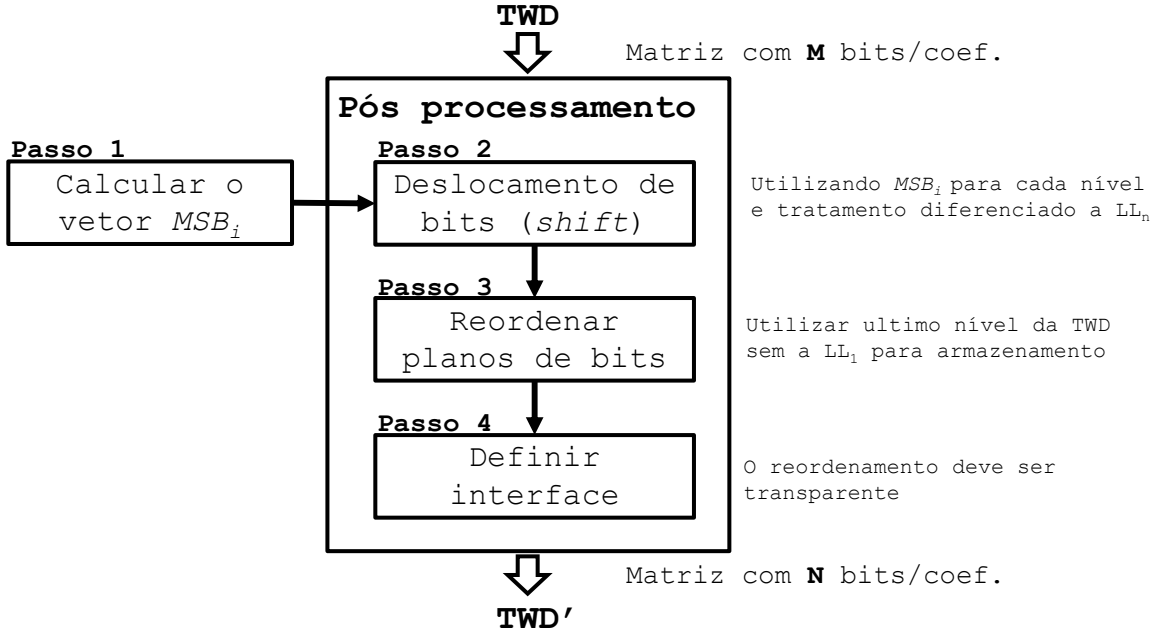


Fig. 3.4. Fluxograma da proposta da transformada *wavelet* com pós-processamento.

O segundo passo, mostrado na Fig. 3.5, corresponde a um reordenamento (*shift*) de cada nível. Tal operação tem como finalidade fazer com que o *bit* mais significativo MSB_i corresponda ao plano de *bits* M em todos os níveis. Especificamente, o reordenamento nos planos de *bit* na região LL_n corresponde a $MSB_{LL} - 1$, o plano MSB_{LL} não é considerado (por não ter variação nessa representação, ou seja é sempre o mesmo sinal positivo). Assim, o vetor $vMSB$ é uma referência do novo ordenamento (é fixo).

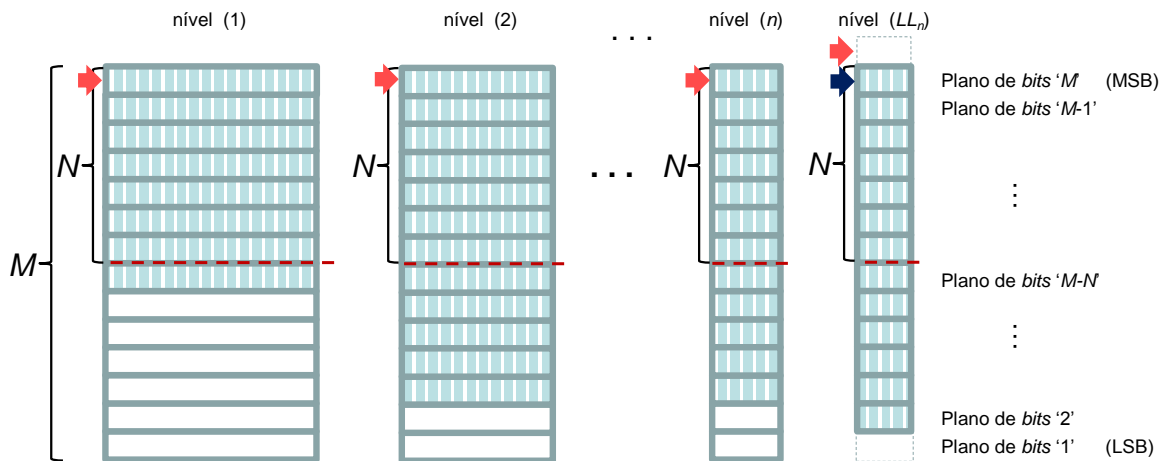


Fig. 3.5. Passo de reordenamento utilizando o vetor $vMSB$ do método de pós-processamento proposto para a TWD.

O terceiro passo mostrado na Fig. 3.6, utiliza um plano de *bits* ' $M-N+1$ ' do nível 1 para salvar nessa região os planos de *bits* ' $M-N$ ', ' $M-N-1$ ', ' $M-N-2$ ' dos outros níveis e LL_n . Cada quadrante HL_1 , LH_1 , HH_1 do nível 1 tem então os dados dos planos de *bits* respectivos.

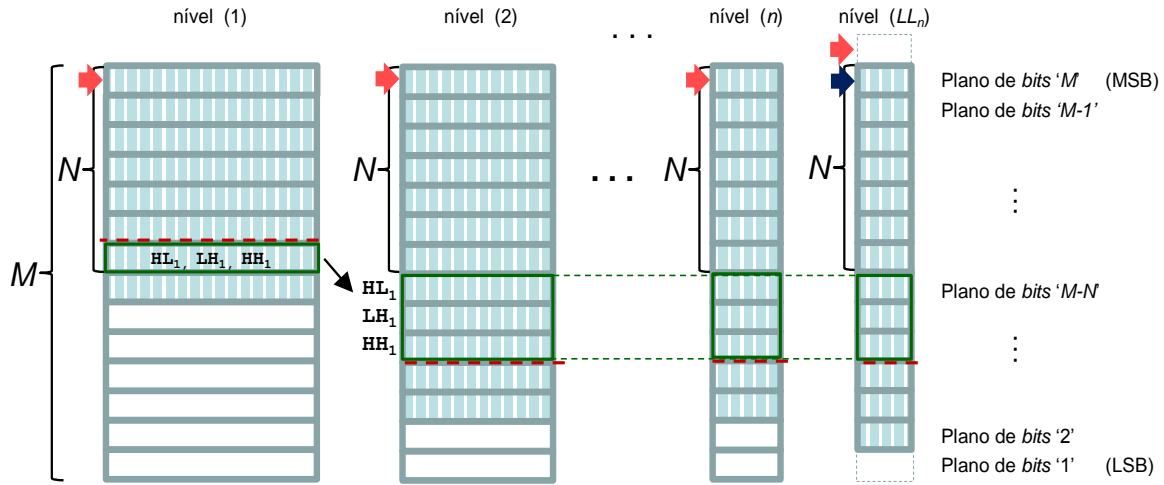


Fig. 3.6. Passo de utilização do último plano necessário (nível 1) para armazenar planos de *bits* dos outros níveis da TWD.

Depois desse passo (passo 3) já se tem a memória física reduzida, dado que são utilizados somente N bits por coeficiente (equivalente). Na Fig. 3.7, observa-se como fica a memória depois desses passos, utilizando $M-N+1$ até M planos de *bits*. O resto pode ser utilizado para outros propósitos, não alocado ou simplesmente liberado.

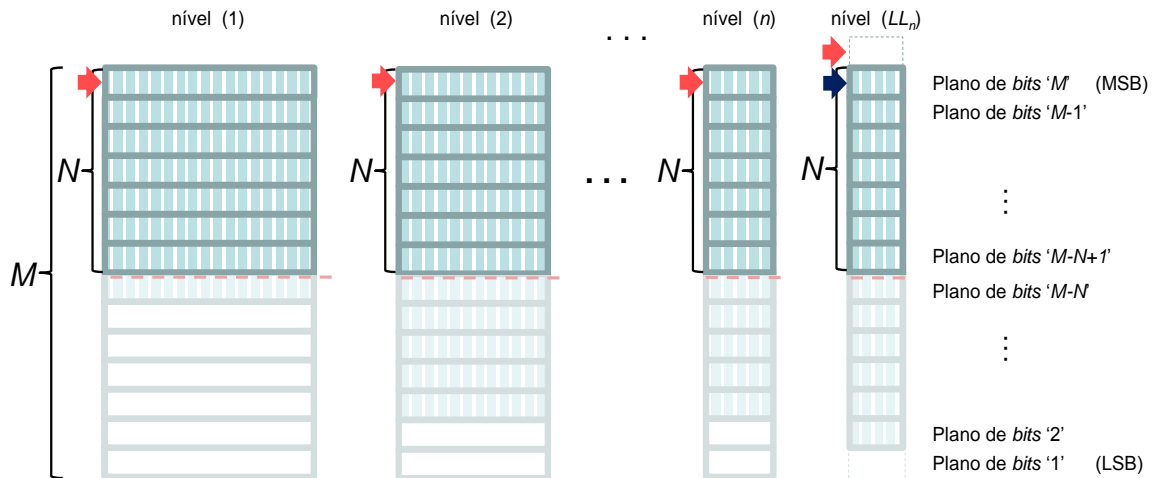


Fig. 3.7. Representação da memória física utilizada: N de M bits, depois do pós-processamento.

No quarto passo, o novo ordenamento anterior deve ser transparente para as aplicações. Assim, a interface de armazenamento mostra os N bits físicos como M_v bits virtuais com alguns planos bits zerados (de acordo com os passos anteriores e com o vetor $vMSB$).

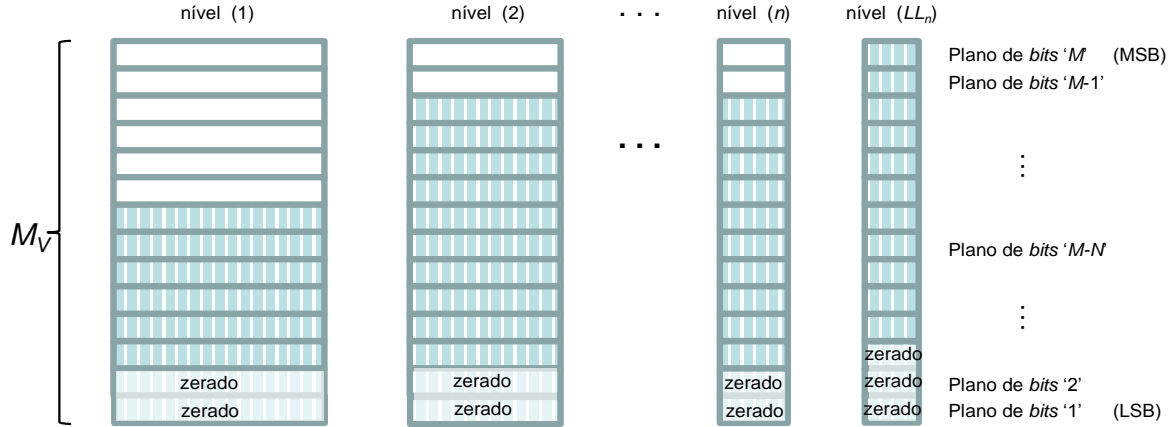


Fig. 3.8. Memória interpretada pela interface no quarto passo da proposta.

Para representar um coeficiente *wavelet* em qualquer nível i , foram utilizados N bits físicos e foram recuperados M_v bits virtuais. Daqueles $M_v = M$ bits virtuais, proporcionados pela interface, os menos significativos de cada nível normalmente são perdidos (ver 'zerados' na Fig. 3.8). No caso específico do nível 1, somente são salvos $N-1$ planos de bits desse nível. Para outros níveis $i=\{2:n\}$, são salvos sempre $N+3$ bits e para a região LL_n , são salvos $N+4$ bits.

3.3 Codificação SPIHT

O codificador SPIHT possui um algoritmo que explora as similaridades entre subbandas em uma decomposição *wavelet* de uma imagem. Em linhas gerais, o algoritmo utiliza primeiro os coeficientes considerados mais importantes. Logo, gera um *bitstream* com os bits desses coeficientes, sendo refinados passo a passo. Assim, é possível obter a imagem original progressivamente. Esse codificador utiliza a metodologia de codificação de planos de bits. Neste trabalho é utilizada a implementação SPITH sem modificações [64] onde a TWD utiliza M bits para a representação do coeficiente. O algoritmo SPIHT está detalhado em [64, 80, 76]. Basicamente, o codificador SPIHT usa um particionamento das árvores de forma a manter os coeficientes *wavelet* insignificantes, melhor agrupados em subconjuntos maiores [68]. Na codificação, um coeficiente é considerado como significativo se o seu valor é maior ou igual ao limiar T , ou como insignificante se seu valor é menor que T . Há dois passos na codificação do SPIHT; a ordenação *sorting* e o passo de refinamento. O diagrama geral, com ênfase ao acesso a memória, é apresentado na Fig. 3.9.

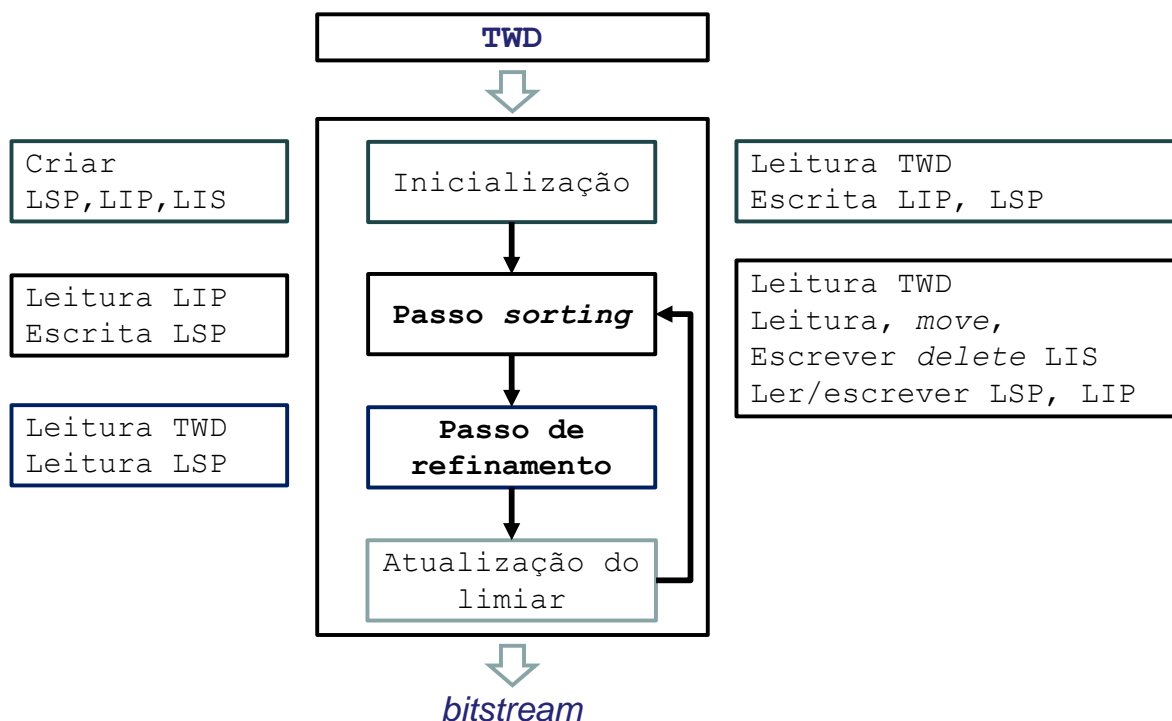


Fig. 3.9. Diagrama de blocos do algoritmo SPIHT enfatizando o acesso à memória.

A codificação de planos de *bits* tradicional exige que a matriz da TWD seja calculada e armazenada em uma memória para a codificação SPIHT. Isso requer um grande espaço de memória que somente será utilizado para leitura. Como se observa na Fig. 3.9, o acesso à memória da TWD pode ser verificado no algoritmo em [64] como característica desse codificador (*encoder*) de acessar um plano de *bits* por passo e não por coeficiente completo. Por ter esse comportamento, a segmentação produzida pela proposta de um coeficiente de forma não sequencial, não é problema para esse tipo de esquema.

A codificação *strip-spiht* em [90] mostra uma implementação que utiliza pouca memória de codificação SPIHT. Trata-se de armazenar algumas linhas de coeficientes *wavelet* em um *strip-buffer* e a codificação SPIHT então é feita de forma *strip-base* calculando parte da TWD e gerando o *bitstream* do SPIHT. Nessa mesma área de pesquisa, o trabalho publicado em [75], utiliza menores níveis de decomposição TWD em conjunto com uma nova estrutura de árvore SOT-C, e tem conseguido reduzir mais a memória necessária para a codificação SPIHT no esquema *strip-base*. O trabalho publicado [75] utiliza uma codificação específica no módulo da TWD fazendo uma codificação por subbanda para reconstruir o coeficiente, adicionando o valor de de-quantização ξ [8, 80].

3.4 Simulações e resultados

O *software* para simulação utilizado foi o Matlab (MathWork®). Nesse *software* foi criada a ferramenta para tratamento de *bits* da TWD. Também foi carregado o codificador SPIHT [64] personalizado para depuração *bit a bit*. Ambos foram inseridos no procedimento genérico de simulação desenvolvido como mostrado na Fig. 3.10. O conjunto de imagens utilizadas 512×512 *pixels* para fazer o teste são: *Airplane, Baboon, Lenna, Barbara, Goldhill, Peppers, Sailboat e Satellite*. A TWD utilizada é biortogonal (MathWork® bior4.4), com 6 níveis de decomposição, e foi utilizada a parte inteira dos coeficientes.

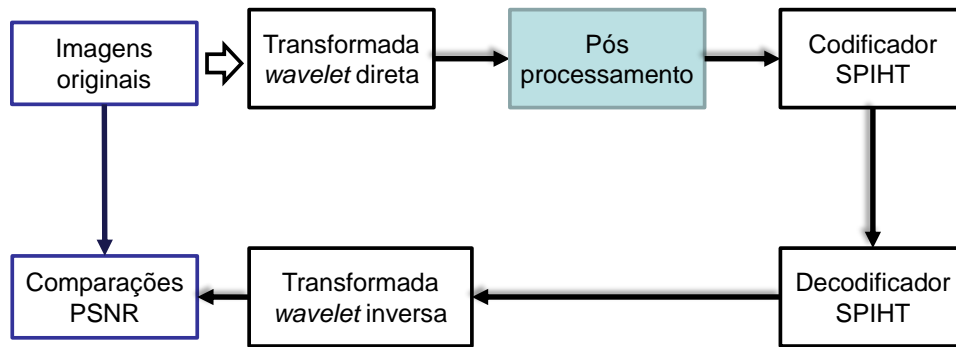


Fig. 3.10. Teste para executar a TWD, o pós-processamento proposto e o SPIHT.

O vetor v_{MSB} pode ser personalizado de acordo com dados da Tabela 3.1. Nessa proposta, a construção desse vetor obedece a regra dada a seguir para otimizar os planos utilizados.

$$v_{MSB} = \{N - 1, N + 3, \dots, N + 3, \dots, N + 3, N + 5\} \quad (3.2)$$

Tem-se: $2 < N \leq 12$ a restrição de comprimento da TWD que gera coeficientes com 15 *bits* para esse teste. Assim, o valor limite do vetor é: $v_{MSB} = \{11, 13, \dots, 13, 15\}$. Os resultados da simulação são mostrados na Fig. 3.11, onde o SPIHT utiliza uma TWD com $N = 11, 12, 13, 14, 15$ (*bits*) e a proposta do SPIHT com TWD com pós-processamento utiliza $N = 6, 7, 8, 9$ (*bits*).

A Fig. 3.11 representa a curva do desempenho SPIHT em PSNR com taxa controlada de 0,2 *bpp* até 1,2 *bpp*. Na Fig.3.12, tem-se uma análise dos resultados da Fig.3.11. Nessa figura, pode-se observar que N diminui e o desempenho chega a um patamar onde não melhora mais. No entanto, esse patamar no SPIHT com pós-processamento para $N=9$ e taxas (de até 1 *bpp*) é melhor que o SPIHT de $N=12$ e de igual desempenho com $N=13, 14$ ou 15 *bits*. Esse patamar é definido na proposta pela quantidade de *bits* zerados para que o algoritmo veja os M_v . Isso, inclusive, pode atrapalhar na operação da *wavelet* quando os

planos de *bits* zerados estiverem não homogêneos (para $N=6$, 7 gera-se uma leve queda depois dos 1,0 bpp). O PSNR, nessa curva, para $N=8$ com uma taxa de até 1,0 bpp apresenta uma insignificante variação (até 0,6 bpp é igual o desempenho ao original $M=15$) utilizando 7 planos de *bits* menos, ou seja, utilizou somente 53% do original (para $M=15$) com desempenho muito aproximado.

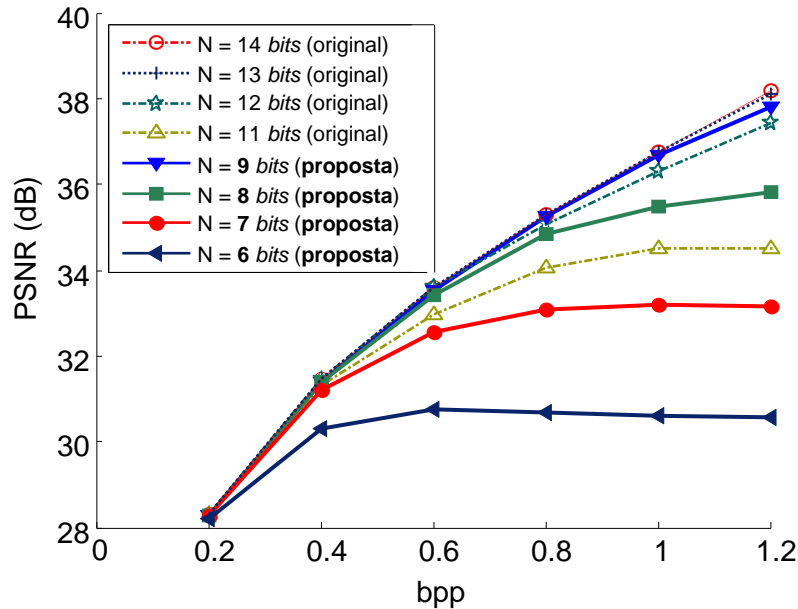


Fig. 3.11. Desempenho da proposta de codificação SPIHT usando TWD com pós-processamento em termos de relação sinal ruído (PSNR).

Na Fig.3.12.a observa-se conforme indicada a configuração de 11 *bits* do SPIHT sem pós-processamento. Ressalta-se a curva mais próxima que corresponde a $N=8$, número usado na proposta. A melhoria é de 2dB aproximadamente. Na Fig.3.12.b se observa que para uma taxa entre 0,2 até 0,4 bpp, a proposta para $N=7$ tem um comportamento igual que para $N=12$ (ou maior). Na Fig.3.12.c se observa que para uma taxa de 0,6 até 0,8 bpp, a proposta com $N=8$ o comportamento é parecido e muito próximo a $N=12$ (ou maior). Na Fig. 3.12.a observa-se que para uma taxa de 0,6 até 0,8 bpp, a proposta com $N=9$ tem comportamento igual a $N=14$. Em cada uma dessas comparações vemos que o valor de N diminui com a proposta de pós-processamento.

Em resumo, a proposta do SPIHT que utiliza TWD com pós-processamento $N=9$ apresenta um desempenho parecido com o daquele que não utiliza a proposta com $N=15$ (maior do que 0,8 bpp). Para taxas menores do que 0,8 bpp, o desempenho é equivalente ao original $M=15$. Assim, pode-se dizer que o método de pós-processamento aqui proposto é importante para a redução da quantidade de memória a ser utilizada no módulo TWD do codificador SPIHT.

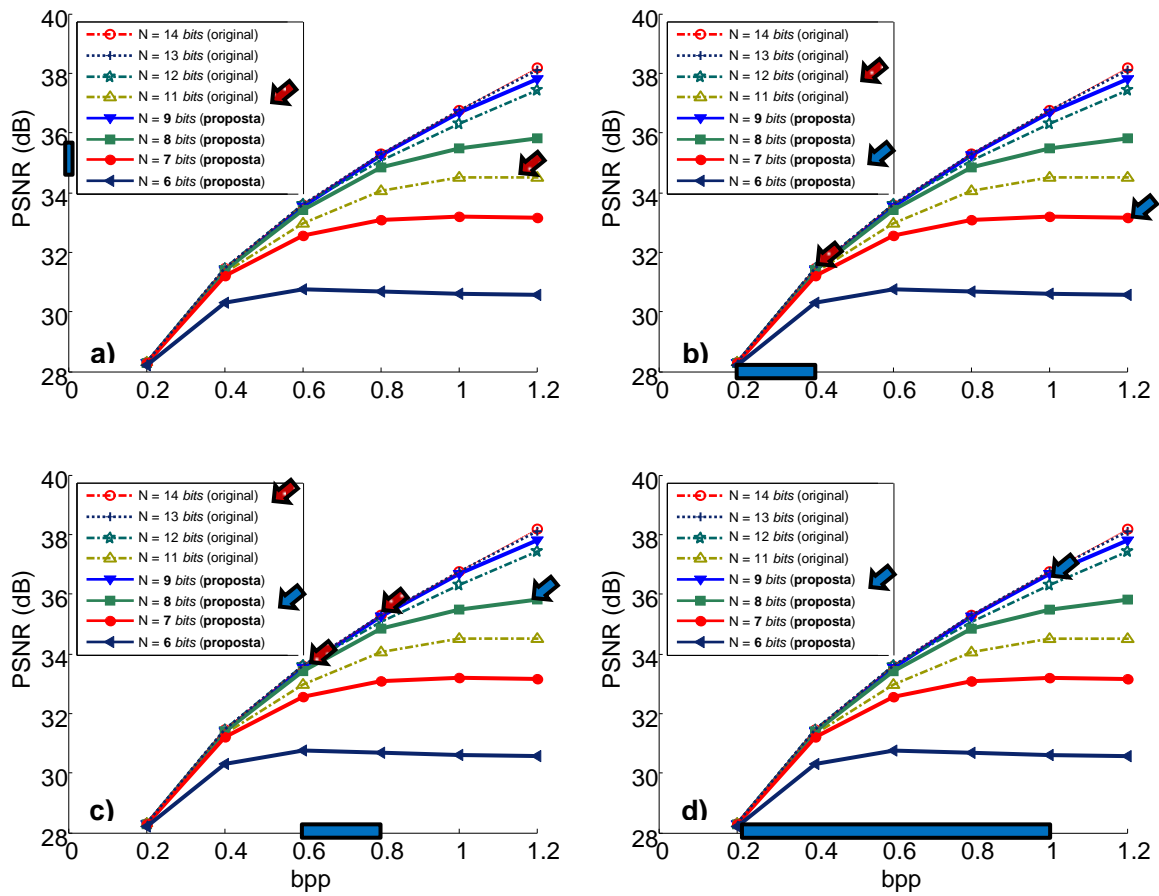


Fig. 3.12. Análise de alguns pontos sobre o resultado do desempenho.

Capítulo 4

Codificação de região de interesse generalizada aplicada ao SPIHT

4.1 Introdução

A codificação da região de interesse ROI (*Region of Interest*) de uma imagem permite codificar regiões dentro da imagem com melhor qualidade ou importância que outras, por exemplo, o fundo da imagem BG (*Background*). Para comunicações e aplicações de armazenamento, é desejável que a imagem esteja substancialmente comprimida. Na compressão, a correlação entre *pixels* é explorada e erros podem aparecer em forma de ruído. Esse ruído pode ocorrer durante a captura, transmissão ou no processamento da imagem, e pode ser dependente ou independente do conteúdo [91]. Para contornar os problemas de ruído é necessária a hierarquização da imagem. A codificação ROI é importante para compressão e também influi diretamente na transmissão com prioridade ou grau de interesse [92]. Primeiramente, as informações importantes (ROI) são enviadas e depois, se necessário, envia-se o resto (BG).

A codificação ROI pode ser aplicada em esquemas de compressão de alto desempenho como o JPEG-2000, onde o uso de ROI é bem pesquisado. No entanto, o padrão H.264/AVC de compressão de vídeos tem desempenho superior na compressão de imagens estáticas (*intra-frame*) da ordem de 0,25 até 0,5 dB do valor de PSNR, dessa forma, superior ao JPEG-2000 [93, 94]. O H.264, porém, não foi utilizado neste trabalho pela alta complexidade perante o JPEG-2000.

Também se pode ressaltar o alto consumo computacional de processamento no JPEG-2000 [95, 74], devido à sua estrutura que contém vários codificadores (também dependendo da implementação). Isso faz com que o SPIHT seja melhor para aplicações de baixo consumo de energia com um bom desempenho na qualidade e possível aplicação da ROI. Isso porque a quantidade de operações ou complexidade é menor no SPIHT que no JPEG-2000. Além disso, em geral, considera-se que a complexidade e o consumo de energia estão relacionados diretamente.

Neste trabalho, utilizou-se o algoritmo SPIHT pela sua baixa complexidade (ou seja, baixo consumo de energia) e um desempenho em PSNR próximo ao JPEG-2000. Outro

fator é o seu desempenho superior ao JPEG, como se observa na Fig. 4.1. Nessa figura, observam-se três resultados correspondentes às imagens *Lena*, *Airplane* e *Boat*. Essas imagens são de conteúdos de diferente natureza. Os resultados mostram um comportamento regular para todas elas.

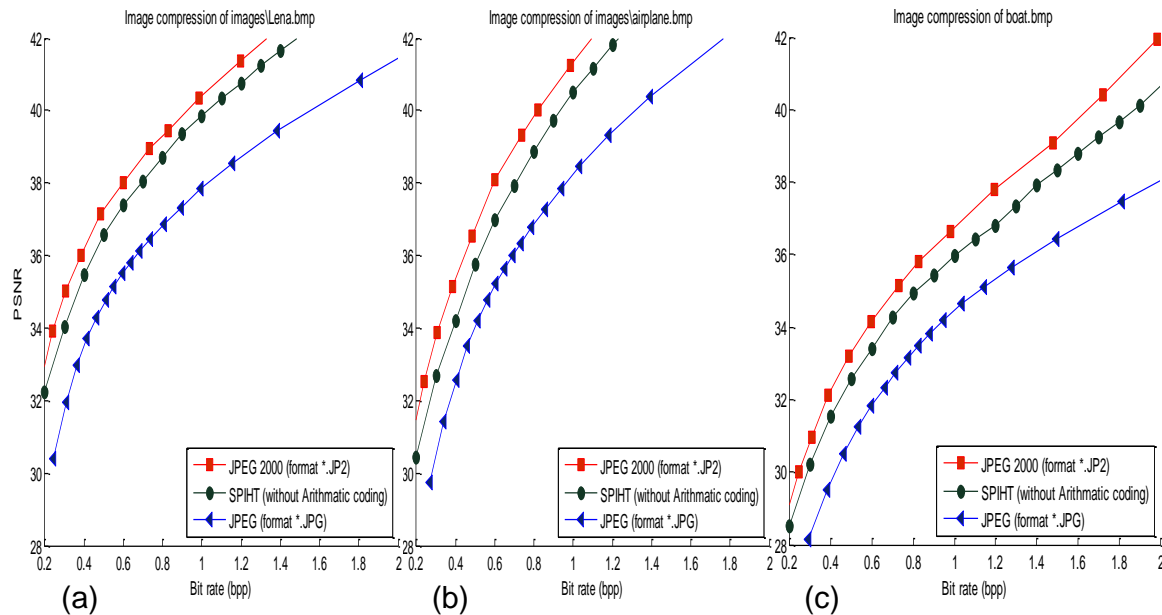


Fig. 4.1. Desempenho do JPEG-2000, SPIHT e JPG.

Nas publicações existentes, a codificação ROI é avaliada utilizando-se métricas objetivas, tais como o PSNR e também por comparações subjetivas. A codificação ROI pode melhorar efetivamente a qualidade subjetiva da imagem, no entanto, existem algumas desvantagens conhecidas nos métodos existentes da codificação ROI [96]. Neste trabalho não será aplicada nenhuma comparação subjetiva e como métrica será utilizada a PSNR para comparação.

A forma de codificar a região de interesse ROI (*Region of Interest*) e o fundo BG (*Background*) está baseada comumente na obtenção de regiões de coeficientes da transformada *wavelet* de uma imagem que correspondem a uma ROI desejada (métodos baseados em TWD [12, 64]). Esses coeficientes são escalonados de diversas formas segundo propostas e padrões estabelecidos. A TWD tem propriedades que são exploradas estatisticamente pelos codificadores (ver seção 2.2).

Neste trabalho será abordada a codificação ROI baseada em *wavelet*.

A ROI tem aplicações além da compressão de imagens, por exemplo, na parte de transmissão de imagens (ex. telemedicina, segurança, buscadores, redes sociais), proteção de erros (codificação de canal [97]), e detecção de objetos. Para sistemas mais completos de

imagens, a ROI pode ser utilizada em conjunto com outras finalidades como *watermarking* [98], transmissão *wireless* [17], e na área médica [99].

Atualmente o JPEG-2000 é um esquema padrão de compressão de imagens utilizando ROI (ver seção 4.2) e existem diversas propostas acadêmicas para ele (compatíveis ou não). Esquemas menos complexos e de desempenho próximo ao do codificador JPEG-2000, como o EZW [9], SPIHT (e variantes propostas) existem na literatura para a codificação ROI [8, 74, 75, 6, 100, 101, 102, 103].

Além da codificação ROI única, existe a codificação de múltiplas regiões de interesse (*multi-ROI*) como o *HBSHift* [104]. Também existem outros esquemas com ênfase à eficiência [105], utilizando a inversa da transformação *wavelet* [106], entre outros esquemas como o *PSBshift* [107]. A ideia de fazer isso é gerar prioridades dentro da mesma imagem.

Um dos problemas encontrados para aplicar ROI no algoritmo SPIHT básico sem modificações é a pouca quantidade de pesquisas nessa área, isso se comparado ao JPEG-2000. As propostas aplicadas ao JPEG-2000 com uma adaptação ou estudo suficiente podem ser aplicadas com alguma vantagem no algoritmo SPIHT.

Neste trabalho, a codificação ROI é aplicada no algoritmo SPIHT de forma transparente, ou seja, a entrada do SPIHT será a TWD modificada pela codificação ROI proposta.

A vantagem de se ter um algoritmo como o SPIHT juntamente com a ROI é conseguir diminuir os custos em processamento em relação ao JPEG-2000. Outra vantagem é que o desempenho de qualidade não cai, dado que a qualidade fica próxima ao JPEG-2000. O algoritmo SPIHT utiliza a seguinte estratégia para organizar os dados dos coeficientes *wavelet*: uma estrutura de conjuntos particionados em árvore para aproximar campos insignificantes ao longo de bandas descendentes. Utiliza e explora as propriedades estatísticas da TWD de modo que se for encontrado um coeficiente pai zero, a probabilidade de seus descendentes serem zero é muito alta (*zerotree*), principalmente a partir dos netos [47, 80].

O algoritmo SPIHT utiliza características da *wavelet* (exceção da clusterização) ao longo de todos os planos de *bits*. Assim, caso se encontre um coeficiente significativo entra-se em um subconjunto que será enviado *bit a bit* sem compressão (atua diretamente sobre os não significativos). Se existir algum tipo de deslocamento de *bits* significativos dos coeficientes, os espaços gerados sem informação são zeros e se enviados não são de utilidade (não desejável).

Para balancear as possíveis vantagens e desvantagens na aplicação de ROI são necessárias ferramentas e pesquisas que auxiliem na análise e comparação das várias configurações possíveis.

4.2 Codificação ROI no JPEG-2000

O JPEG-2000 [94, 12, 13] é considerado por muitos, o padrão no estado da arte em compressão de imagens (ver seção 2.5 JPEG-2000).

Os métodos mais conhecidos de escalonamento dos planos de *bits* (para codificar) ROI são o *MaxShift* e *ScaleShift*, que são usados como padrão no JPEG-2000 e utilizam regiões retangulares e elípticas [12, 13, 82]. Também existem propostas de codificações ROI flexíveis, como o *HBSHift* [104] e algumas formas de escalonamento ROI como o método *BbBSHift* [108] e o método *GBbBSHift* [109] observados na Fig. 4.2.

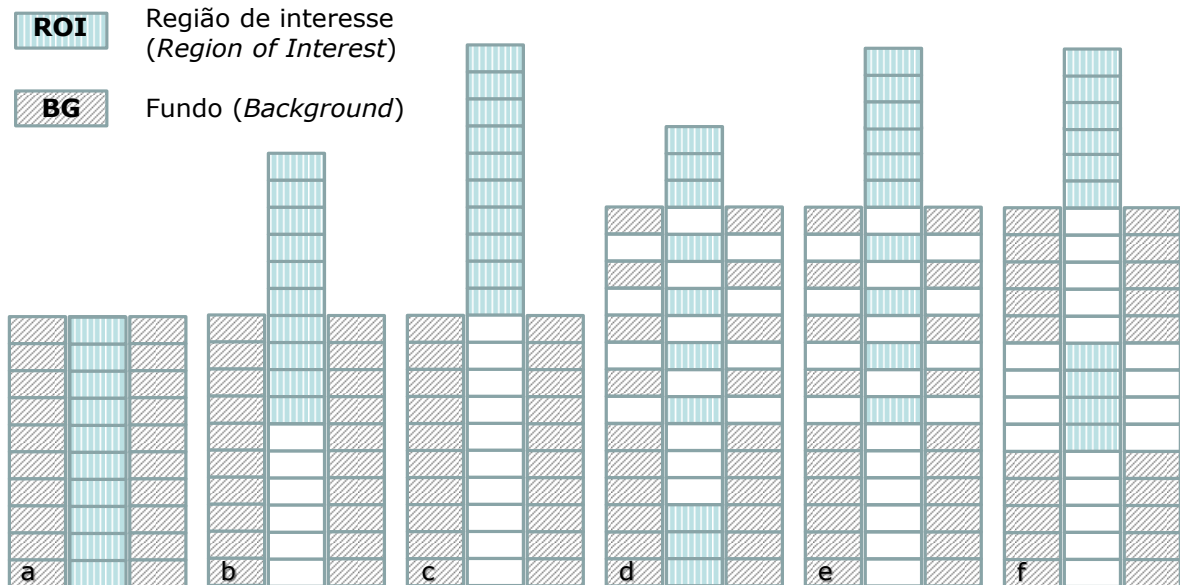


Fig. 4.2. Métodos de escalonamento dos planos de *bits* da ROI a) Não ROI; b) Método *ScaleShift* de escalonamento geral $s=6$; c) Método *Maxshift* $s=10$; d) Método híbrido *HBSHift* $s_1=3$ e $s_2=4$; e) Método *BbBSHift* $s_1=6$ e $s_2=4$; f) Método *GBbBSHift*.

Como se observa nessa Fig. 4.2, a matriz que contém a TWD vista de lado, mostra que os *bits* superiores são os mais significativos e os inferiores os menos significativos. A TWD sem modificações está representada na Fig. 4.2.a, onde não é codificada a ROI. Nas Fig. 4.2.b e Fig. 4.2.c tem-se a codificação ROI do JPEG-2000, onde os *bits* da ROI são deslocados para aumentar a sua significância. Nas Fig. 4.2.d e Fig. 4.2.e tem-se um método híbrido onde o deslocamento acontece tanto na ROI quanto no BG de uma forma parametrizada (intercalada). Isso permite especificar cada uma delas utilizando os parâmetros s_1 e s_2 . Finalmente, tem-se na Fig. 4.2.f, como exemplo de codificação ROI, o método *GBbBSHift*. O deslocamento acontece em ambas as regiões ROI e BG, mas com apenas 4 partes dos *bits* especificadas com parâmetros, sem solapamento e na seguinte ordem específica: (1) primeira parte da ROI, (2) primeira parte do BG, (3) segunda parte da ROI e (4) BG.

Desvantagens da codificação ROI no JPEG-2000

O método *maxshift* é uma forma simples e eficiente de se codificar uma região de interesse. No entanto, apresenta desvantagens relatadas em [109, 104]. A primeira é que esse método não tem flexibilidade ao definir a importância da ROI e BG (falta de *scaling* arbitrário). Isso pode fazer com que sejam priorizados os últimos planos de *bits* (*bitplanes*) da ROI que é a informação de ruído e detalhes imperceptíveis. A segunda desvantagem é que o método decodifica toda a ROI (somente) para logo depois acessar o BG. Isso gera deslocamentos (*shift*) compridos que incrementam significativamente o número total de *bitplanes* (não desejado no codificador porque aumenta a utilização de memória para o processo). A terceira desvantagem é a não flexibilidade sobre navegadores e aplicações interativas (se o *download* for lento, provavelmente, será enviado somente a ROI até o usuário escolher outra imagem). Finalmente, quando se tem múltiplas regiões de interesse dentro de uma mesma imagem, a prioridade de cada região não pode ser diferente durante a codificação e transmissão.

Na codificação ROI, uma região é escolhida de forma estática ou dinâmica. As regiões de interesse estáticas são definidas no tempo de codificação, enquanto as dinâmicas são definidas interativamente (manual ou automaticamente) pelo usuário durante uma transmissão progressiva. No JPEG-2000, para se utilizar diferentes prioridades para diferentes regiões, é possível utilizar o método *scaling-based*, mas a forma dessas regiões somente pode ser retangular ou elíptica. Isso representa uma falta de flexibilidade, não sendo possível escolher regiões em outras formas.

4.3 Codificação da ROI no SPIHT

O estudo da ROI no SPIHT não é tão desenvolvido como no JPEG-2000, por isso é necessária uma forma generalizada para o teste. Muitas formas de aplicação da ROI são encontradas na bibliografia, mas na maioria delas, o algoritmo é modificado para obter os resultados [102, 103], dificultando a reutilização de módulos e aumentando a utilização de memória dinâmica [103].

Diversas aplicações no campo da telemedicina aplicam a codificação ROI pela alta quantidade de dados transmitidos. O SPIHT foi utilizado com um esquema para conservar a ROI (*lossless*) com a maior fidelidade [101] (também em aplicações de análise de texto [110]), isso utilizando esquemas de escalonamento [111] parecidos com o padrão '*scaling-based*' do JPEG-2000 (ver Fig. 4.2.b). Nos trabalhos sobre o SPIHT [8, 10, 6, 79], sempre é feita alguma modificação no algoritmo. Nesta proposta, para deixar o método transparente à codificação ROI, é apenas iniciado um módulo TWD, gera-se uma máscara que corresponde à região escolhida, codifica-se a ROI e os coeficientes modificados são enviados para o algoritmo SPIHT.

4.4 Codificação generalizada da ROI

Nesta seção é feito um resumo das formas de como se fazer uma codificação da região de interesse ROI e o fundo BG. Buscou-se aplicações que utilizam decomposição *wavelet* como o JPEG-2000, EBCOT, LZW, EZW, SPIHT, SPECK. Assim, é possível utilizar as vantagens dessa transformada para priorizar regiões de interesse na compressão, transmissão, detecção de formas e objetos, entre outros propósitos.

Neste trabalho é feita uma implementação de um esquema generalizado de codificação ROI utilizando uma máscara de planos de *bits roi_mask* para a ROI [109], e para definir a forma de escalonamento da ROI e o fundo BG. Esses dois parâmetros são chamados aqui de máscara de vetor de escalonamento *vsROI* e *vsBG*. Essa implementação permite generalizar, com dois vetores, qualquer outra codificação ROI utilizando vetores binários, incluindo *general scaling-based*, *maxshift*, e outras variantes de codificações com e sem sobreposição. Um exemplo é apresentado na Fig. 4.3, onde se observa a forma de codificação da ROI e a flexibilidade ao priorizar os planos de *bits* com vetores binários. Isso é importante porque para cada forma de priorizar pode-se adaptar ao comportamento dos planos de *bits*, que são influenciados por várias fontes seja da imagem, do filtro *wavelet* utilizado ou do codificador de planos de *bits* (diferente para cada tipo de aplicação). A forma flexível pode adaptar-se e retornar um melhor desempenho em PSNR, por exemplo.

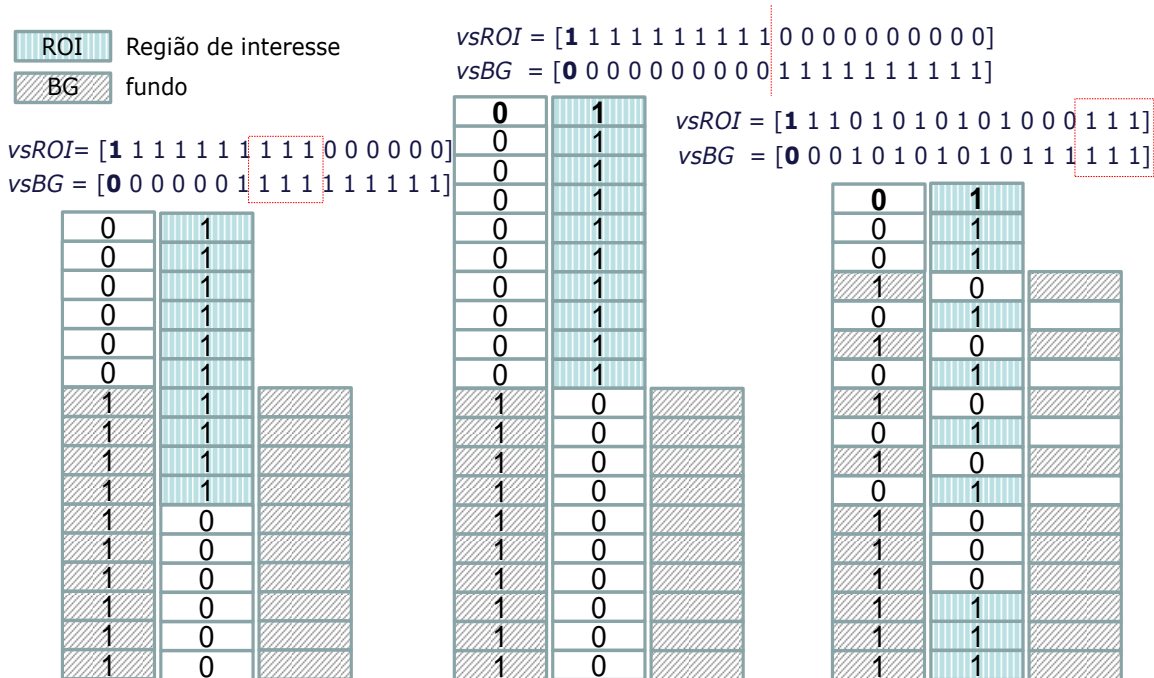


Fig. 4.3. Parâmetro ‘vetor de escalonamento’ da região de interesse: *vsROI*; do fundo: *vsBG*.

Os desenvolvedores (e pesquisadores) têm uma variedade de tipos e seleções para a estratégia de planos de *bits* em certas regiões de interesse.

A forma de codificação se inicia com a definição de uma matriz $mDWT(i, j)$ que contém os coeficientes procedentes de alguma decomposição *wavelet* qualquer. Para cada coeficiente da linha i e coluna j está associada uma quantidade de *bits* b de forma que a matriz pode ser representada na forma $mDWT_b(i, j)$, onde podem-se obter as informações necessárias dos planos de *bits* e da posição.

As codificações da ROI e BG são realizadas utilizando deslocamentos *shift* de acordo com os 1's e 0's do vetor $vsROI$ e $vsBG$ de forma totalmente separada. Os 1's significam que esses planos de *bits* correspondem aos dados de um outro plano de *bits* de uma outra ordem. Já os valores zeros 0's significam que foi feito um *shift* e nessa posição não tem informação (o *codec* entende como 0). Isso para ambos os vetores de escalonamento $vsROI$ e $vsBG$.

Logo, é definido um b apropriado que depende do número de *bits* por coeficiente c utilizado (por exemplo, $b_{max}=15$). O plano de *bits* menos significativo (LSB) é o primeiro plano de *bits* $b=1$ e o plano de *bits* mais significativos (MSB) é $b=b_{max}$.

No exemplo na Fig. 4.3.a, é possível observar que o vetor $vsROI=[1111111111000000]$ e $vsBG=[0000001111111111]$ equivale ao '*scaling-based*' com $s=6$. Na Fig. 4.3.b, equivale ao esquema *maxshift* com $s=10$ e na Fig. 4.3.c equivale a um esquema arbitrário com sobreposição nos últimos planos de *bits* menos significativos $mDWT_{b=1,2,3}$.

Assim, no esquema de compressão dos planos de *bits*, a transmissão é feita por prioridades. No decodificador, os planos de *bits* de algum coeficiente c são escalonados de forma que volte para o coeficiente original sem escalonamento. Essa forma generalizada permite utilizar qualquer variação de codificação ROI e *multi-ROI*. Assim, é necessário utilizar uma máscara para classificar a ROI do BG.

4.5 Geração da máscara da ROI

Para determinar a região de interesse ROI e o BG, é necessário definir os limites dessa região manualmente ou automaticamente. Normalmente, essa região é de forma retangular ou elíptica, usada como padrão no JPEG-2000 [12, 13]. Também pode ser utilizada uma máscara que determina quais coeficientes pertencem à região de interesse ROI e quais pertencem ao fundo BG. Como se observa na Fig. 4.4, em geral, ao aplicar uma codificação de ROI, é necessário uma matriz de coeficientes *wavelet* como entrada de dados. Para delimitar qualquer tipo de forma, é necessário dispor de parâmetros de forma da ROI e gerar uma máscara. Finalmente, deve-se escalonar esses coeficientes com algum método e assim comprimir a imagem.

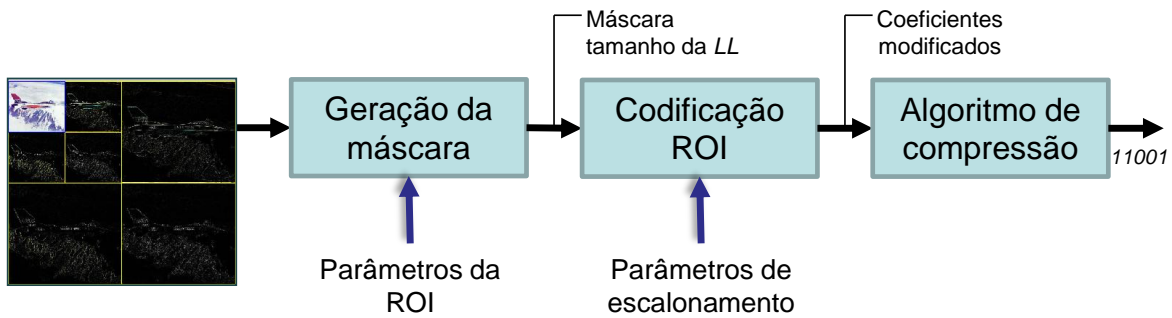


Fig. 4.4. Diagrama de codificação para compressão utilizando ROI a partir de uma TWD.

Existem várias opções de informar ao codificador e decodificador a região de interesse. Entre as mais utilizadas tem-se: enviar coordenadas da máscara (caso seja elíptica ou retangular), enviar a matriz máscara para delimitar as regiões, e na especificação das regras no escalonamento (codificação de escalonamento ROI). Cada uma delas possui vantagens e desvantagens em relação à flexibilidade ou à quantidade de informações enviadas para a decodificação.

Neste capítulo é abordado e dada ênfase à codificação ROI aplicada ao algoritmo SPIHT com múltiplas regiões de interesse (*multi-ROI*). Assim, a forma de representar a região de interesse pode ter várias formas. Como será aplicada para uma codificação ROI generalizada, a forma de enviar a informação da máscara é por meio de uma matriz *roi_mask*, isso sem alterar as regras do algoritmo de compressão (módulo independente).

Na TWD tem-se sempre n níveis de decomposição e o nível mais elevado é a região da *LL*, onde fica a informação visual (resolução menor) para fazer a seleção da região de interesse. Essa informação é suficiente se a relação de tamanho da *LL* (resolução) é aceitável subjetiva ou objetivamente para o usuário. Como se observa na Fig. 4.5 da imagem original o usuário escolhe uma ROI, então essa mesma ROI pode ser determinada aproximadamente na *LL* da TWD no nível superior.

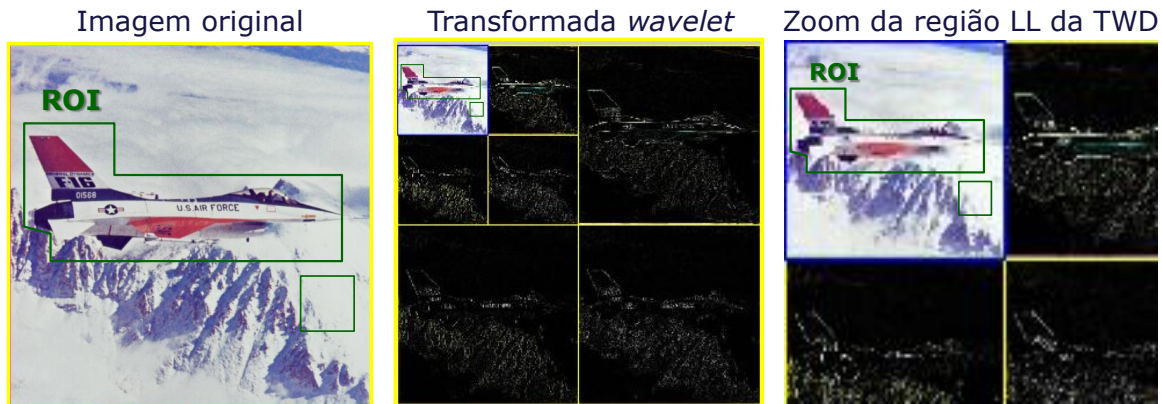


Fig. 4.5. Região de interesse vista na imagem original e na região da *LL* da TWD.

Assim, antes de definir o algoritmo, é necessário saber quais informações devem ser enviadas para a geração do cabeçalho ou o formato de dados a ser transmitido. A informação de *bits* da matriz *mDWT* definida é enviada junto com as informações como: tamanho da matriz *wavelet*, número níveis de decomposição, número de *bits* utilizado para representar cada coeficiente, parâmetros de escalonamento e a matriz da máscara da ROI (máscara proposta *roi_mask* é uma matriz binária de tamanho da *LL* 8 *bytes* para uma TWD de 6 níveis de decomposição).

4.5.1 Algoritmo proposto

O algoritmo é composto dos seguintes passos:

1. Aplica-se a TWD da imagem com *nlevels* níveis e o resultado é armazenado na matriz *mDWT*. Localiza-se a região *LL* da matriz *mDWT* e coloca-se essa parte na matriz *m_LL*.
2. Escolhe-se a ROI direto na região da *m_LL* baseada ou não na ROI da imagem original, ou seja, uma aproximação. Cria-se uma matriz binária *m_ROI_LL* chamada máscara da ROI que corresponde a *LL*. Os valores 1's correspondem ao ROI e os 0's ao fundo BG.
3. Encontra-se a posição de todos os elementos da árvore de decomposição de cada coeficiente *wavelet* que pertence a ROI (ex. filhos, netos), utilizando *m_ROI_LL*. Gera-se uma nova matriz binária *m_ROI* que contém a máscara da ROI total da matriz *mDWT* como se observa na Fig. 4.6.

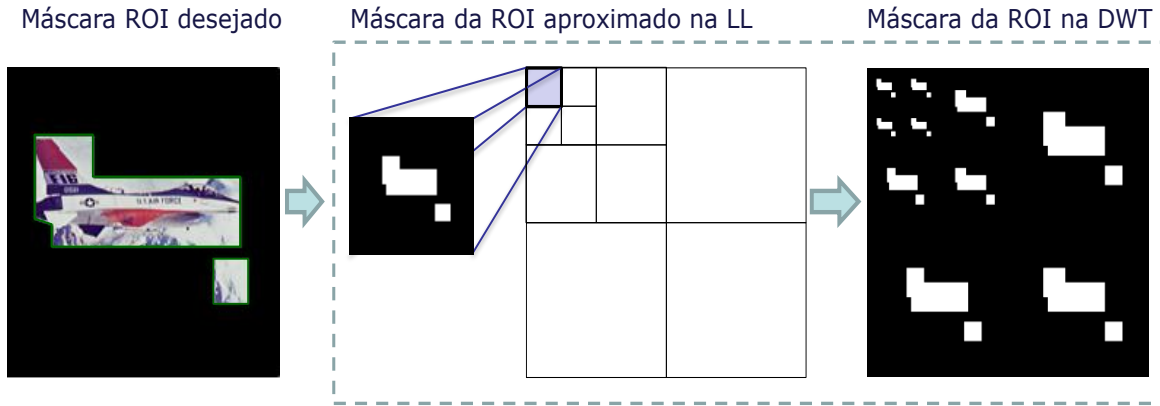


Fig. 4.6. Forma de aplicar a ROI desejada para uma ROI somente na *LL* (aproximada) e depois para outros níveis de decomposição.

Antes de iniciar o algoritmo, o usuário deve escolher a ROI direto na imagem por métodos manuais ou automáticos. No segundo passo podem-se utilizar ferramentas que transformem uma ROI típica escolhida na imagem, em uma equivalente na *LL* (ex. interpolação *bilinear*). Para a implementação proposta, ambos o codificador e o decodificador iniciam somente da máscara m_{ROI_LL} que gera a matriz m_{ROI} . Essa máscara m_{ROI_LL} é a mesma roi_mask .

Uma vantagem de utilizar a codificação ROI utilizando somente a *LL* é a utilização de uma menor quantidade de *bits* para representar a mesma região (menor que fazer uma máscara de toda a imagem). Quanto mais níveis têm a transformada *wavelet* menor será o tamanho da máscara (gerados no codificador/decodificador). Também ao utilizar uma máscara é possível definir regiões de interesse múltiplas (*multi-ROI*) e o esquema apresentado pode ser utilizado em qualquer módulo de compressão ou análise baseado em decomposição *wavelet*.

A principal desvantagem é que se a região (ou regiões) de interesse não for parecida com as formas geométricas (blocos reticulados), então a precisão definida inicialmente diminuirá à medida que aumenta o número de níveis da TWD (aumentam os níveis *wavelet* e a largura da *LL* diminui $(1/2)^n$). Esse compromisso entre precisão e tamanho não é crítico devido ao fato de que a determinação de uma região de interesse ROI não é necessariamente uma segmentação de objetos (onde deve ser escolhido somente o objeto). Assim, encontrar uma região de interesse é ter um objeto ou alvo com uma porção de fundo para saber o contexto em que se encontra a imagem.

4.6 Simulações e resultados

Para as simulações foi escolhido o *software* Matlab. Nesse *software* foi criada a ferramenta para tratamento de *bits* para a codificação ROI sobre um módulo da TWD. Também foi carregado o codificador SPIHT [64]. Ambos foram inseridos no procedimento genérico de simulação desenvolvido como mostrado na Fig. 4.7.

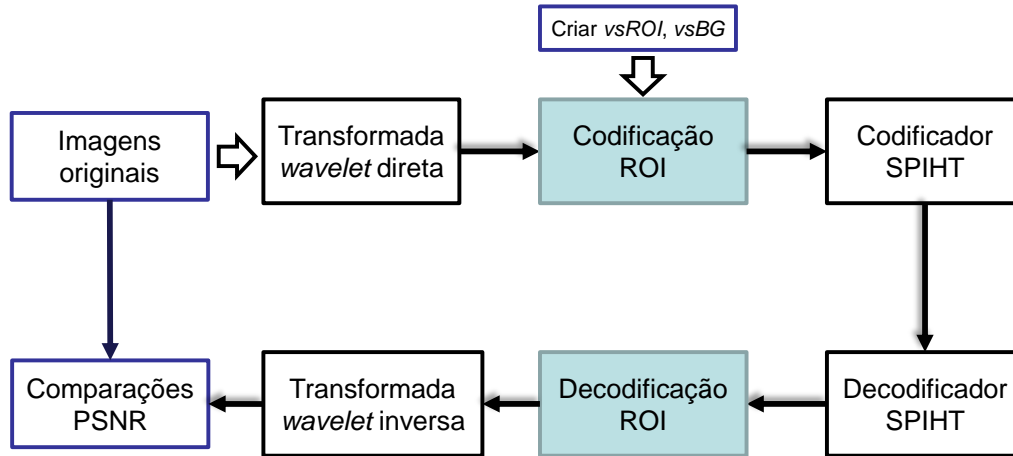


Fig. 4.7. Procedimento teste para executar a compressão de imagens com ROI.

O conjunto de imagens utilizado, 512×512 pixels, para fazer o teste contém: *Airplane*, *Baboon*, *Lenna*, *Barbara*, *Goldhill*, *Peppers*, *Sailboat* e *Satellite*. A TWD utilizada é a biortogonal (bior4.4). Utiliza-se a parte inteira dos coeficientes (15 bits para cada coeficiente). O número de níveis para essa implementação é um importante parâmetro. Assim, o comportamento observado para essa configuração é testado para validar esse parâmetro (ver Fig. 4.8). Escolhem-se utilizar 6 níveis de decomposição.

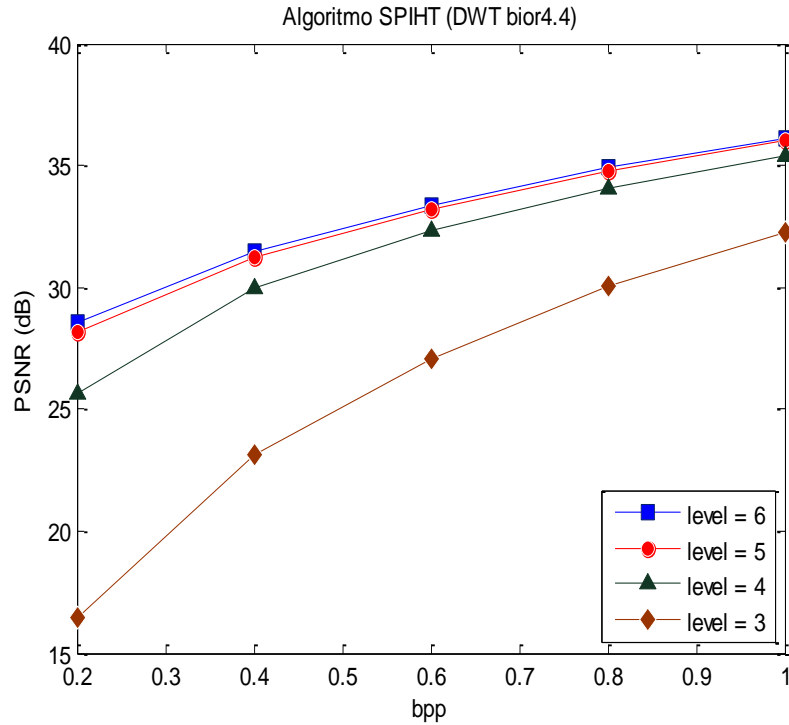


Fig. 4.8. Desempenho do algoritmo SPIHT personalizado na parte da TWD do conjunto de imagens teste (TWD bior4.4, somente parte inteira e com 15 *bits* para representar cada coeficiente).

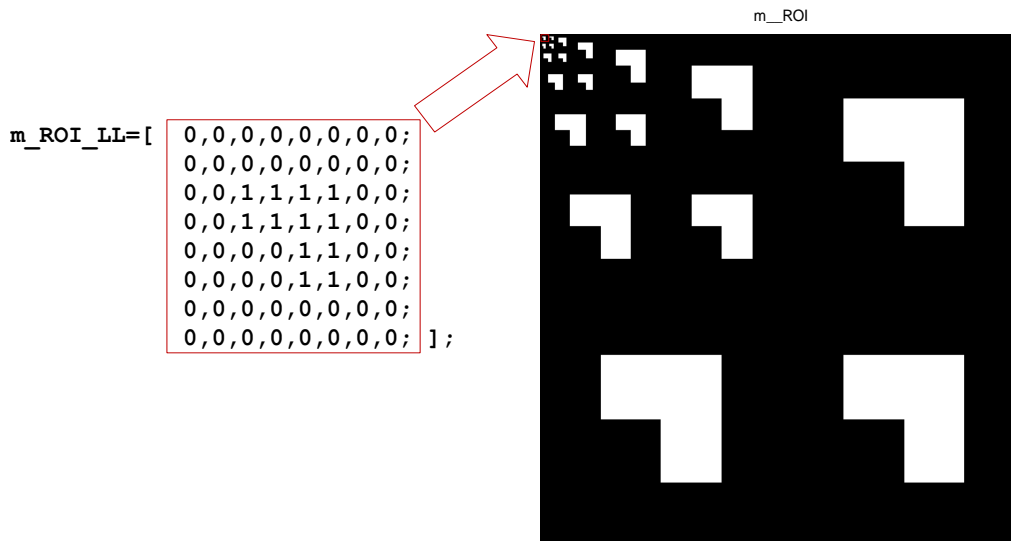


Fig. 4.9. Máscara da ROI na LL m_{ROI_LL} de tamanho 8×8 pixels; máscara gerada de toda a matriz de coeficientes m_{ROI} 512×512 pixels.

A ROI representa o 18,75% da imagem total em um formato (centrado, mas sem comportamento especial) como se observa na Fig. 4.9.

Os parâmetros que são fixos são definidos para que seja possível fazer comparações com a ROI, número de níveis e quantidade de *bits* por coeficiente. Faz-se necessário duplicar a quantidade de *bits* por coeficiente (para evitar um *overflow* se o escalonamento é máximo).

Os vetores *vsROI* e *vsBG* são escolhidos utilizando-se os métodos das pesquisas no estado de arte e adaptados nesses vetores. Métodos como *maxshift*, método híbrido *HBSHift*, método *BbBSHift*, método *GBbBSHift* e métodos com sobreposição (ver Fig. 4.10). Como a configuração desse escalonamento pode ser também variável (métodos híbridos), essas variações podem trazer vantagens ou não (depende de muitos fatores, dentre elas, a imagem original que será decomposta pelas *wavelet* específica). Para cada conjunto de imagens é necessário fazer esse tipo de teste e personalizar a codificação. Para essa personalização, foi construída a ferramenta meta-heurística de convergência rápida descrita no apêndice B que gera um *vsROI* e *vsBG* de forma automática, e dessa forma não ter que testar todas as possibilidades e encontrar uma configuração apropriada.

Devido à quantidade de combinações possíveis mostram-se os resultados desejáveis com PSNR maior do que ~20dB, na maioria de casos.

Para melhor entendimento dos resultados dessa parte, as figuras e os parâmetros que são utilizados obedecem às seguintes notações: a) Na legenda de cada figura tem-se o vetor binário referente a ROI e ao BG; b) O *whole* refere-se ao cálculo do PSNR de toda a imagem; c) Cada título contém também os vetores correspondentes a ROI e ao BG na forma decimal.

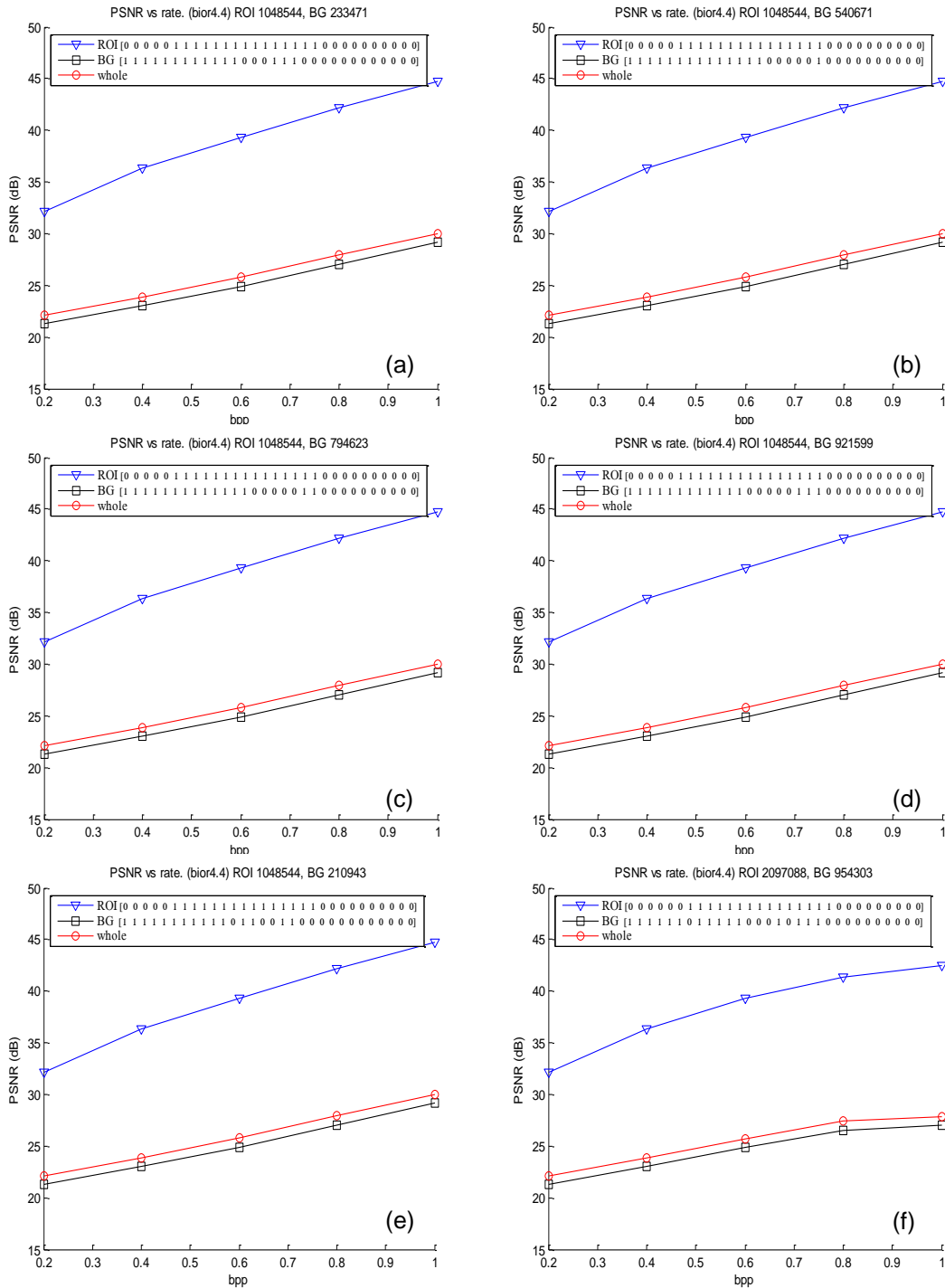


Fig. 4.10. Resultados utilizando configuração híbrida que teve semente nos métodos anteriores à exceção do *scaling-based*.

O método *scaled-based* já possuía pesquisas com o SPIHT, mas com alterações dentro do algoritmo. Nesta implementação, o método é totalmente transparente e se consegue resultados aceitáveis para uma codificação desse tipo (ver Fig. 4.11). O termo subjetivo

aceitável é em relação ao PSNR, onde se considera maior que 25 dB como aceitável. Isso leva em conta a distorção visual da imagem a menos que essa seja muito forte. Na Fig. 4.11.a se observa que a qualidade é aceitável e se tem um bom desempenho em baixas taxas. Para taxas maiores que 0,4 bpp, tem-se um menor desempenho que na Fig. 4.11.b . Na Fig. 4.11.b se observa que a qualidade é aceitável e se tem um bom desempenho em relação às Fig. 4.11.c e Fig. 4.11.d.

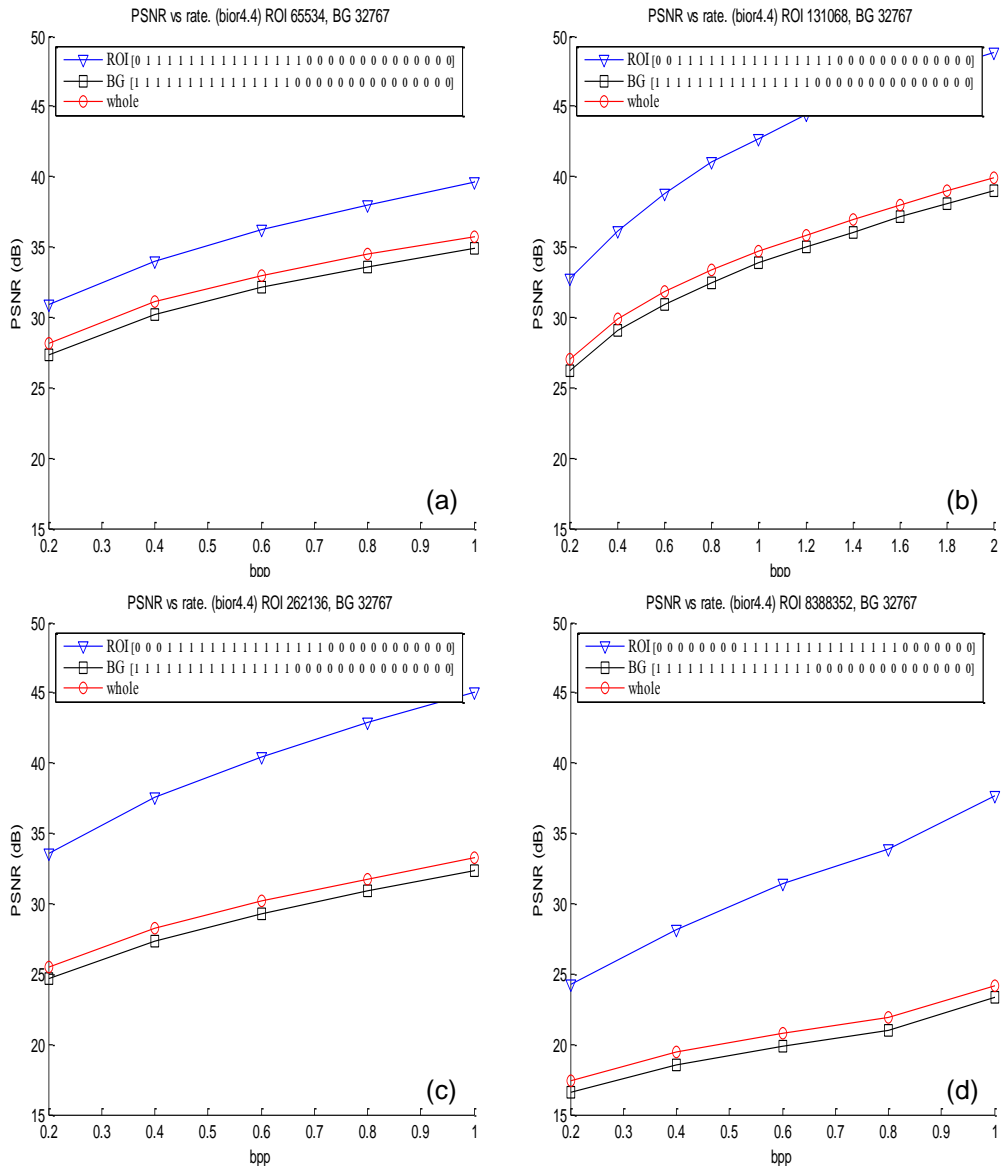


Fig. 4.11. Resultados utilizando configuração equivalente ao *scaling-based*.

Como resultado, tem-se que a maioria dos métodos relatados aumenta muito, o valor dos

coeficientes e diminuem a eficiência ao comprimir (adicionam muitos zeros). A exceção deve-se ao *scaled-based* e métodos híbridos com sobreposição (ver Fig. 4.12 e Fig. 4.13).

Para configurações de métodos híbridos com sobreposição, em alguns casos se obtêm melhores resultados em relação ao PSNR. Isso ocorre usando-se a ROI para uma PSNR maior que 30dB e também próximo da taxa de 2 bpp com uma relação PSNR próxima de 50dB . Essas configurações do *vsROI* e *vsBG* estão baseadas no método *scaled-based* (ver Fig. 4.12) com um mesmo conjunto de imagens testadas. Outras combinações híbridas testadas têm resultados com PSNR menor que 25 dB (ver Fig. 4.14).

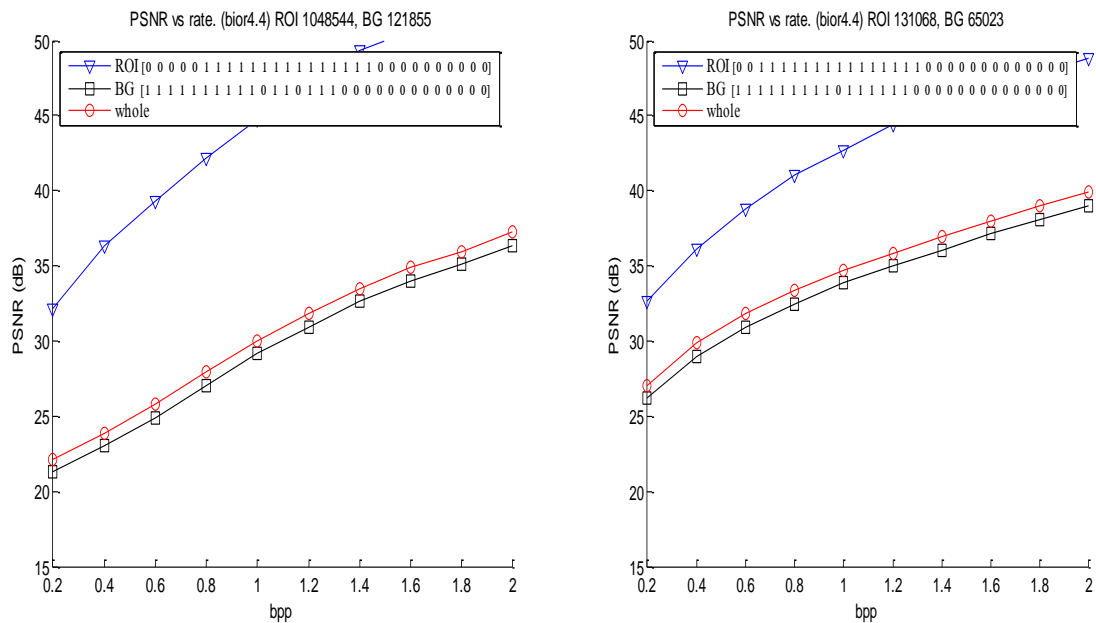


Fig. 4.12. Resultado desejável encontrado utilizando configuração híbrida que teve semente no *scaling-based*.

Uma das características do SPIHT é que ele explora os coeficientes não significantes em árvore *zerostree*. Se algum escalonamento é aplicado e for escolhido como significativo nos passos seguintes do algoritmo (especificamente no passo de refinamento), então os 0's gerados são enviados, sem critério de compressão, o que diminui o desempenho da compressão. Por isso, não é qualquer forma que será viável, pois isso depende da informação dos coeficientes e como se comportam na ROI escolhida.

Na Fig. 4.13 pode-se observar um comportamento diferente. Na Fig. 4.13.a, por exemplo, a curva começa bem, mas o desempenho cai à medida que aumenta a quantidade de *bits*. Na Fig. 4.13.b-f, se observa um comportamento semelhante entre as curvas, que apresentam variações apenas no início das curvas (baixas taxas).

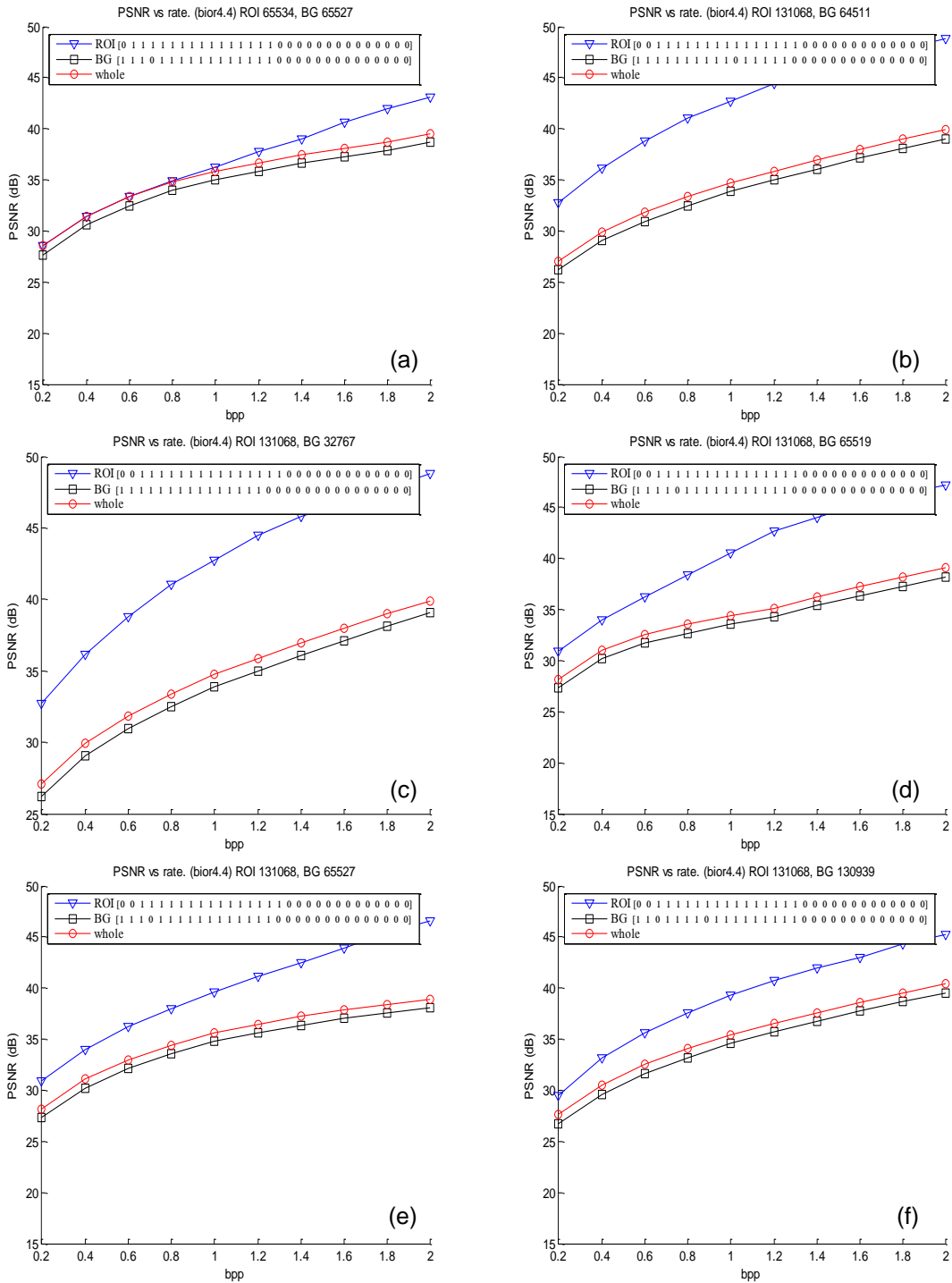


Fig. 4.13. Resultados utilizando configuração híbrida que teve semente no *scaling-based*.

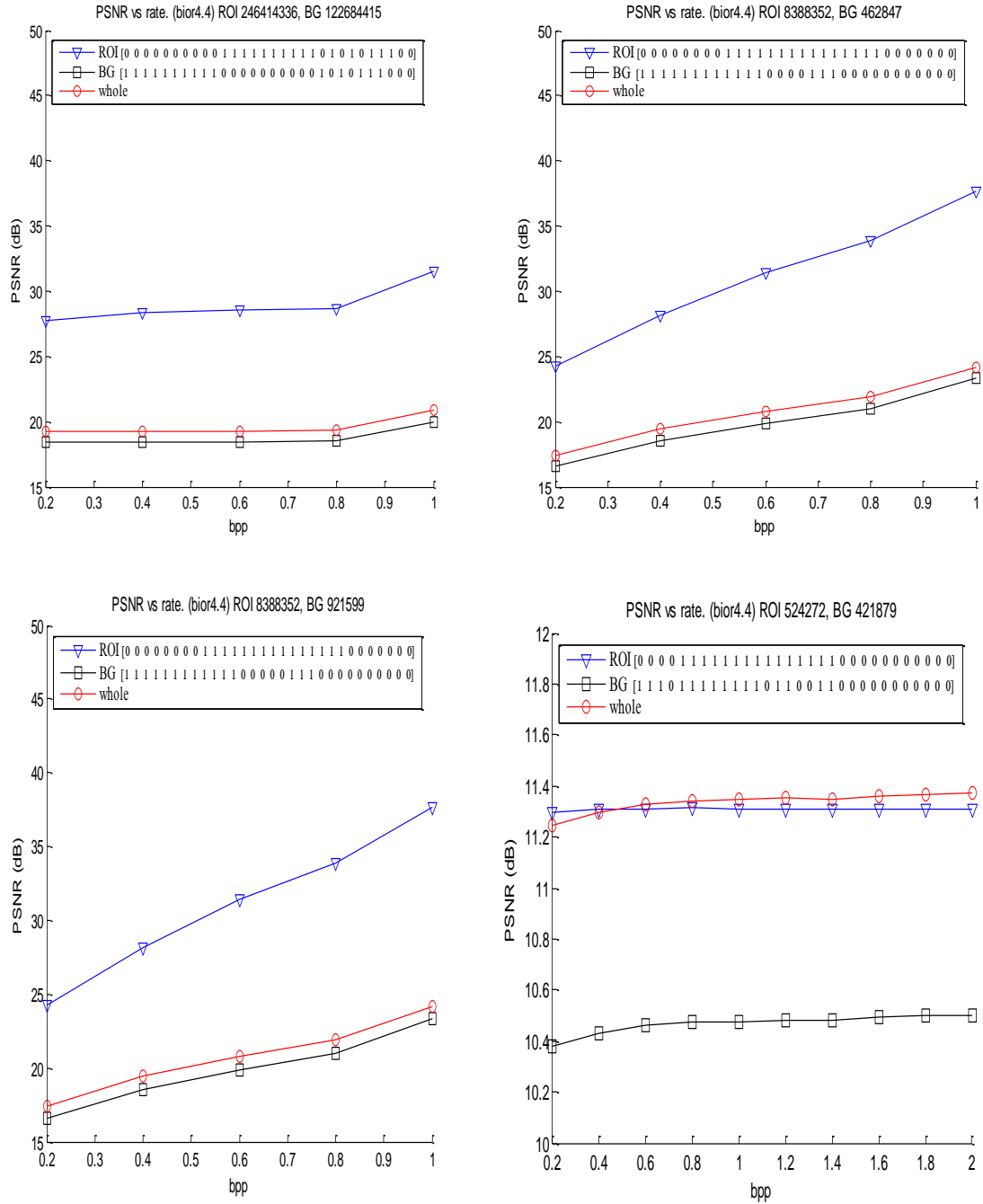


Fig. 4.14. Resultados não desejáveis utilizando configuração híbrida que teve semente nos métodos mencionados à exceção do *scaling-based*.

Capítulo 5

Comparações turbo produto TPC, turbo convolucional TCC e códigos LDPC para aplicações de baixa potência

5.1 Introdução

Em 1948, Shannon provou que um sinal transmitido, submetido a ruídos, possui uma taxa teórica máxima chamada de capacidade de canal [112]. Assim, as técnicas de correção antecipada de erros podem ser aplicadas a fim de atingir o limite de capacidade. O objetivo da codificação de canal, nas últimas décadas, consiste no desenvolvimento de códigos práticos capazes de chegar perto desse limite.

A exploração do espaço (*deep space*), realizada pela Nasa e o JPL (*Jet Propulsion Laboratory*), sempre enfrenta a extrema atenuação do sinal. Assim, pesquisam-se novos códigos perto do limite de Shannon, a fim de que possam ser aplicados comercialmente [113], bem como, padrões eficientes de comunicações por satélite.

Nos anos 70, a melhor opção para comunicações por satélite foi o par código convolucional/ algoritmo Viterbi. A partir disso, códigos convolucionais foram testados em algumas missões espaciais [114, 115]. A partir de uma proposta simples da teoria de Forney, de conectar dois códigos convolucionais [116], um codificador de menor complexidade tem sido aplicado na comunicação via satélite-terra, resolvendo problemas sem perder muita taxa [117].

No campo das comunicações por satélite, a introdução de códigos turbos convolucionais (TCC) em 1993, foi proposta por Berrou, Glavieux e Thitimajshima. Esses autores melhoraram consideravelmente os resultados de taxa de codificação, combinando uma concatenação simples e eficiente em paralelo de códigos convolucionais com complexidade substancialmente menor [118].

As inovações fundamentais foram uma introdução de um entrelaçador entre os dois codificadores convolucionais e uma decodificação iterativa baseada no algoritmo MAP (*Maximum a Posteriori Probability*). Os pesquisadores do JPL analisaram e verificaram propostas sobre codificação turbo e estenderam o conceito de código turbo com dois códigos constituintes para vários códigos [119, 120, 121, 122, 123, 124, 125, 126]. Eles também desenvolveram entrelaçadores de forma mais geral [127], mudaram de concatenação paralela à série [128] e melhoraram os códigos com novos esquemas experimentais [129].

Anos mais tarde, MacKay apresentou um código de matriz de paridade de baixa densidade ou código LDPC (*Low Density Parity Check Code*) com base na teoria original do Gallager [130], que pode ser projetado para funcionar de modo igual ou melhor que os códigos turbos [131]. A redescoberta do LDPC tem promovido pesquisas em todo o mundo e são os códigos com graus de liberdade maiores em comparação com códigos turbos, melhorando o desempenho de correção, diminuindo o patamar de erro (*error floor*) e aumentando a taxa de transmissão, entre outros atributos.

Como resultado dessa flexibilidade, muitos projetos diferentes LDPC foram desenvolvidos, mas sempre apresentando implementações complexas e não práticas. Uma técnica de utilizar o código LDPC baseada em *protographs* [132] tornou-se a versão comercial padrão DVB-S2 (*Digital Video Broadcasting for Satellite*) para decodificadores de alta velocidade no JPL [113] aplicando uma concatenação serial de dois códigos binários lineares: um código BCH externo (Bose-Chaudhuri-Hocquenghem) e um código LDPC interno [113]. Os códigos BCH formam uma classe código de correção de erro parametrizáveis [133] e têm apresentado desempenho favorável concatenando LDPC e BCH. Lee *et al* também mostram que a concatenação entre os códigos LDPC e turbo apresentam um excelente desempenho com baixa complexidade de decodificação em valores baixos de BER, evitando quando possível o patamar de erro de implementações turbos em valores altos de PSNR [134].

O renascimento da LDPC não encerrou com a investigação dos códigos turbos, uma vez que os códigos turbos permanecem como melhor solução para baixas taxas [117]. Essa separação natural significa que os códigos turbos em taxas de 1/6, 1/4, 1/3 e 1/2 [135] poderiam conviver em harmonia com os códigos LDPC em taxas de 1/2, 2/3, 4/5, e 7/8 [136, 137].

Os códigos turbos têm sido utilizados na terceira geração de sistemas de comunicações móveis, os códigos turbos produtos (TPC), também chamados de códigos de bloco turbos (BTC), também foram amplamente pesquisados e aplicados [138, 139].

Enquanto o código LDPC consiste basicamente na concatenação de códigos de blocos, os códigos turbos são feitos geralmente a partir da concatenação de códigos convolucionais.

Em tais implementações, o uso de entrelaçador não é necessário, uma vez que ocorre um processo natural de entrelaçamento entre linhas e colunas.

Após as melhorias de Pyndiah no algoritmo de decodificação SISO, o desempenho dos códigos turbos produtos chega perto dos níveis ideais a baixas taxas de BER e as pesquisas sobre esse assunto foram intensificadas [140, 141, 142, 143]. Mais tarde, Ranking e Gulliver demonstraram a técnica que ficou conhecida como SPC-TPC (*Single Parity-Check*), apresentando um procedimento simples de decodificação [144].

O principal interesse da TPC é a preocupação a respeito da sua distância mínima, que é o produto das distâncias mínimas dos códigos parciais (caso típico). TPC em duas dimensões TPC-2D também tem sido aplicada e pesquisada em *hardware* [145]. Quando comparado com os códigos turbos TCC, o TPC-2D apresenta vantagens em relação aos valores de piso de baixo erro, de baixa complexidade na codificação/decodificação de processos, ausência de entrelaçador e decodificação subóptima capaz de atingir o limite de Shannon a taxas mais elevadas [145]. Como *trade-off*, o TPC-2D apresenta desempenho inferior ao TC em relação ao SNR.

Wu *et al.* têm pesquisado esquemas TPC melhorados pelo aumento da dimensionalidade. Esses autores, por exemplo, propuseram três esquemas de decodificação melhorados que compartilham todas as informações extrínsecas sobre as três direções, em vez de dividir o TPC-3D (3 dimensões) em TPC-2D [140].

Todos esses trabalhos citados foram extremamente importantes e têm apresentado excelentes soluções para sistemas de codificação de canal. No entanto, as comparações entre os desempenhos globais de códigos, que poderia ser elucidativo para a escolha de um código específico para uma *aplicação/taxa* específica de acordo com os requisitos dos sistemas, não foram apresentadas. No tocante à questão de contribuir para essa decisão, neste trabalho se faz comparações entre códigos de correção de erro, com a intenção de apresentar exigências e *trade-offs* de acordo com cada desempenho dos codificadores.

A organização deste capítulo é a seguinte: a seção 5.2 inclui o código turbo baseado em códigos convolucionais e a seção 5.3 envolve os códigos turbos produto com várias dimensões. A seção 4 refere-se ao código LDPC, e na seção 5.5 apresentam-se resultados de testes e comparações envolvendo também os códigos concatenados. Na seção 5.6, são apresentadas as conclusões.

5.2 Código turbo convolucional TCC

O código turbo é constituído por uma concatenação serial ou paralela de codificadores convolucionais TCC onde as distâncias espectrais dos componentes são combinadas para gerar uma distância maior que contém a maior quantidade possível de combinações. Para se

alcançar esse objetivo, é necessário aplicar um entrelaçador capaz de gerar o máximo de decorrelação entre os componentes.

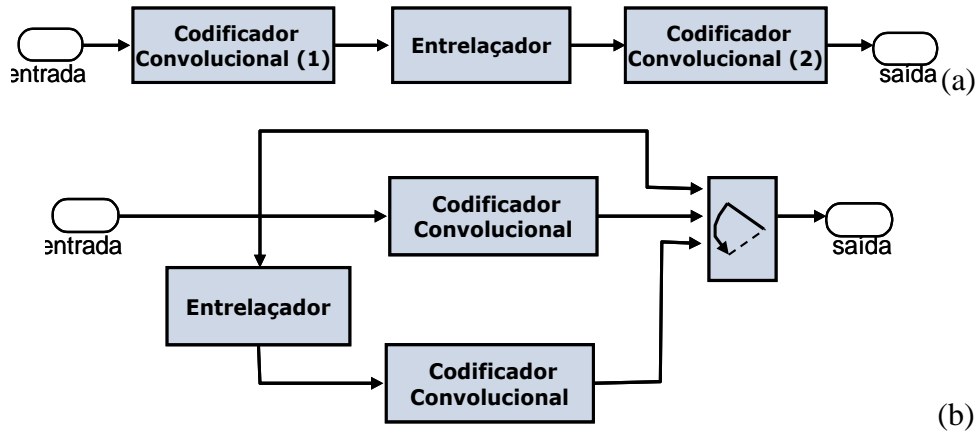


Fig. 5.1. Codificador turbo convolucional a) serie e b) paralelo.

Os códigos paralelos turbos foram introduzidos em 1993 [119]. Nesse esquema, a entrada de um codificador é entrelaçada e a saída é a combinação entre o *bit* sistemático e os *bits* de paridade dos dois codificadores convolucionais. As treliças usadas em ambos os componentes do codificador podem ser iguais (ver Fig. 5.1).

Na implementação de código turbo série, os *bits* de entrada passam pelo primeiro codificador e sua saída é combinada pelos *bits* de paridade e os *bits* sistemáticos como relatado em [117]. Essa informação é correlacionada pelo entrelaçador Π e segue para o segundo codificador que irá gerar a paridade adicional. Ambos os codificadores são necessariamente diferentes.

O processo de codificação de um código turbo genérico com taxa $R = m / n$ pode ser representada graficamente por uma treliça cuja entrada é o vetor $u = u_1, u_2, \dots, u_m$, onde u_t pode tomar os valores -1 ou +1 a priori com uma distribuição de probabilidade $P(u_t)$. Na decodificação, a saída da treliça anterior pode ser representada por um vetor $x = x_1, x_2, \dots, x_n$, onde x_t representa o símbolo produzido pelo codificador no instante t . A razão de verossimilhança logarítmica (*log likelihood ratio* - LLR) é definida pela associação da probabilidade a priori como:

$$L(u_k) = \ln \frac{P(u_n = +1)}{P(u_n = -1)} \quad (5.1)$$

O caminho correto para cada codificador será dado pela probabilidade máxima de verossimilhança da entrada (o algoritmo Viterbi), ligado à probabilidade a priori, definida como a LLR. Assim, se os *bits* são u_t equiprováveis, então essa LRR a priori, é nula.

Ao processo de decodificação cabe fazer a melhor estimativa possível da LLR a

posteriori para todos os *bits* da sequência, o que indicará o valor mais provável da sequência original transmitida u . A realização dessa estimativa pode ser feita por meio de diversos algoritmos presentes na bibliografia, entre os quais o esquema BCJR (*Bahl-Cocke-Jelinek-Raviv*) [146], também conhecido como algoritmo MAP (*Maximum a Posteriori*) e o algoritmo de Viterbi [147, 148]. As expressões a posteriori LLR são dadas pela equação (5.2) e o puncionamento (*puncturing*) é aplicado para a sequência, a fim de alterar a taxa de código de transmissão [149].

$$L(u_k | y) = \ln \frac{P(u_n = +1 | y)}{P(u_n = -1 | y)} \quad (5.2)$$

5.3 Código turbo produto TPC

Pyndiah introduziu o uso de códigos de blocos em esquemas de codificação turbo [30]. O TPC trabalha basicamente com dois codificadores na configuração em série ou paralelo. Nesses esquemas, cada matriz de verificação da paridade pode ser combinada e apresenta menor complexidade que o código turbo convolucional, uma decodificação mais rápida, melhor desempenho em taxas mais elevadas [142, 150], e flexibilidade entre a taxa de código resultante e o comprimento do bloco. Essa flexibilidade pode ser alcançada em função da dimensão do codificador [150, 144].

Considerando-se os k *bits* como uma matriz $k_1 \times k_2$, os *bits* da paridade são adicionados pelos dois codificadores, a fim de compor uma matriz $n_1 \times n_2$. Na codificação serial, o codificador primeiro adiciona *bits* de paridade às linhas, enquanto o segundo codificador adiciona *bits* de paridade às colunas. Além da paridade da mensagem, existem também os *bits* de paridade relacionados com outros *bits* de paridade (paridade das paridades). Assim, existe uma palavra código (etapa 2 na Fig. 5.2) em ambas as orientações.

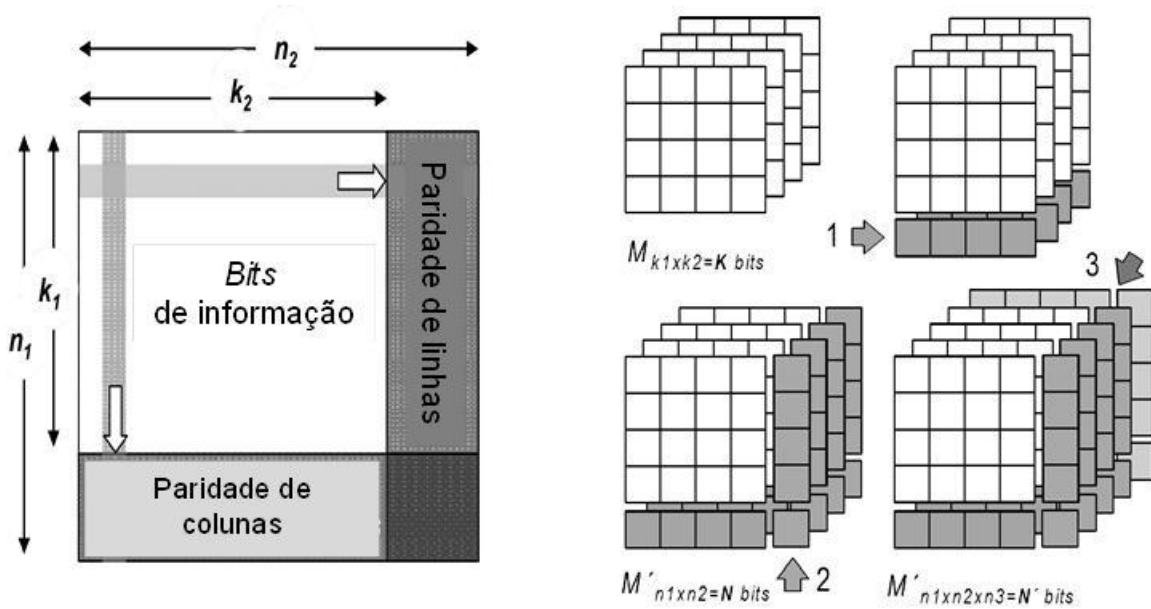


Fig. 5.2. Construção de código turbo produto série em 2D e 3D.

No exemplo da Fig. 5.2 tem-se um $n = 2$ e $n = 3$ e observam-se três etapas para a construção. Primeiro, é adicionada uma paridade a cada coluna, depois é adicionada outra paridade a cada linha (incluindo a linha de paridades gerada na etapa anterior), e terceiro, é gerada uma paridade em cada linha de profundidade.

De acordo com [144], um código produto d-dimensional pode ser construído de forma que cada dimensão q é processada separadamente usando-se a notação (n_q, k_q, h_q) , onde n é o comprimento da palavra-código, k é o comprimento da mensagem e h é a distância mínima de Hamming do código em cada dimensão.

Usando parâmetros de dimensão, é possível calcular a entrada do codificador e tamanhos de saída dos blocos, a taxa de código e a distância mínima. Ranking e Gulliver também apresentaram a SPC-TPC (*Single Parity Check - TPC*) [144] que proporcionam o aumento a uma distância mínima de código a fim de alcançar uma melhor capacidade de correção de erros. Esse melhor desempenho é atingido com dimensões iguais construídas pela concatenação serial de códigos de paridade única. O SPC é geralmente recomendado para sistemas que apresentam várias dimensões, uma vez que é possível reduzir a complexidade e manter um valor mínimo aceitável da distância.

Os sistemas mais robustos TPC aplicam mais de dois códigos de blocos, a fim de organizar os dados como um cubo n-dimensional associando com *bits* de paridade a cada dimensão (incluindo a paridade das paridades). Quanto maiores as dimensões, maior a distância mínima de Hamming, e simultaneamente, diminui a taxa de *bits*. Como consequência, o desempenho do codificador é melhorado. Embora, esse procedimento também produza o crescimento do bloco e, portanto, aumentam os requisitos de memória e

o atraso de codificação. Além disso, o tempo de atraso pode-se tornar um problema significativo em comunicações em tempo real e outros sistemas de restrições de tempo.

Atualmente, a decodificação é realizada separada por entrelaçadores. Os códigos turbos usam algoritmos iterativos de decodificação SISO (*Soft-Input Soft-Output*). Nesses algoritmos, as informações sobre a confiabilidade de um dos componentes de código alimentam o processo de decodificação do outro componente do código como uma entrada suave. Assim, a cada iteração, existe maior confiabilidade na estimação dos dados.

A entrada suave está relacionada com a saída do detector no receptor. Após o sinal transmitido passar pelo canal, é possível determinar a entrada suave do decodificador SISO usando um LLR a posteriori, do símbolo correspondente dada por:

$$L'(\hat{x}_n) = \ln \frac{P(x_n = +\sqrt{E} \mid y_n)}{P(x_n = -\sqrt{E} \mid y_n)} \quad (5.3)$$

Nessa equação tem-se que y_n é a sequência recebida e E é a energia média por *bit* transmitido.

Frequentemente, os algoritmos sub-ótimos (tais como o logMAP) são utilizados, apresentando menor complexidade em relação aos algoritmos ótimos.

5.4 Códigos LDPC (*Low Density Parity Check Codes*)

Os códigos turbos e LDPC compartilham uma similar natureza de seus algoritmos na decodificação. Os códigos LDPC são caracterizados principalmente pela utilização de grafos bipartidos de Tanner e gráficos de fatores. Eles são decodificados com algoritmos genéricos de transmissão de mensagens entre os nós (*nodes*) dos gráficos.

Um código LDPC é um código de bloco binário (n, k, w_c) caracterizado por uma matriz de verificação de paridade H esparsa de dimensão $(n-k) \times n$, isto é, uma matriz em que o número de ‘uns’ ($w_c \ll m = n - k$ por coluna e $w_r = w_c \times n / m$ por linha) é muito pequeno em relação ao número de ‘zeros’ (baixa densidade de ‘uns’). Essa característica implica em um pequeno número de variáveis envolvidas em cada equação de paridade. As linhas de H representam as equações de paridade e as colunas representam os *bits* de palavra de código. O elemento da linha j e coluna i só é 1 se o *bit* x_i fizer parte da j -ésima equação de paridade. Caso, a matriz H apresente um número constante de ‘uns’ por linha ou por coluna, o código é chamado LDPC regular. A matriz de verificação de paridade H de um código LDPC é gerada sujeita às restrições de ‘esparsidade’.

A maneira mais simples para projetar um código LDPC (n, k) envolve a construção da matriz de verificação da paridade H que satisfaça requisitos estabelecidos. Obviamente, para qualquer conjunto de restrições, há um enorme conjunto de códigos que atendem essas especificações. O trabalho de Mackay e Neal estabelece as regras para a construção de códigos LDPC [131].

Tendo uma matriz H é possível criar uma matriz geradora G . A mensagem u é codificada utilizando a expressão $v = u \cdot G$. Assim serão obtidas as palavras código v referentes aos blocos de dados u . Mas, no caso de matrizes enormes isso é inviável na prática. Então são aplicados diversos esquemas como algoritmos RA e IRA (*Regular e Irregular Repeat-Accumulate*) na codificação.

No que diz respeito à decodificação, existem duas abordagens: decodificação abrupta (*hard*) e decodificação suave (*soft*) e é comum para trabalhar com algoritmos iterativos para obter melhor desempenho, como o soma do produto, que basicamente consiste em transferir as mensagens entre os nós em um gráfico de Tanner.

5.5 Simulações e resultados

Nesta seção, são apresentados resultados de simulações que ilustram a comparação de desempenho entre turbo convolucional (TCC), turbo produto (TPC), e códigos LDPC. Os resultados são agrupados em quatro conjuntos, um para cada tipo de código e a final para as comparações entre eles considerando também códigos concatenados e aumento da dimensionalidade. Todos esses experimentos foram realizados em um PC com processador Pentium 4 3.6GHz trabalhando em *clock* e memória RAM de 2GB, e usando Matlab e Simulink 7,01 6,01 (MathWorks®).

5.5.1 Configuração de simulações turbo código convolucional (TCC)

Seleciona-se como representante, o turbo convolucional em série com *frame* tamanho $N = 1024$ usando-se três taxas de código comumente utilizadas ($1/3$, $1/2$, $2/3$). Essa configuração foi escolhida devido à sua relação custo-benefício excelente sobre o uso de memória, complexidade computacional e desempenho de correção de erros apresentados entre o extenso conjunto de simulações. O número máximo de iterações é de aproximadamente seis. Isso por ser um ponto onde se consegue a correção bem sucedida dos erros para a desejada precisão. Os tempos de simulação adquiridos com a ferramenta *Simulink Profile Report* são mostrados na Tabela 5.1.

Tabela 5.1: Resultado de simulação para código turbo convolucional serial.

Rate (R)	Total seg. (s)	Encoder		Decoder		E_b/N_0	E_b/N_0
		(s)	(%)	Inner (%)	Outer (%)	Zero error 4 th iteration	Zero error 6 th iteration
1/3	139	0,105	0,08	86,20	10,90	2,00 dB	1,70 dB
1/2	143	0,290	0,20	86,10	10,80	2,55 dB	2,30 dB
2/3	135	0,140	0,10	86,20	10,70	5,00 dB	4,00 dB

Nos testes do código turbo convolucional foi observado que quanto menor a taxa de código R , melhor é o desempenho da taxa de distorção (dB). O tempo de processamento necessário não varia significativamente entre as taxas porque as taxas de $1/2$ e $2/3$ são formadas por puncionamento da taxa de $1/3$. No entanto, a redução do número de iterações, reduzirá o tempo proporcionalmente para as taxas testadas. Assim, considerando a configuração de melhor compromisso entre consumo de energia e desempenho do código, a escolha é o código com $R = 1/3$ e 4 iterações, porque está livre de erros de decodificação no ruído de $E_b/N_0 = 2$ dB, que é muito próximo ao do código de melhor desempenho de $R = 1/3$ e 6 iterações, gastando menos tempo de processamento (e, portanto, menor consumo de energia). A Fig. 5.3 apresenta as curvas de taxa de erro de *bits* para o TCC.

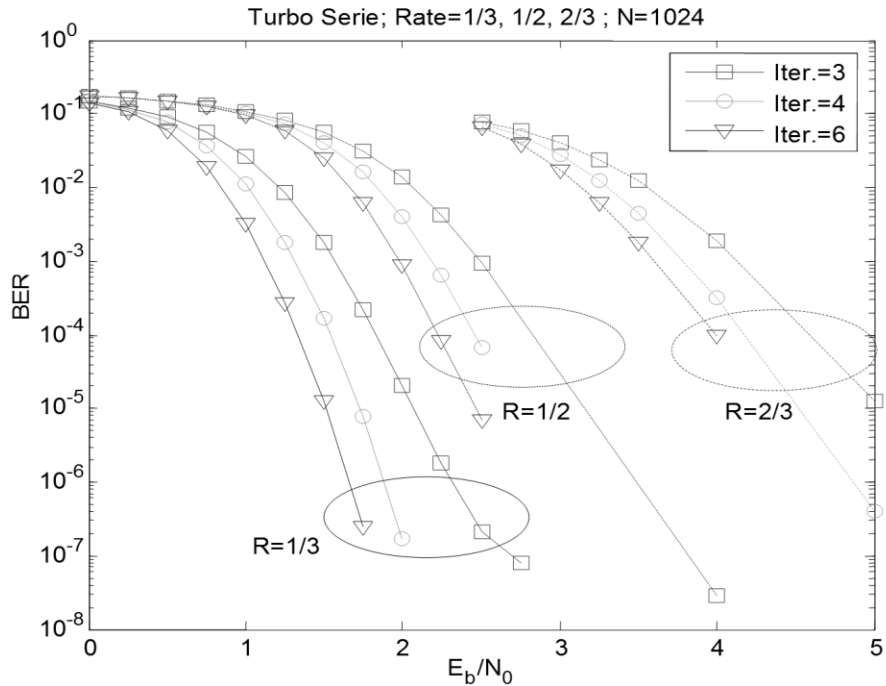


Fig. 5.3. Curvas com número de iterações do código turbo convolucional com *frame* tamanho $N = 1024$ e taxas $R = 1/3$, $R = 1/2$ e $R = 2/3$.

As vantagens dos códigos turbos convolucionais são a simplicidade e o baixo consumo de memória, e sua principal desvantagem é a dificuldade para controlar a convergência (o número pré-estabelecido de iterações é realizado até mesmo sem interferência).

5.5.2 Configuração das simulações código turbo produto (TPC)

Para o TPC, dois subconjuntos de simulações foram realizados em canais AWGN (E_b/N_0 variando de 3-7 dB) e até 15 iterações. Em primeiro lugar, as simulações com os códigos de 4D a taxas código perto de 1/3, 1/2, 2/3 são apresentados, a fim de comparar o ganho de tempo usando algoritmos com controle de iterações.

Um sistema de controle automático de iteração foi implementado, a fim de reduzir o número de iterações de forma independente em cada dimensão. De acordo com Marjan [151], o TPC funciona melhor em taxas mais elevadas. Assim, a Tabela 5.2 mostra resultados apenas para mensurar o impacto do controle de iteração sobre o tempo de convergência e mostrar a variação de tempo de processamento para as taxas testadas.

Os tamanhos dos blocos utilizados foram (3/4) 4, (4/5) 4 e (9/10) 4 para taxas de código 1/3, 1/2 e 2/3 (que se aproximam dessas taxas), respectivamente.

Tabela 5.2: Resultado de simulação para código turbo produto 4D (3-7 dB).

Rate (R)	Total time (s)	Encoder		Decoder		Media de iterações (max=15)
		(s)	(%)	(%)	(%)	
~ 1/3	327	3,98	1,22	1,45	94,83	01,45
~ 1/2	407	1,23	0,30	3,09	96,25	03,09
~ 2/3	1472	0,14	0,01	15,0	88,93	15,00

Nos resultados de simulações, o algoritmo de controle de iteração aumenta significativamente o desempenho do sistema o tempo de processamento em canais de baixo ruído.

O segundo subconjunto de simulações foi feito para comparar o número de dimensões e o impacto sobre o desempenho TPC. A Fig. 5.4 mostra as curvas de taxa de distorção para uma taxa de código de perto de 0,41 para 2D ($k = 22$), 3D ($k = 33$), 4D ($k = 44$) e 5D ($k = 6 \times 54$) códigos de verificação da paridade única.

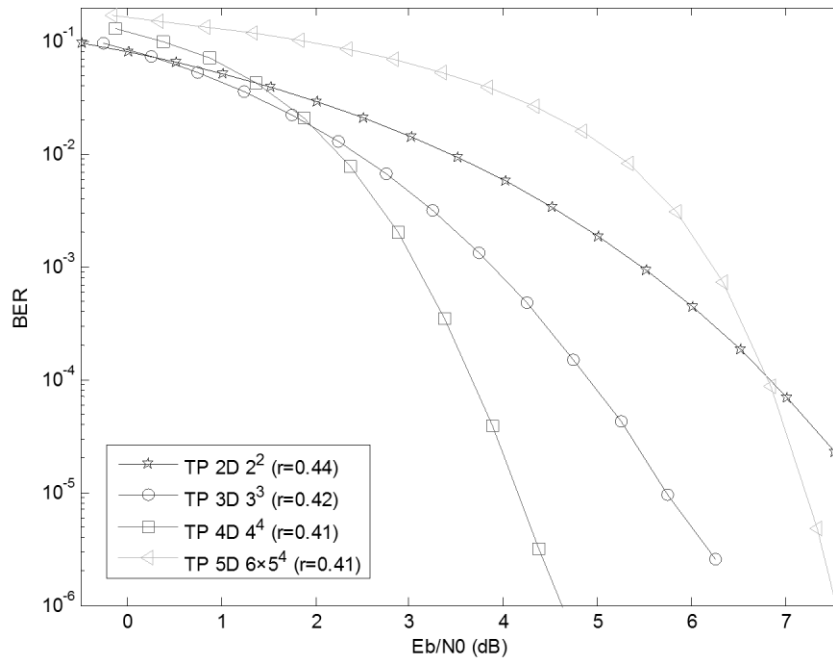


Fig. 5.4. Desempenho do turbo produto 2D, 3D, 4D e 5D para $R \approx 0,41$.

Os resultados mostram que o aumento das dimensões tem um impacto positivo sobre o desempenho até o código 4D usando-se a mesma taxa. Para o código 5D o desempenho é inferior ao do 4D, mas a inclinação da curva é mais acentuada do que a dos outros,

indicando um melhor desempenho para valores mais baixos de BER (abaixo de 10^{-6}).

No entanto, o crescimento das dimensões também aumenta a complexidade do codificador e o uso de memória, bem como o tamanho do bloco, também aumentou, reduzindo a vantagem inerente do código turbo produto. Esse problema também impactou no atraso da codificação.

Portanto, pode-se concluir que a utilização de TPC-4D é recomendado para sistemas que operam nos canais de ruído relativamente baixo (perto de E_b/N_0 , de cerca de 5dB) e tendo menos memória/recursos disponíveis de processamento ou codificação de restrição de atraso.

O tempo de processamento e a complexidade são diretamente dependentes da dimensão do código [151]. No entanto, comparando-se com o códigos turbos convolucional, os sistemas de codificação e decodificação do TPC são muito mais simples, apresentando menores exigências de memória e complexidade computacional.

5.5.3 Configuração das simulações do código LDPC

Nesta seção, os resultados de simulação são apresentados, ilustrando o desempenho do LDPC, concatenando LDPC + BCH e LDPC + códigos convolucionais. Todas as matrizes de verificação da paridade LDPC são as versões descritas no padrão DVB-S2 [113].

A Fig. 5.5 mostra o desempenho da taxa de distorção para o código LDPC com as taxas 1/3, 1/2 e 2/3. Pode-se notar que a redução da taxa de código causa um pequeno aumento no desempenho da taxa de distorção. Claramente, observa-se que o LDPC pode ser aplicado para taxas maiores de 2/3 com um bom desempenho (0,5 dB de diferença).

Também pode ser observado que o desempenho da taxa de distorção aumenta com o número de iterações. No entanto, cada iteração LDPC apresenta uma alta complexidade computacional [152]. Assim, a redução do número de iterações pode diminuir significativamente o custo de codificação computacional.

Outra tentativa de reduzir a complexidade computacional, sem muita perda de desempenho é a utilização de um código concatenado em série, juntamente com o LDPC. O objetivo desse código concatenado é melhorar o desempenho de um decodificador LDPC com limitadas iterações.

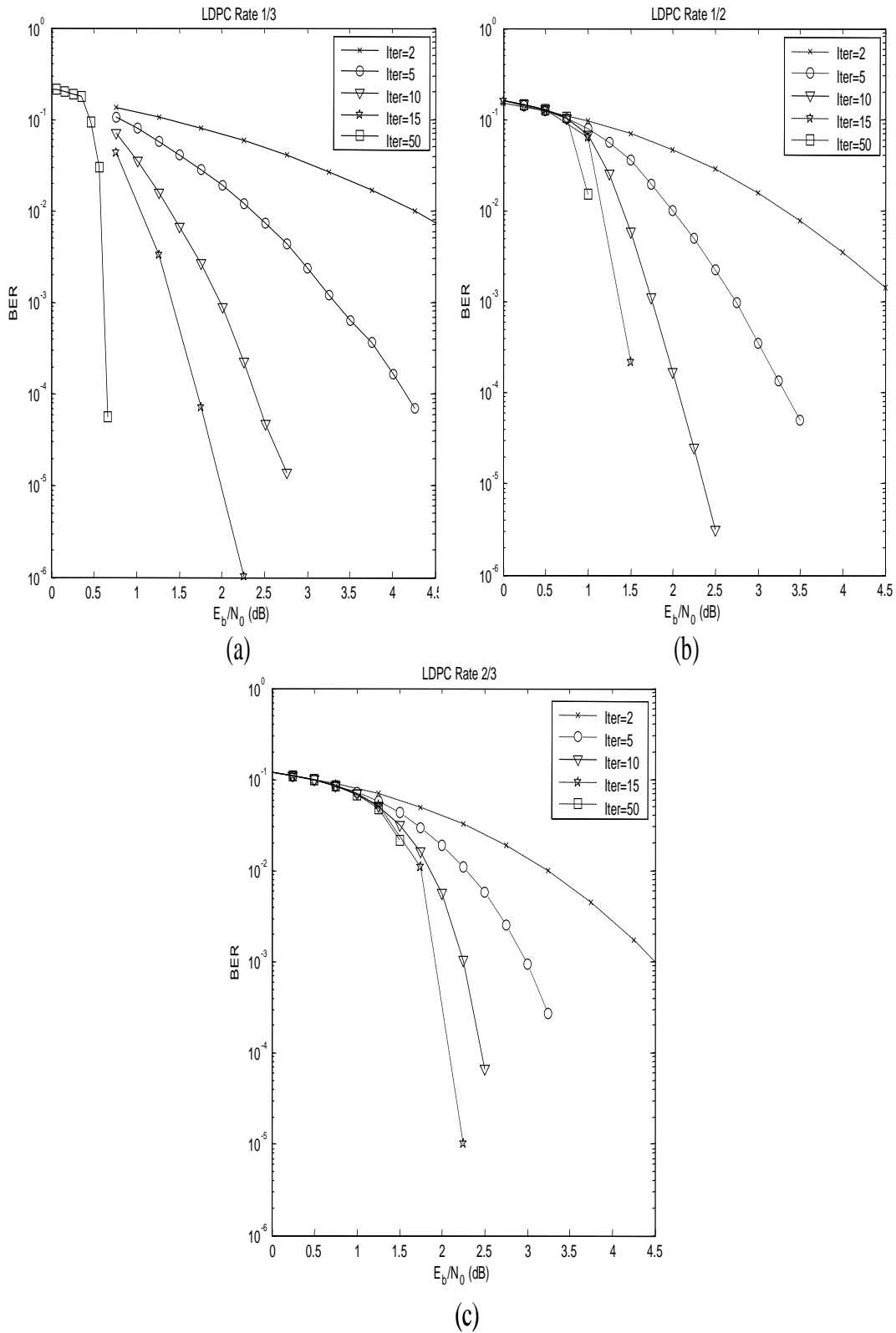


Fig. 5.5. Curvas da taxa de distorção do LDPC. (a) $R = 1 / 3$; (b) $R = 1 / 2$, (c) $R = 2 / 3$.

Para as simulações de código concatenado em série com LDPC, foi necessário manter a taxa de código global em $1/3$, variando a distribuição da taxa entre os LDPC e o código convolucional. As treliças utilizadas no código convolucional são: $(7, (171\ 133))$ para a taxa de $1/2$; $((5\ 4); (23\ 35\ 0, 0\ 5\ 13))$ para a taxa de $2/3$, e $(7, (171\ 133))$ para a taxa de $5/6$. Os resultados das simulações são mostrados na Fig. 5.6.

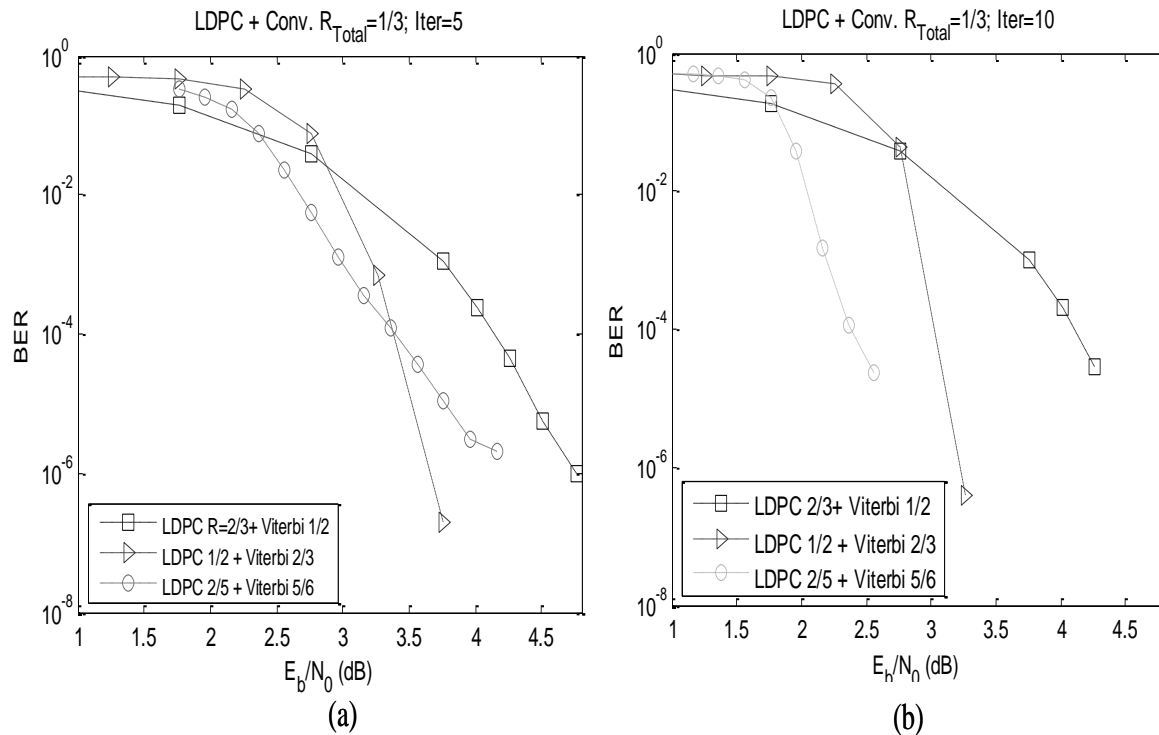


Fig. 5.6. Resultados do código convolucional + LDPC. (a) 5ª iteração, (b) 10ª iteração.

Comparando os resultados, pode-se notar que na 5ª iteração do código concatenado é atingido o nível de BER de 10^{-6} com uma distorção de cerca de 3,7 dB. Para se alcançar o mesmo nível, o código LDPC puro usa mais de 5 iterações (ver Fig. 5.5).

Assim, para manter o desempenho da taxa-distorção desses códigos concatenados, é necessário trocar as iterações do LDPC na decodificação pelo algoritmo Viterbi (para decodificar o código convolucional). Essa troca é muito vantajosa, uma vez que a complexidade computacional do algoritmo Viterbi é muito menor que a decodificação LDPC, mesmo para uma iteração. Essa redução da complexidade computacional leva a uma diminuição no consumo de energia, o que é crítico em sistemas embarcados com energia limitada (como comunicação via satélite). Para a iteração 10ª, e distorção de código concatenado em $BER = 10^{-7}$, os resultados são semelhantes.

Outro teste de código concatenado foi feito usando BCH do padrão DVB-S2. Nesse esquema, a inclusão de código BCH provoca um aumento da taxa mínima de código (taxa

de código é de aproximadamente, ~ 1). A Fig. 5.7 mostra a 5ª, 10ª e 50ª iteração de decodificação para um canal AWGN usando-se taxas de código 1/4, 1/3, 1/2 e 2/3, e utilizando o código LDPC com e sem o BCH concatenado.

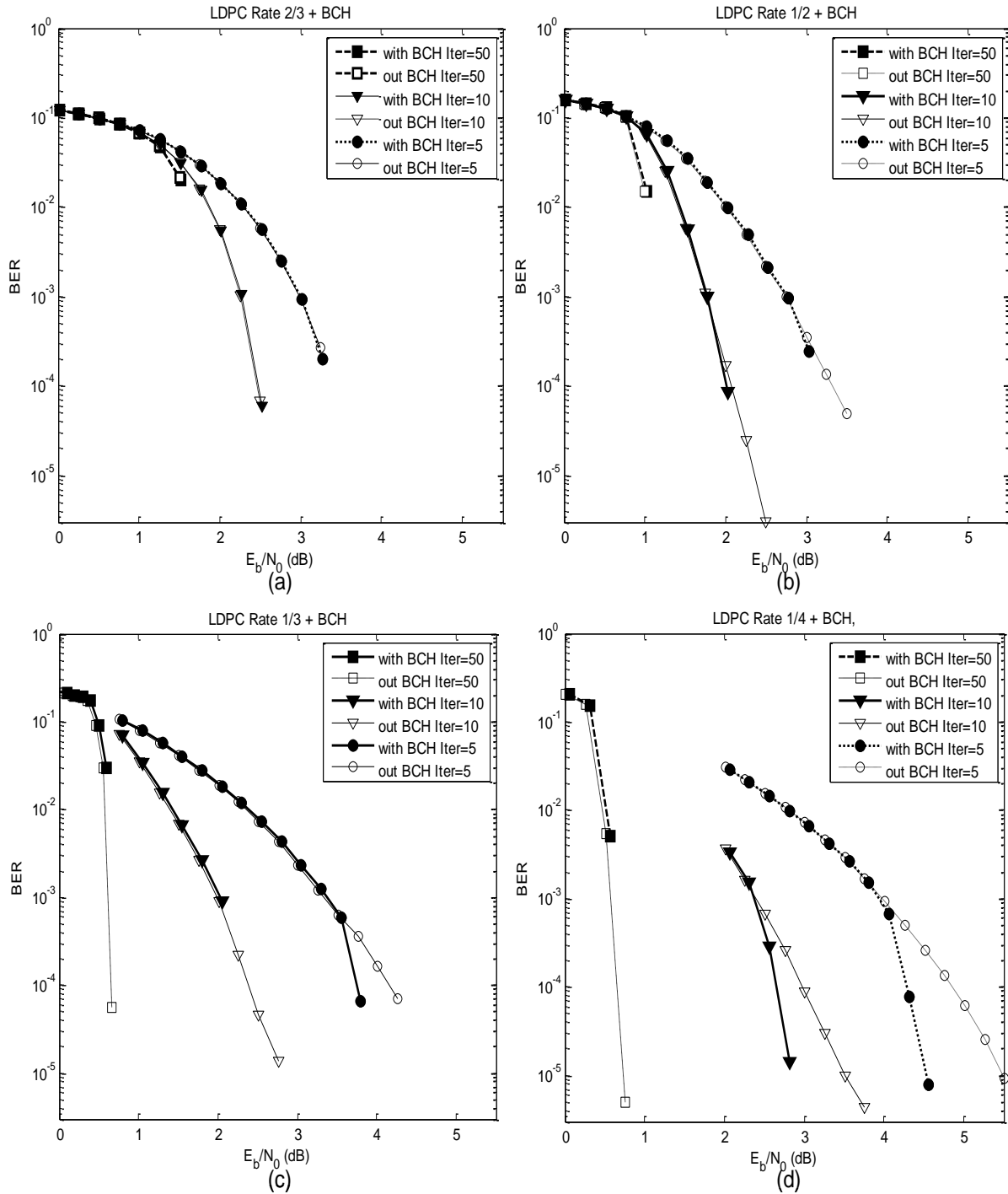


Fig. 5.7. LDPC com e sem BCH comparação a taxas diversas. (a) $R = 2/3$, (b) $R = 1/2$, (c) $R = 1/3$; (d) $R = 1/4$.

Analisando-se as curvas, pode-se observar que a melhora obtida usando-se o código BCH é maior quando se usa uma taxa de código de baixa e poucas iterações. Quando o BER cai o suficiente o código BCH torna-se eficaz e ele começa a contribuir com a convergência LDPC, corrigindo alguns dos erros (restantes).

5.5.4 Comparação entre diversas configurações dos códigos

A Fig. 5.8 apresenta uma comparação entre os LDPC + BCH, LDPC + Viterbi, TCC e TPC. Nessa comparação, foram selecionados os resultados mais relevantes de cada taxa de código no intervalo de 1-5 E_b/N_0 . Em algumas curvas, como TPC, a taxa de código = 2/3, não se encaixa nessa faixa.

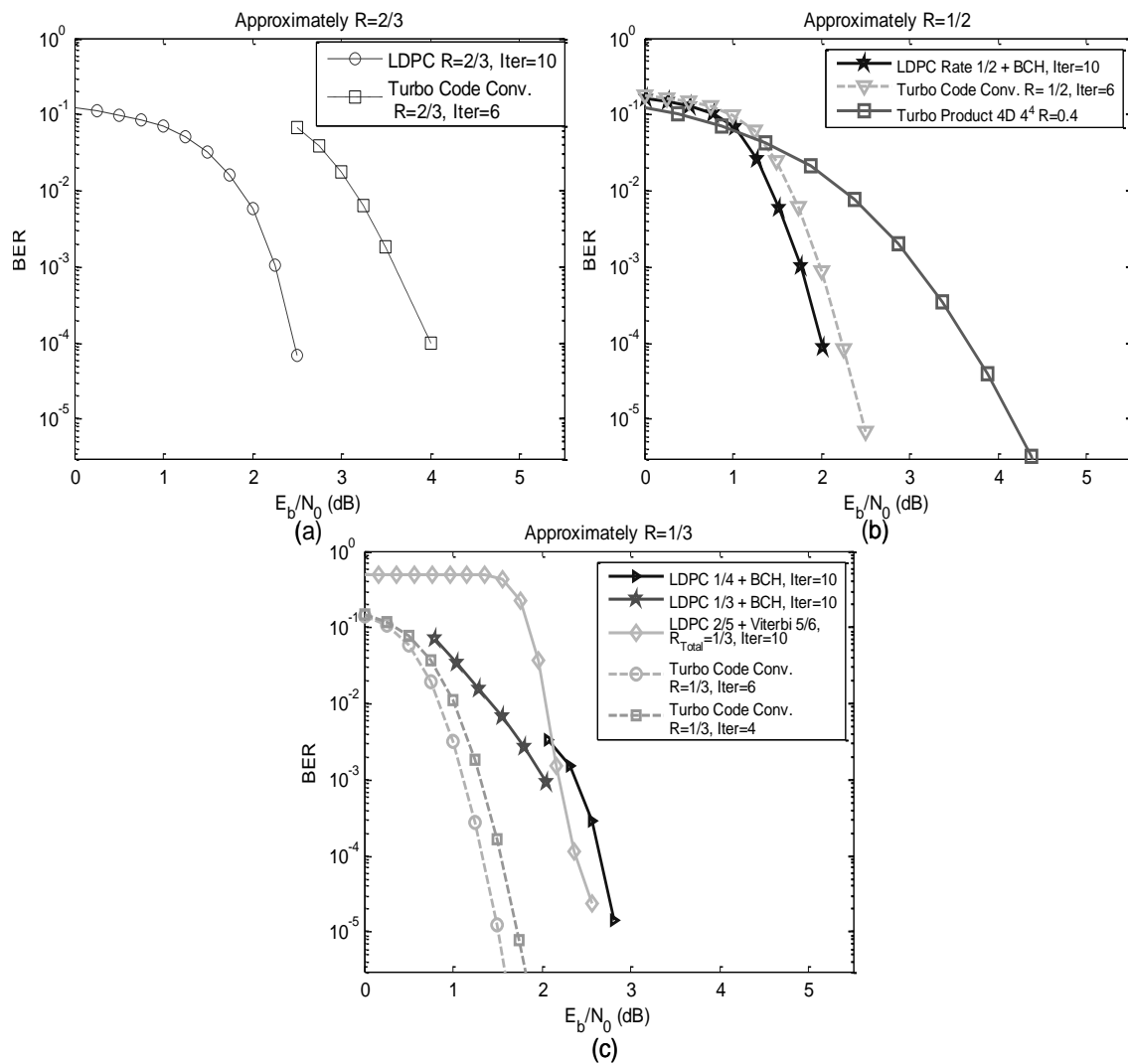


Fig. 5.8. Curvas da taxa de distorção de código de várias taxas. (a) $R = 2/3$, (b) $R = 1/2$, (c) $R = 1/3$.

O desempenho da taxa de distorção mostra que para taxas mais elevadas de código, o código LDPC apresenta os melhores resultados, mas enquanto a taxa de código diminui, o TCC supera o desempenho LDPC. Relativo ao regime de TPC, seus melhores resultados são alcançados perto do 1/2 da taxa de código, e seus desempenhos são mais pobres que os dos outros códigos.

O código concatenado Viterbi + LDPC sempre melhora os desempenhos do puro LDPC ou LDPC + BCH. Em baixas taxas de código, ele se aproxima do turbo convolucional, reduzindo a vantagem de 1 dB.

Em relação às análises da complexidade computacional e uso de memória dos algoritmos de decodificação, é possível se estimar os custos. De acordo com Marjan [151], as complexidades computacionais para os códigos testados são apresentados na Tabela 5.3.

Tabela 5.3: Complexidade computacional de códigos de correção de erros [151].

Tipo de código	Complexidade computacional de decodificação
Convolucional/Viterbi	$O(N2^d)$
TCC	$O(JN(1+2^{d1}+2^{d2}))$
LDPC	$O(JN(W_r+W_c))$

Tem-se: N é o tamanho do *frame*, d é o comprimento de restrição, J é o número máximo de iterações, W_r é o grau da linha e W_c são os graus da coluna (número de *bits* envolvidos em cada paridade).

Considerando-se que ambos os comprimentos de restrição para o TCC são geralmente 3, o comprimento restrição do decodificador Viterbi é 7, o grau médio de LDPC são $W_r \approx 7$ e $W_c \approx 5,5$ [113], e da complexidade TPC está perto do algoritmo Viterbi [153], pode-se estimar que o turbo produto é cerca de 3 vezes mais rápido do que os códigos turbos. Essa velocidade também indica uma redução do consumo de energia. No entanto, o decodificador LDPC, apresenta uma complexidade computacional muito maior do que as dos outros códigos devido ao seu tamanho maior de *frame*. Essa alta complexidade computacional do LDPC também justifica o uso de códigos concatenados como a proposta de um (LDPC + Viterbi) porque se podem regular as iterações LDPC por decodificação Viterbi.

Sobre o uso de memória, todos os códigos precisam de memória com tamanho de um *frame*, que é muito pequeno em turbo convolucional e muito grande no LDPC. O número de dimensões do turbo produto tem um impacto direto sobre o tamanho do *frame* e, portanto, em seus requisitos de memória. No entanto, para os códigos até 4D, esse tamanho de *frame* tende a ser semelhante às do turbo convolucional.

Capítulo 6

Conclusões e trabalhos futuros

6.1 Conclusões

O estudo apresenta importantes conclusões que foram modularizadas em propostas e comparações dentro de cada capítulo. A seguir, as conclusões são apresentadas por capítulo.

Capítulo 2

Os coeficientes da TWD utilizada pelos esquemas estudados podem ser limitados em *bits* com um propósito de melhor utilização de recursos. Essa limitação produz erros já na reconstrução da inversa da TWD. Esses erros na média podem ser mensurados e dependendo da taxa-aplicação, a transformada pode ser aplicada com sucesso.

Capítulo 3

O método de pós-processamento para TWD proposto neste trabalho reduz o número de *bits* necessários para representar cada coeficiente *wavelet*. Foram introduzidos erros de cálculo nos *bits* menos significativos, ou seja, de pouca significância para $N=9$ até 1,0 bpp e 0,8 bpp. Procurou-se a melhor configuração para $N=8$ *bits* que fosse viável para um esquema de compressão com perdas.

Os resultados de simulação mostraram que o desempenho da codificação SPIHT utilizando a TWD, com o pós-processamento proposto, possui PSNR equivalente ao do SPIHT para altas taxas de compressão.

Capítulo 4

A partir dos resultados das simulações realizadas, pôde-se observar que o método proposto não pode ser aplicado diretamente, como no JPEG-2000. O melhor esquema nesses testes para codificar o escalonamento da ROI é o método *scaling-based* com um escalonamento '*s*' curto ($s=2$) e versões híbridas semelhantes a essa. Outras formas também são aceitáveis, no entanto, o conjunto de imagens é muito geral. Se o tipo ou modelo do conteúdo das imagens forem especializados, então os resultados podem melhorar para uma configuração específica.

A ferramenta demonstra que é possível representar muitas formas de codificação segundo os pesquisadores na área.

A geração da máscara é de tamanho apropriado, dado que a região da *LL* é pequena e aceita *multi-ROI*. À medida que a região da *LL* for mais curta (diminui os dados da máscara), a forma de segmentar uma ROI torna-se mais difícil (segmentações são mais imprecisas).

Capítulo 5

Esse capítulo apresentou extensas comparações entre códigos corretores de erro no estado de arte, com a intenção de apresentar exigências mais completas e os *trade-offs* de acordo com cada desempenho. Comparando-se o desempenho da taxa de distorção, complexidade computacional e uso de memória, cada tipo de código apresenta uma vantagem em domínios específicos. O código turbo convolucional, o mais antigo entre eles, é o melhor dos três para ruído gaussiano para a taxa de 1/3, mas tem o problema de não permitir um controle de iteração simples de convergência para os casos de canais de baixo ruído, e isso aumenta o seu custo computacional na decodificação. O turbo produto tem uma baixa complexidade computacional na decodificação e um algoritmo de controle eficiente de iterações, mas tem um desempenho da taxa de distorção menos atraente. O LDPC apresenta o melhor desempenho da taxa de distorção na taxa de código de 2/3 e um controle de iteração eficiente, mas tem uma complexidade computacional e de decodificação de alto uso de memória. Além disso, a complexidade computacional do LDPC pode ser reduzida pelo uso de um código concatenado em série, que reduz o número de iterações. Os resultados aqui apresentados podem ser elucidativos para a escolha de um código específico para determinada aplicação/taxa.

6.2 Sugestões para trabalhos futuros

O estudo aponta questões importantes a serem investigadas no prosseguimento deste trabalho. Apresenta-se a seguir possíveis extensões que podem resultar em novas linhas de pesquisa.

- Juntar as propostas em uma solução global para teste e geração de parâmetros para conjuntos de imagens (desde a imagem original até a reconstruída).
- Aplicar, testar e avaliar o desempenho das propostas em bases de dados de imagens que incluam base de imagens de íris, base de imagens em alta resolução da retina e imagens de segurança.
- Parametrizar e quantificar as variáveis que identifiquem melhor a qualidade do sistema (para um sistema específico). Criar recomendações condensadas para a utilização desses parâmetros para cada caso.

- Medir o comprometimento da capacidade de processamento no codificador e decodificador de cada um dos métodos, seguido de medição do consumo de energia.
- Documentar e propor especificações e recomendações condensadas para as ferramentas criadas neste trabalho.

Capítulo 2

- Aprofundar o estudo do sistema recomendado pela CCSDS orientado á imagens específicas de satélite. Também estudar outras variantes ou novas formas de compressão com algoritmos rápidos.
- Adequar um código em C ou outra linguagem desses algoritmos para se tornar ferramentas para aplicativos embarcados.

Capítulo 3

- O método de pós-processamento proposto nesse trabalho pode ser incorporado em implementações *line-based* [90], ou variantes melhoradas como o *strip-based* [75] e no pós-processamento de [75], isso devido a que a interface está totalmente separada e é de acesso transparente. Especificamente, esse último trabalho pode ser aplicado no passo 6 da proposta apresentada onde se geram zeros nos diversos níveis que podem ser substituídos pela adição de valor $\xi=0,38$ (aproximação) [8, 80].
- Avaliar o *strip-based* como possível forma de diminuir a memória, sem aumentar o processamento e adaptar a proposta.
- O trabalho voltado para codificação para satélite, também pode ser utilizado na decodificação SPIHT com a adaptação de escrita na memória da TWD. A adaptação também deve aceitar o esquema de compressão sem perdas.
- Aplicar os cálculos de *bits* para dimensionar recursos a fim de futuramente desenvolver sistemas aplicados a:
 - Imagens multiespectrais: retina (32 *bits*).
 - Imagens de retina em alta resolução (banco de imagens)
 - Imagens de íris (banco de imagens)
 - Imagens de satélite (banco de imagens)
 - Imagens de vídeo (televisão digital)
- A ferramenta criada é informativa no caso de algum *overflow*, então se pode implementar um tratamento de *overflow* para reduzir os efeitos não desejáveis.
- A proposta pode ser utilizada recursivamente para otimizar o espaço na TWD. Recomenda-se generalizá-la para utilizar mais do que somente o último nível. Nesse esquema, seria necessário um esquema de *software* embarcado reconfigurável.

Capítulo 4

- A proposta pode ser melhorada com um método de escolha iterativa da ROI [154] e é possível se utilizar junto à propostas como [17].

A proposta recomenda utilizar níveis maiores ou iguais ao terceiro nível *wavelet*, mas como observado para um desempenho melhor é necessário mais de 4 níveis de decomposição (recomendado 6). Assim, a relação entre desempenho e facilidade de escolha da aplicação poderá ser definida para cada objetivo específico.

Para contornar o problema de gerar zeros, é possível utilizar depois do SPIHT um codificador aritmético que quando utilizado diminui as sequências de zeros e também atribui à compressão um melhor desempenho.

A informação da máscara pode estar na parte correspondente ao sinal na *LL* (segmento do plano de *bits* do sinal). De fato, uma característica da *LL* é que uma aproximação da imagem em uma TWD de uma imagem positiva, a aproximação (*LL*) deve ser positiva.

Recomenda-se não utilizar menos de 5 níveis para a TWD, como observado na Fig. 4.8, e para cada conjunto de imagens fazer o teste novamente para validar esse parâmetro.

Tanto no passo 2 quanto no passo 3 pode ser utilizada a máscara binária ou de prioridade, se combinado com *scaling-based*.

- Na parte da proposta de aplicar uma codificação ROI para armazenamento com degradação controlada é possível utilizar e degradar não por taxa mas bem por qualidade.
- Os esquemas ROI propostos devem ser aplicados ao JPEG-2000 e outros algoritmos como o SPECK que estão baseados em TWD.
- Implementar uma detecção automática de ROI e aplicar um treinamento para escolher a melhor codificação de região de interesse (aplicado a imagens de retina, íris, TVD e satélite).
- Aplicar a proposta de ROI utilizando outras medidas de qualidade para um objetivo específico na área de reconhecimento.

Capítulo 5

- Começar uma pesquisa conjunta sobre codificação fonte-canal, utilizando algoritmos SPIHT e código turbo convolucional TCC.
- Implementar em *hardware* o codificador/decodificador turbo convolucional TCC e produto TPC.
- Testar a recomendação do CCSDS sobre a codificação turbo flexível [129]. Pode-se ter um codificador com complexidade baixa e utilizando várias configurações.

Referências bibliográficas

- [1] A. Aguiar, “Atualização em foco: Software sem computador,” *Computação Brasil*, Jul-Set 2011. [Online]. Available: www.sbc.org.br
- [2] M. Rokunuzzaman, K. Sekiyama, and T. Fukuda, “Automatic roi detection and evaluation in video sequences based on human interest,” in *Journal of Robotics and Mechatronics*, vol. 22, no. 1, 2010, pp. 65–75. [Online]. Available: <http://www.fujipress.jp/finder/-xslt.php?mode=present&inputfile=ROBOT002200010008.xml>
- [3] P. B. de Souza, T. Nakanishi, and J. B. S. Cunha, “Um ambiente para o desenvolvimento de software embarcado em satélite,” in *Anais do I WORCAP*, 2001. [Online]. Available: http://mtc-m18.sid.inpe.br/col/lac.inpe.br/worcap/2004/05.10.18.49/doc/artigo%2021%20pp%2063-65%20M%20Primavera_IworCAP.pdf
- [4] Instituto Tecnológico de Aeronáutica, Instituto Nacional de Pesquisas Espaciais, Agência Espacial Brasileira, “Programa de Satélites Universitários ITASAT,” acesso: Nov. 2009. [Online]. Available: <http://www.itasat.ita.br/>
- [5] P. Lamsrichan, “A fast algorithm for low-memory embedded wavelet-based image coding without list,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*, may 2011, pp. 979–982.
- [6] Y. Sun, H. Zhang, and G. Hu, “Real-time implementation of a new low-memory spiht image coding algorithm using dsp chip,” *Image Processing, IEEE Transactions on*, vol. 11, no. 9, pp. 1112–1116, sep 2002.
- [7] C. Chrysafis and A. Ortega, “Line based reduced memory, wavelet image compression,” in *Data Compression Conference, 1998. DCC '98. Proceedings*, mar-1 apr 1998, pp. 398–407.
- [8] L. W. Chew, L.-M. Ang, and K. P. Seng, “Reduced memory spiht coding using wavelet transform with post-processing,” in *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on*, vol. 1, aug. 2009, pp. 371–374.
- [9] J. Shapiro, “An embedded hierarchical image coder using zerotrees of wavelet coefficients,” in *Data Compression Conference, 1993. DCC '93.*, 1993, pp. 214–223.
- [10] W. A. Pearlman and A. Said, “Spiht image compression with set partitioning in hierarchical trees,” web, Oct. 2004, last updated: 15 Oct. 2004. [Online]. Available: <http://www.cipr.rpi.edu/research/SPIHT/>
- [11] A. Said and W. Pearlman, “Image compression using the spatial-orientation tree,” in

- Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on,* may 1993, pp. 279 –282 vol.1.
- [12] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The jpeg 2000 still image compression standard,” *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 36 – 58, sep 2001.
- [13] D. S. Taubman and M. W. Marcellin, *JPEG2000 : Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Boston, 2002.
- [14] S. Jenisch, S. Lukesch, and A. Uhl, “Comparison of compression algorithms’ impact on iris recognition accuracy,” *Proceedings of SPIE*, vol. 6819, no. May 2011, pp. 68190M–68190M–9, 2008. [Online]. Available: <http://link.aip.org/link/PSISDG/v6819/i1/p68190M/s1&Agg=doi>
- [15] Y. Jin, Y. Wang, Q. Ruan, and X. Wang, “A new scheme for 3d face recognition based on 2d gabor wavelet transform plus lbp,” in *Computer Science Education (ICCSE), 2011 6th International Conference on*, aug. 2011, pp. 860 –865.
- [16] A. Mascher-Kampfer, H. Stogner, and A. Uhl, “Comparison of compression algorithms’ impact on fingerprint and face recognition accuracy,” C. W. Chen, D. Schonfeld, and J. Luo, Eds., vol. 6508, no. 1. SPIE, 2007, p. 650810. [Online]. Available: <http://link.aip.org/link/?PSI/6508/650810/1>
- [17] R. Guo and J. lin Liu, “Image communications based on adaptive segmentation and wpc-ldpc over wireless channels,” in *Information Acquisition, 2006 IEEE International Conference on*, aug. 2006, pp. 1406 –1410.
- [18] V. Velisavljevic, B. Beferull-Lozano, and M. Vetterli, “Space-frequency quantization for image compression with directionlets,” *Image Processing, IEEE Transactions on*, vol. 16, no. 7, pp. 1761 –1773, july 2007.
- [19] J. Rombaut, A. Pizurica, and W. Philips, “Low complexity error concealment of wavelet coded images in lossy packet networks,” in *Visual Media Production, 2005. CVMP 2005. The 2nd IEE European Conference on*, nov. - 1 dec. 2005, pp. 78 – 85.
- [20] T. Li, Q. Li, S. Zhu, and M. Ogihara, “A survey on wavelet applications in data mining,” *SIGKDD Explor. Newsl.*, vol. 4, pp. 49–68, December 2002. [Online]. Available: <http://doi.acm.org/10.1145/772862.772870>
- [21] B. J. Woodford, D. Deng, and G. L. Benwell, “A wavelet-based neuro-fuzzy system for data mining small image sets,” in *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32*, ser. ACSW Frontiers '04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 139–143. [Online]. Available: <http://dl.acm.org/citation.cfm?id=976440.976461>

- [22] C.-T. Lu and L. R. Liang, "Wavelet fuzzy classification for detecting and tracking region outliers in meteorological data," in *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, ser. GIS '04. New York, NY, USA: ACM, 2004, pp. 258–265. [Online]. Available: <http://doi.acm.org/10.1145/1032222.1032260>
- [23] R. S. S. Kumari and V. Sadasivam, "A novel algorithm for wavelet based ecg signal coding," *Comput. Electr. Eng.*, vol. 33, pp. 186–194, May 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1232939.1232969>
- [24] X. Li, J. Li, and X. Yao, "A wavelet-based data pre-processing analysis approach in mass spectrometry," *Comput. Biol. Med.*, vol. 37, pp. 509–516, April 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1224799.1224882>
- [25] C. C. Aggarwal and P. S. Bradley, "On the use of wavelet decomposition for string classification," *Data Min. Knowl. Discov.*, vol. 10, pp. 117–139, March 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1061897.1061907>
- [26] G. Bernabé, R. Fernández, J. M. Garcá, M. E. Acacio, and J. González, "An efficient implementation of a 3d wavelet transform based encoder on hyper-threading technology," *Parallel Comput.*, vol. 33, pp. 54–72, February 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1227861.1227958>
- [27] W. B. Franceschi, "Procedimentos e Práticas para Digitalização de Imagens Médicas," Master's thesis, Universidade Federal de Paraná, Parana, Brasil, Dez. 2006.
- [28] U. Bick and H. Lenzen, "Pacs: the silent revolution." *European Radiology*, vol. 9, no. 6, pp. 1152–1160, 1999. [Online]. Available: <http://www.springerlink.com/index/66C4J57QV0RHYPYL.pdf>
- [29] R. Behroozmand and F. Almasganj, "Optimal selection of wavelet-packet-based features using genetic algorithm in pathological assessment of patients' speech signal with unilateral vocal fold paralysis," *Comput. Biol. Med.*, vol. 37, pp. 474–485, April 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1224799.1224885>
- [30] L. Dettori and L. Semler, "A comparison of wavelet, ridgelet, and curvelet-based texture classification algorithms in computed tomography," *Comput. Biol. Med.*, vol. 37, pp. 486–498, April 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1224799.1224886>
- [31] A. Akhbardeh, S. Junnila, M. Koivuluoma, T. Koivistoinen, V. Turjanmaa, T. Koobi, and A. Varri, "Towards a heart disease diagnosing system based on force sensitive chair's measurement, biorthogonal wavelets and neural networks," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 493–502, 2007. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S095219760600128X>

- [32] J. Ng and U. of Cambridge. Engineering Dept, *Wavelet restoration of medical pulse-echo ultrasound images in an em framework*, ser. CUED/F-INFENG/TR. University of Cambridge, Dept. of Engineering, 2006. [Online]. Available: <http://books.google.com.br/books?id=nK68HAAACAAJ>
- [33] C. Zanchettin and T. B. Ludermir, “Wavelet filter for noise reduction and signal compression in an artificial nose,” *Appl. Soft Comput.*, vol. 7, pp. 246–256, January 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1221581.1221766>
- [34] N. Bourbakis, P. Yuan, and S. Makrogiannis, “Object recognition using wavelets, 1-g graphs and synthesis of regions,” *Pattern Recogn.*, vol. 40, pp. 2077–2096, July 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1234417.1234627>
- [35] C. Bhattacharya and P. Mahapatra, “A discrete wavelet transform approach to multiresolution complex sar image generation,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 4, no. 3, pp. 416–420, July 2007.
- [36] L. Ghouti, A. Bouridane, M. Ibrahim, and S. Boussakta, “Digital image watermarking using balanced multiwavelets,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 4, pp. 1519–1536, April 2006.
- [37] C. Xu, N. C. Maddage, X. Shao, and Q. Tian, “Content-adaptive digital music watermarking based on music structure analysis,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, February 2007. [Online]. Available: <http://doi.acm.org/10.1145/1198302.1198303>
- [38] N. Zhang and M. Kezunovic, “Transmission line boundary protection using wavelet transform and neural network,” *Power Delivery, IEEE Transactions on*, vol. 22, no. 2, pp. 859–869, April 2007.
- [39] R. F. Larico, Y. Iano, O. Saotome, and R. S. Higa, “A Coding Scheme for Storage Surveillance in Timeline using SPIHT and ROI Coding,” May. 2011. [Online]. Available: http://www.inatel.br/iwt/component/docman/doc_download/34-proceedings-iwt-2011
- [40] A. Hore´ and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug. 2010, pp. 2366–2369.
- [41] R. Arthur, “Novas propostas para otimização de receptores de TV digital baseados em OFDM em ambientes de redes de frequência única regionais,” Ph.D. dissertation, Engenharia Elétrica e Computação, Dept. Comunicações, Universidade Estadual de Campinas, Campinas, São Paulo, Brasil, Feb. 2007. [Online]. Available: <http://www.bibliotecadigital.unicamp.br/document/?code=vtls000410841>
- [42] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

- [43] Y. Meyer and R. Ryan, *Wavelets: algorithms & applications*, ser. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1993, no. v. 1. [Online]. Available: <http://books.google.com.br/books?id=0vpQAAAAMAAJ>
- [44] A. Cohen, I. Daubechies, and J.-C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485–560, 1992. [Online]. Available: <http://dx.doi.org/10.1002/cpa.3160450502>
- [45] D. Alani, A. Averbuch, and S. Dekel, “Image coding with geometric wavelets,” *Image Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 69–77, jan. 2007.
- [46] L. Oliveira, A. L. Cruz, F. Silva, Y. Iano, R. Chavez, and V. Sablón, “Uma visão sobre codificação wavelet de imagens,” *Revista Ciência e Tecnologia*, vol. 11, no. 18, 2010. [Online]. Available: <http://revistavirtual.unisal.br:81/seer/ojs-2.2.3/index.php/123/article/view/82>
- [47] V. Sablón, L. Mendez, and Y. Iano, “A transformada wavelet no processamento e compressão de imagens,” *Revista Ciência e Tecnologia*, vol. 6, no. 9, 2003. [Online]. Available: <http://revistavirtual.unisal.br:81/seer/ojs-2.2.3/index.php/123/article/view/33>
- [48] S. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674–693, jul 1989.
- [49] A. Poularikas, *Transforms and Applications Handbook*, ser. Electrical engineering handbook series. CRC Press, 2009. [Online]. Available: <http://books.google.com.br/books?id=LfTNdPDLOkYC>
- [50] Y. Sheng, *Transforms and Applications Handbook*, ser. Electrical engineering handbook series. CRC Press, 2009, ch. Ch. 10 Wavelet Transform. [Online]. Available: <http://books.google.com.br/books?id=LfTNdPDLOkYC>
- [51] CCSDS, “Image data compression,” PDF, Consultative Committee for Space Data Systems, Office of Space Communication, NASA, Washington, DC 20546, USA, Green Book, 123.0-G-1 123, June 2007, informational report. [Online]. Available: <http://public.ccsds.org/publications/archive/120x1g1e1.pdf>
- [52] R. Rice and P. Yeh, *Algorithms for a very high speed universal noiseless coding module*. NASA, Jet Propulsion Laboratory, California Institute of Technology, 1991, no. 91-1. [Online]. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19910010420_1991010420.pdf
- [53] P. Yeh, *The CCSDS Lossless Data Compression Algorithm for Space Applications*. Goddard Space Flight Center; Jet Propulsion Laboratory, 2001, no. E-12720; NAS 1.26:210762; NASA/CR-2001-210762. [Online]. Available: <http://www.archive.org/>

details/nasa_techdoc_20010073033

- [54] P.-S. Yeh, R. F. Rice, and W. H. Miller, “On the optimality of a universal noiseless coder,” no. AIAA PAPER 93-4540, 1993.
- [55] GICI group, *TER development manual ver1.1*, Department of Information and Communications Engineering Universitat Autònoma Barcelona, Jan. 2007, gICI group. [Online]. Available: http://gici.uab.es/TER/TER-1_1/TERDevManual.pdf
- [56] —, *TER user manual ver1.1*, Department of Information and Communications Engineering Universitat Autònoma Barcelona, Jan. 2007, gICI group. [Online]. Available: http://gici.uab.es/TER/TER-1_1/TERDevManual.pdf
- [57] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, “An overview of jpeg-2000,” in *Data Compression Conference, 2000. Proceedings. DCC 2000*, 2000, pp. 523–541.
- [58] “Intel® integrated performance primitives (intel® ipp) 6.0,” PDF, 2009. [Online]. Available: <http://software.intel.com/file/15405>
- [59] M. Azkarate-askasua, “JPEG2000 image compression in Multi-Processor System on Chip,” Master’s thesis, Delft University of Technology, The Netherlands, Jul. 2008, internal Technical Report CE-TR-2008-05. [Online]. Available: http://www.liacs.nl/~stefanov/pdf/Azkarate_MSc08.pdf
- [60] H.-C. Fang, Y.-W. Chang, C.-C. Cheng, C.-C. Chen, and L.-G. Chen, “Memory efficient jpeg2000 architecture with stripe pipeline scheme,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 5, march 2005, pp. v/1 – v/4 Vol. 5.
- [61] M. Adams and R. Ward, “Jasper: a portable flexible open-source software tool kit for image coding/processing,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 5, may 2004, pp. V – 241–4 vol.5.
- [62] F. Ebrahimi, M. Chamik, and S. Winkler, “Jpeg vs. jpeg 2000: an objective comparison of image encoding quality,” *Proceedings of SPIE*, vol. 5558, pp. 300–308, 2004. [Online]. Available: <http://link.aip.org/link/?PSI/5558/300/1&Agg=doi>
- [63] M. Boliek, C. Christopoulos, and E. Majani, “Jpeg 2000 part i final committee draft version 1.0,” ISO/IEC JTC1/SC29 WG1, <http://www.jpeg.org/public/fcd15444-1.pdf>, Mar. 2000.
- [64] A. Said and W. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, no. 3, pp. 243–250, jun 1996.
- [65] A. Bovik, *Handbook of image and video processing*, ser. Communications, Networking and Multimedia. Elsevier Academic Press, 2005. [Online]. Available:

http://books.google.com.br/books?id=UM_GCfJe88sC

- [66] P. Topiwala, *Wavelet image and video compression*, ser. Kluwer international series in engineering and computer science. Kluwer Academic, 1998. [Online]. Available: <http://books.google.com.br/books?id=4fZtU2pPMZIC>
- [67] K. Rao and P. Yip, *The transform and data compression handbook*, ser. Electrical engineering and signal processing series. CRC Press, 2001. [Online]. Available: <http://books.google.com.br/books?id=Fj1x1woS1ggC>
- [68] K. Sayood, *Introduction to data compression*, ser. Morgan Kaufmann series in multimedia information and systems. Morgan Kaufmann Publishers, 2000. [Online]. Available: <http://books.google.com.br/books?id=ChSOjgiY84YC>
- [69] Y. Iano, F. da Silva, and A. Cruz, “A fast and efficient hybrid fractal-wavelet image coder,” *Image Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 98 –105, jan. 2006.
- [70] A. L. M. C. S. da Silva, “Procedimentos para metodo hibrido de compressão de imagens digitais utilizando transformadas wavelet e codificação fractal,” Ph.D. dissertation, Engenharia Elétrica e Computação, Dept. Comunicações, Universidade Estadual de Campinas, Campinas, São Paulo, Brasil, Jul. 2005. [Online]. Available: <http://www.bibliotecadigital.unicamp.br/document/?code=vtls000355467&fd=y>
- [71] H.-C. Hsin, J.-J. Lien, and T.-Y. Sung, “A hybrid spiht-ebc image coder,” in *IMECS*, 2007, pp. 1854–1857.
- [72] W. Li, Z. P. Pang, and Z. J. Liu, “Spiht algorithm combined with huffman encoding,” in *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, april 2010, pp. 341 –343.
- [73] H. Pan, W. Siu, and N. Law, “Efficient and low-complexity image coding with the lifting scheme and modified spiht,” in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, june 2008, pp. 1959 –1963.
- [74] A. Kumarayapa, X.-F. Zhang, and Y. Zhang, “Simplifying spiht for more memory efficient onboard machine-vision codec and the parallel processing architecture,” in *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 3, aug. 2007, pp. 1482 –1486.
- [75] L. W. Chew, L.-M. Ang, and K. P. Seng, “New virtual spiht tree structures for very low memory strip-based image compression,” *Signal Processing Letters, IEEE*, vol. 15, pp. 389 –392, 2008.
- [76] W. A. Pearlman and A. Said, “Set partition coding: Part i of set partition coding and image wavelet coding systems,” *Found. Trends Signal Process.*, vol. 2, pp. 95–180,

- February 2008. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1482328.1482329>
- [77] R. F. Larico and Y. Iano, “Exemplo spiht 64x64,” ODS, Nov. 2009, laboratorio Lev-Decom-Feec-Unicamp. [Online]. Available: <http://www.decom.fee.unicamp.br/~rlarico/SPIHT/>
- [78] J. E. Fowler, “Qccpack: An open-source software library for quantization, compression, and coding,” in *Proceedings of the Conference on Data Compression*, ser. DCC '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 554–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=789087.789762>
- [79] W. Pearlman, A. Islam, N. Nagaraj, and A. Said, “Efficient, low-complexity image coding with a set-partitioning embedded block coder,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 11, pp. 1219 – 1235, nov. 2004.
- [80] W. A. Pearlman and A. Said, “Image wavelet coding systems: Part ii of set partition coding and image wavelet coding systems,” *Found. Trends Signal Process.*, vol. 2, pp. 181–246, March 2008. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1482330.1482331>
- [81] Consultative Committee for Space Data Systems (CCSDS), “CCSDS.org - Publications,” Office of Space Communication, NASA, Washington, DC 20546, USA, Copyright 2011. [Online]. Available: <http://public.ccsds.org/publications/>
- [82] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.
- [83] X. Xu and Y. Zhou, “Design of image data compression ip core based on processor local bus,” in *Database Technology and Applications (DBTA), 2010 2nd International Workshop on*, nov. 2010, pp. 1 –4.
- [84] J. Zhang, M. Jing, and W. Shaochuan, “Robust transmission of progressive images in the deep space communication,” in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, june 2010, pp. 4 –8.
- [85] K. Kannan and S. Arumuga Perumal, “Optimal decomposition level of discrete wavelet transform for pixel based fusion of multi - focused images,” in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 3, dec. 2007, pp. 314 –318.
- [86] S. Rein and M. Reisslein, “Low-memory wavelet transforms for wireless sensor networks: A tutorial,” *Communications Surveys Tutorials, IEEE*, vol. 13, no. 2, pp. 291 –307, quarter 2011.

- [87] J. Kulkarni, "Wavelet transform applications," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 1, april 2011, pp. 11 –17.
- [88] J. Oliver and M. Perez Malumbres, "On the design of fast wavelet transform algorithms with low memory requirements," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 237 –248, feb. 2008.
- [89] A. Klappenecker, F. U. May, and A. Nueckel, "Lossless image compression using wavelets over finite rings and related architectures," A. Aldroubi, A. F. Laine, and M. A. Unser, Eds., vol. 3169, no. 1. SPIE, 1997, pp. 139–147. [Online]. Available: <http://link.aip.org/link/?PSI/3169/139/1>
- [90] R. K. Bhattar, K. Ramakrishnan, and K. Dasgupta, "Strip based coding for large images using wavelets," *Signal Processing: Image Communication*, vol. 17, no. 6, pp. 441 – 456, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092359650200019X>
- [91] B. C. Rao and M. M. Latha, "Analysis of multi resolution image denoising scheme using fractal transform," *The International Journal of Multimedia and Its Applications (IJMA)*, vol. 2, no. 3, pp. 63–74, 2010.
- [92] L.-B. Zhang, "Low-complexity multiple roi image coding method based on different degrees of interest," in *Electronic Imaging and Multimedia Technology V*, L. Zhou, C.-S. Li, and M. M. Yeung, Eds., vol. 6833, no. 1. SPIE, 2007, p. 68331E. [Online]. Available: <http://link.aip.org/link/?PSI/6833/68331E/1>
- [93] R. de Queiroz, R. Ortis, A. Zaghetto, and T. Fonseca, "Fringe benefits of the h.264/avc," in *Telecommunications Symposium, 2006 International*, sept. 2006, pp. 166 –170.
- [94] R. G. de Oliveira, "Avaliação do desempenho de transformadas sobrepostas e wavelets nos codificadores padrão jpeg2000 e h.264/avc," Master's thesis, Universidade de Brasilia, 2008.
- [95] B. Mortara, "O formato jpeg 2000, o sucessor do jpeg," <http://www.revistadesktop.com.br>, April 2011, n. 122. [Online]. Available: <http://www.revistadesktop.com.br>
- [96] Y. Xie and G.-Q. Han, "Roi coding with separated code block," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 9, aug. 2005, pp. 5447 –5451 Vol. 9.
- [97] C. Jung, D. Jun, J. Oh, H. Park, and J. Ha, "Region-of-interest based pixel domain wyner-ziv coding," in *Military Communications Conference, 2010 - Milcom 2010*, nov. 2010, pp. 283 –286.
- [98] P. Artameeyanant, "Image watermarking using adaptive tabu search," in *ICCAS-*

- SICE*, 2009, aug. 2009, pp. 1941 –1944.
- [99] P. Bharti, S. Gupta, and R. Bhatia, “Comparative analysis of image compression techniques: A case study on medical images,” in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, oct. 2009, pp. 820 –822.
- [100] F. W. Wheeler and W. A. Pearlman, “Low-memory packetized spiht image compression,” *Symposium A Quarterly Journal In Modern Foreign Literatures*, vol. 2, pp. 2–6, 2002. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/-wrapper.htm?arnumber=831896>
- [101] I. Zyout, I. Abdel-Qader, and H. Al-Otum, “Embedded roi coding of mammograms via combined spiht and integer wavelet transforms,” in *Electro/Information Technology, 2007 IEEE International Conference on*, may 2007, pp. 173 –177.
- [102] A. Cuhadar and S. Tasdoken, “Multiple arbitrary shape roi coding with zerotree based wavelet coders,” in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 3, july 2003, pp. III – 157–60 vol.3.
- [103] K. hyeong Park, C. S. Lee, and H. Park, “Region-of-interest coding based on set partitioning in hierarchical trees,” in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, 2001, pp. 804 –807 vol.3.
- [104] L.-B. Zhang and K. Wang, “New approach for jpeg2000 region of interest image coding hybrid bitplane shift,” in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 6, aug. 2004, pp. 3955 – 3960.
- [105] F. Idris and F. Atef, “An efficient method for region of interest coding in jpeg2000,” in *Proceedings of the 5th WSEAS international conference on Signal processing*, ser. SIP'06. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 65–69. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1983937.1983953>
- [106] Z. Li-Bao, “Region of interest image coding using iwt and partial bitplane block shift for network applications,” in *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*, sept. 2005, pp. 624 – 628.
- [107] L. Liu and G. Fan, “A new jpeg2000 region-of-interest image coding method: partial significant bitplanes shift,” *Signal Processing Letters, IEEE*, vol. 10, no. 2, pp. 35 – 38, feb 2003.
- [108] Z. Wang and A. Bovik, “Bitplane-by-bitplane shift (bbbshift) - a suggestion for jpeg2000 region of interest image coding,” *Signal Processing Letters, IEEE*, vol. 9, no. 5, pp. 160 –162, may 2002.
- [109] Z. Wang, S. Banerjee, B. Evans, and A. Bovik, “Generalized bitplane-by-bitplane

- shift method for jpeg2000 roi coding,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, 2002, pp. III–81 – III–84 vol.3.
- [110] X.-W. Yin, A. Downton, M. Fleury, and J. He, “A region-of-interest method for texturally-rich document image coding,” *Signal Processing Letters, IEEE*, vol. 11, no. 11, pp. 912 – 917, nov. 2004.
- [111] E. Atsumi and N. Farvardin, “Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees,” in *Image Processing, 1998. ICIIP 98. Proceedings. 1998 International Conference on*, vol. 1, oct 1998, pp. 87 –91 vol.1.
- [112] C. E. Shannon, W. Weaver, and Shannon, *The Mathematical Theory of Communication*. University of Illinois Press, Sep. 1998. [Online]. Available: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0252725468>
- [113] A. Morello and V. Mignone, “Dvb-s2: the second generation standard for satellite broad-band services,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, 2006. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03-wrapper.htm?arnumber=1566630>
- [114] O. Collins, F. Pollara, S. Dolinar, and J. Statman, “Wiring viterbi decoders (splitting debruijn graphs),” Communications System Research Section NASA, Pasadena, Calif, TDA Prog. Report 42-96, Oct 1988. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-96/96H.PDF
- [115] O. Collins, S. Dolinar, R. McEliece, and F. Pollara, “A vlsi decomposition of the debruijn graph,” *J. ACM*, vol. 39, pp. 931–948, October 1992. [Online]. Available: <http://doi.acm.org/10.1145/146585.146620>
- [116] G. D. Forney, “Concatenated codes,” Ph.D. dissertation, Massachusetts Institute of Technology, Massachusetts, Cambridge, Dec 1966.
- [117] K. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. Jones, and F. Pollara, “The development of turbo and ldpc codes for deep-space applications,” *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2142 –2156, nov. 2007.
- [118] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes,” in *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, may 1993, pp. 1064 –1070 vol.2.
- [119] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1261 – 1271, oct 1996.
- [120] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and

- convolutional codes,” *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 429–445, mar 1996.
- [121] P. Robertson, “Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes,” in *Global Telecommunications Conference, 1994. GLOBECOM '94. Communications: The Global Bridge.*, IEEE, vol. 3, nov- 2 dec 1994, pp. 1298–1303 vol.3.
- [122] S. Benedetto and G. Montorsi, “Unveiling turbo codes: some results on parallel concatenated coding schemes,” *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 409–428, mar 1996.
- [123] D. Divsalar and F. Pollara, “Turbo codes for deep-space communications,” Communications System Research Section NASA, Pasadena, Calif, TDA Prog. Report 42-120, Feb. 1995. [Online]. Available: http://www.stanford.edu/class/ee379b/class_reader/jpl1.pdf
- [124] S. Benedetto and G. Montorsi, “Average performance of parallel concatenated block codes,” *Electronics Letters*, vol. 31, no. 3, pp. 156–158, feb 1995.
- [125] —, “Performance evaluation of turbo-codes,” *Electronics Letters*, vol. 31, no. 3, pp. 163–165, feb 1995.
- [126] D. Divsalar and F. Pollara, “Multiple turbo codes for deep-space communications,” Communications System Research Section NASA, Pasadena, Calif, TDA Prog. Report 42-121, May. 1995. [Online]. Available: http://www.stanford.edu/class/ee379b/class_reader/jpl4.pdf
- [127] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “A soft-input soft-output app module for iterative decoding of concatenated codes,” *Communications Letters, IEEE*, vol. 1, no. 1, pp. 22–24, jan 1997.
- [128] —, “Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding,” *Information Theory, IEEE Transactions on*, vol. 44, no. 3, pp. 909–926, may 1998.
- [129] CCSDS, “Flexible serially concatenated convolutional turbo codes with near-shannon bound performance for telemetry applications,” PDF, Consultative Committee for Space Data Systems, Office of Space Communication, NASA, Washington, DC 20546, USA, Orange Book, 131.1-O-2 131, September 2007. [Online]. Available: <http://public.ccsds.org/publications/archive/131x1o2e2.pdf>
- [130] R. G. Gallager, “Low-density parity-check codes,” Ph.D. dissertation, Massachusetts Institute of Technology, Massachusetts, Cambridge, may 1963.
- [131] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *Information Theory, IEEE Transactions on*, vol. 45, no. 2, pp. 399–431, mar 1999.

- [132] J. Thorpe, “Low-density parity-check (ldpc) codes constructed from protographs,” *IPN Progress Report 42154 JPL*, vol. 42, no. 154, pp. 1–7, 2005. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-154/154C.pdf
- [133] X. Chen, *Error-Control Coding for Data Networks*, I. S. Reed, Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1999.
- [134] S. H. Lee, J. A. Seok, and E. K. Joo, “Serial concatenation of ldpc and turbo code for the next generation mobile communications,” in *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on*, march 2005, pp. 425 – 427.
- [135] CCSDS, “Tm synchronization and channel coding,” PDF, Consultative Committee for Space Data Systems, Office of Space Communication, NASA, Washington, DC 20546, USA, Blue Book, 131.0-B-1 130, September 2003. [Online]. Available: <http://public.ccsds.org/publications/archive/131x0b2.pdf>
- [136] K. Andrews, S. Dolinar, D. Divsalar, and J. Thorpe, “Design of low-density parity-check (ldpc) codes for deep-space applications,” *IPN Progress Report*, vol. 42, p. 159, 2004.
- [137] CCSDS, “Low density parity check codes for use in near-earth and deep space,” PDF, Consultative Committee for Space Data Systems, Office of Space Communication, NASA, Washington, DC 20546, USA, Orange Book, 131.1-O-2 131, September 2007. [Online]. Available: <http://public.ccsds.org/publications/archive/131x1o2e2.pdf>
- [138] Y. He and P. Ching, “Performance evaluation of adaptive two-dimensional turbo product codes composed of hamming codes,” in *Integration Technology, 2007. ICIT '07. IEEE International Conference on*, march 2007, pp. 103 –107.
- [139] R. Pyndiah, “Near-optimum decoding of product codes: block turbo codes,” *Communications, IEEE Transactions on*, vol. 46, no. 8, pp. 1003 –1010, aug 1998.
- [140] X. Wu, Y. He, and G. Zhu, “Performance of improved three-dimensional turbo product code decoder,” in *Integration Technology, 2007. ICIT '07. IEEE International Conference on*, march 2007, pp. 563 –567.
- [141] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, “Near optimum decoding of product codes,” in *Global Telecommunications Conference, 1994. GLOBECOM '94. Communications: The Global Bridge., IEEE*, nov- 2 dec 1994, pp. 339 –343 vol.1.
- [142] A. P. M. Samia, “Códigos de bloco com decodificação turbo adaptados às especificações do padrão dvb-rct,” Ph.D. dissertation, INATEL/ENGENHARIA ELÉTRICA, Santa Rita do Sapucaí, Brasil, 2004.
- [143] “Primer: Turbo product codes,” PDF, 2009, comtech EF Data Corporation. [Online]. Available: http://www.aha.com/wp-content/uploads/ANtpc01_0499.pdf

- [144] D. Rankin and T. Gulliver, “Single parity check product codes,” *Communications, IEEE Transactions on*, vol. 49, no. 8, pp. 1354 –1362, aug 2001.
- [145] R. M. Shiratori, “Análise e comparação de códigos corretores de erros,” Ph.D. dissertation, Institute Technology and Aeronautic/Engenharia Eletrônica e Computação, SP, Brazil, 2007.
- [146] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate (corresp.),” *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284 – 287, mar 1974.
- [147] A. Viterbi, “An intuitive justification and a simplified implementation of the map decoder for convolutional codes,” *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 260 –264, feb 1998.
- [148] M. C. Valenti, “Turbo codes and iterative processing,” in *IEEE New Zealand Wireless Communications Symposium*, 1998, pp. 216–219.
- [149] M. Kousa and A. Mugaibel, “Puncturing effects on turbo codes,” *Communications, IEE Proceedings-*, vol. 149, no. 3, pp. 132 – 138, jun 2002.
- [150] D. A. Guimarães, “Noções sobre decodificação turbo: Um estudo de caso para os códigos spc-tpc,” *Revista Telecomunicações*, vol. 8, no. 1, pp. 15 – 29, Setembro 2005.
- [151] M. Karkooti, “Semi-parallel architectures for real-time ldpc coding,” Ph.D. dissertation, RICE UNIVERSITY, Houston, Texas, May 2004.
- [152] H. Wymeersch, H. Steendam, and M. Moeneclaey, “Computational complexity and quantization effects of decoding algorithms for non-binary ldpc codes,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 4, may 2004, pp. iv–669 – iv–672 vol.4.
- [153] A. W. Hunt, “Hyper-codes: High-performance low-complexity error-correcting codes,” Ph.D. dissertation, Carleton University, Institute of Electrical Engineering, Ottawa, Ontario, Canada, May 1998.
- [154] L. Tong, S. He, K. Tang, and Y. Wang, “An interactive-roi image transmission method,” in *Information Science and Engineering (ICISE), 2009 1st International Conference on*, dec. 2009, pp. 1153 –1156.
- [155] Surrey Satellite Technology Limited, “Earth observation constellations,” web, © SSTL 2012. [Online]. Available: <http://www.sstl.co.uk/divisions/earth-observation-science/eo-constellations>
- [156] G. Berlocher, “Small satellite technology: Gains open space to more players,” *Via satellite*, Aug. 2008. [Online]. Available: http://www.satellitetoday.com/via/features/-Small-Satellite-Technology-Gains-Open-Space-to-More-Players_23881.html

- [157] P. N. de Souza, “Histórico do programa espacial brasileiro,” pdf, Apr. 2011, unidade 1/Parte 1.4/Versão 5.1. [Online]. Available: http://www.inpe.br/twiki/pub/Main/-IntroducaoTecnologiaSatelites/140_Satelites_P1.4_v5.1_2011.pdf
- [158] “TELECO Secção: TV Digital,” TELECO, 04/12/2010. [Online]. Available: <http://www.teleco.com.br/tvdigital.asp>
- [159] M. Takada and M. Saito, “Transmission system for isdb-t,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 251–256, jan. 2006.
- [160] A. Morello, “Lo Standard DVB di seconda generalizione per l’HDTV,” RAI research centre, Rome, April 2008. [Online]. Available: <http://www.fub.it/public/-Morello020408B.pdf>
- [161] Y. Iano, D. Gara Caetano, R. Arthur, E. Carrara, and C. Akamine, “Performance comparison of convolutional and turbo convolutional codes for isdb-t in awgn and brazilian channels,” in *Telecommunications Symposium, 2006 International*, sept. 2006, pp. 361–366.
- [162] T. Halbach, “Comparison of open and free video compression systems,” in *Proc. International Conference on Imaging Theory and Applications (IMAGAPP)*, Lisboa (Portugal), Feb. 2009.
- [163] C. Keimel and K. Diepold, “On the use of reference monitors in subjective testing for hdtv,” in *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, june 2010, pp. 35–40.
- [164] C. Akamine, F. Yamada, G. Bedicks, C. da Silva Dantas, F. Sukys, R. Motoyama, R. de Mendonca Maroja, G. de Melo Valeira, and E. Horta, “Field trial for brazilian dtv using space diversity,” in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, march 2010, pp. 1–11.
- [165] D. V. de Almeida, “Tvs conectadas já são uma realidade e um caminho sem volta,” *revista da SET – Sociedade Brasileira de Engenharia de Televisão*, vol. 21, no. 124, pp. 26–27, Nov-Dez 2011.
- [166] L. A. V. da Costa Ferreira, “Metadados em Multimídia: Aplicações e Conceitos em MPEG-7,” Master’s thesis, Engenharia Elétrica e Computação, Dept. Comunicações, Universidade Estadual de Campinas, Campinas, SP, Março 2007.
- [167] A. Gradvohl and Y. Iano, “An approach for interactive television based on insertion of hypermedia information in mpeg standard video,” in *Consumer Electronics, 2004 IEEE International Symposium on*, 1-3, 2004, pp. 25–30.
- [168] S.-F. Chang, T. Sikora, and A. Purl, “Overview of the mpeg-7 standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 688–695, jun 2001.

- [169] A. L. S. Gradvohl, “TV Interativa Baseada na Inclusão de Informações Hiperídia em Vídeos no Padrão MPEG,” Ph.D. dissertation, Engenharia Elétrica e Computação, Dept. Comunicações, Universidade Estadual de Campinas, Campinas, São Paulo, Brasil, 2005.
- [170] S. R. M. de Carvalho, “Utilização da Rede SFN para Expansão da Rede de Retransmissão Regional da TV Digital Aberta Terrestre.,” Master’s thesis, Engenharia Elétrica e Computação, Dept. Comunicações, Universidade Estadual de Campinas, Campinas, SP, Agosto 2006.
- [171] R. Arthur, Y. Iano, S. R. M. de Carvalho, and R. F. Larico, “Expansion planning of the digital tv broadcasting service in brazil using sfn,” *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 5, no. 8, pp. 573 –578, dec. 2007.
- [172] C. Akamine and F. Sukys, “TV Digital,” TELECO, 23/06/2003. [Online]. Available: <http://www.teleco.com.br/tutoriais/tutorialtvd/>

Apêndices

Apêndice A:

Visão geral de aplicações de imagens em pequenos satélites e o projeto ITASAT

A.1 Introdução

O Brasil possui diversos pesquisadores que detém competências e especializações em padrões e técnicas de compressão/ processamento/ codificação de imagens. Testar sistemas nacionais desenvolvidos nessa área, a partir da oportunidade disponibilizada pelo projeto e lançamento de pequenos satélites universitários, reflete-se em ganho tecnológico e financeiro para a nação nos seguintes aspectos:

- Permite reduzir em longo prazo, os custos de imagens compradas no exterior referentes à análise de recursos em solo nacional ou monitoramento de fenômenos, por exemplo, previsão do tempo, fenômenos oceânicos, detectar e monitorar furacões, monitoramento de encostas, queimadas e desmatamentos. Além da geração de mapas geológicos, de solos e de uso da terra, mapear os recursos hídricos, as áreas agrícolas e as áreas urbanas e acompanhar sua transformação e expansão, monitoramento de pesca, monitoramento de cargas perigosas, rastreamento de animais, planejamento agrícola, entre outras aplicações: a maioria delas com grande retorno financeiro e de grande impacto social.
- Permite desenvolver parcerias com entidades privadas para aplicações de alto retorno econômico, tais como aplicações que envolvam telecomunicações: comunicação especializada, rastreamento, imageamento, sensoriamento remoto e missões militares.
- Permite embarcar e testar demonstradores de tecnologia, desenvolvendo-se tecnologia e *know-how* nacionais em substituição à compra dos mesmos. Gera-se propriedade intelectual nacional, reduzem-se custos para o país em projetos futuros, permitindo que o Brasil, a longo prazo, firme-se como potência tecnológica que, de fato, é capaz.

A meta principal dessa parte é ser informativa para poder visualizar a importância de

gerar um esquema de codificadores (fonte e canal) que tenha as características necessárias para ser embarcado em um satélite.

A.2 Projeto ITASAT

O projeto ITASAT [4], consiste em uma parceria do Instituto Tecnológico da Aeronáutica (ITA), Ministério da Ciência e Tecnologia (MCT), Agência Espacial Brasileira (AEB) e Instituto Nacional de Pesquisas Espaciais (INPE). A Universidade Estadual de Campinas (UNICAMP), USP-São Carlos e outras instituições atuaram como participantes para desenvolver pesquisas associadas referentes à partes desse programa. Figuram entre os objetivos promover a capacitação de alunos universitários de graduação e pós-graduação na área de engenharia elétrica, mecânica e afins, por meio do desenvolvimento de um satélite artificial de pequeno porte, a figurar como o primeiro satélite universitário brasileiro [4]. A fase de implementação e prototipagem mostra os resultados em sentido prático.

Uma vez colocada a importância da aplicação de imagens em pequenos satélites e o contexto das aplicações comerciais no panorama nacional e mundial, firmando esse segmento como o futuro comercial e tecnológico no segmento de tecnologia espacial, espera-se configurar a importância dos procedimentos e estudos realizados pelo subsistema experimental de carga útil: códigos corretores de erros (canal) e compressão de dados (fonte) gerados pelo acordo ITA-Unicamp dentro do contexto da missão ITASAT.

A.3 Aplicações de imagens em pequenos satélites

O acesso aos avanços da microeletrônica, microprocessadores, redução das dimensões dos *gadgets*, estudos com nanotecnologia e o desenvolvimento de pequenos satélites permitiram que serviços com as mais diversas aplicações e vantagens ultrapassassem as cercas das grandes corporações; sendo acessíveis para países, organizações, governos e instituições de menor porte, organizando seus próprios programas de pequenos satélites. Tais aplicações vêm sendo apontadas como uma nova fronteira para competição doméstica e internacional de uma indústria economicamente dinâmica.

Imagens espaciais: importância

Dentro desse contexto, como um bom exemplo da importância do domínio de imagens espaciais hoje (aquisição, tratamento, codificação, compressão e manipulação), as imagens obtidas de satélite podem ser utilizadas no estudo e monitoramento dos mais variados fenômenos e feições da superfície terrestre. A partir da interpretação de imagens é possível fazer a previsão de tempo, estudar fenômenos oceânicos, detectar e monitorar furacões, queimadas e desflorestamentos, gerar mapas geológicos, de solos e de uso da terra, mapear

os recursos hídricos, as áreas agrícolas e as áreas urbanas, e acompanhar sua transformação e expansão, entre outras aplicações.

Para se ter noção de tal importância, considerando, por exemplo, o monitoramento do desmatamento na floresta amazônica, com a estrutura disponível hoje pelo Brasil, o INPE (Instituto Nacional de Pesquisas Espaciais) tem que fazer uso hoje não só de imagens dos CBERS (*China-Brazil Earth-Resources Satellite*) e *Landsat*, mas também frequentemente tem que ser compradas imagens no exterior, da rede DMC (*Disaster Monitoring Constellation*). As imagens são usadas para aprimorar os dados com estimativas anuais de desmatamento da região.

A.4 Contexto: Aplicações comerciais de pequenos satélites

Satélites maiores usualmente dominavam a indústria, relegando aos pequenos satélites somente aplicações científicas. O Intelsat-6, por exemplo, um satélite de comunicações, projetado para vida útil de 10-14 anos, pesava 4600kg com dimensões de 6,4×3,6×11,8m, gerando 2600W e podendo suportar até 120 mil canais telefônicos e 3 canais de TV. Satélites assim possuem altos custos e uma única falha dentro da órbita pode ser bastante custosa financeiramente [155].

Como uma reação ao avanço da microeletrônica (em particular, microprocessadores) e à diminuição das dimensões dos *gadgets*, a indústria espacial começou desde os anos 90 a enfatizar a filosofia do “*smaller, faster, better, cheaper*” e tem focado atenção nas capacidades e vantagens que pequenos satélites podem trazer as novas aplicações. Assim, os serviços ‘via satélite’ ultrapassaram o território das grandes corporações, antes, as únicas capazes de arcar com os altos preços dos serviços fornecidos pelas grandes espaçonaves. De custos bastante reduzidos, os pequenos satélites foram sendo adquiridos por várias organizações, de diversos segmentos e portes. Indústrias, governos, universidades e outras organizações estão iniciando seus próprios programas de pequenos satélites. Além disso, os satélites de pequeno porte permitiram que países sem tradição possam, agora, almejar um caminho para alcançar o espaço [156]. Um micro-satélite moderno típico pesa 50kg, com dimensões 0,6×0,4×0,3m e gera 30W. Pequenos satélites possuem tempo de desenvolvimento mais curto e como se costuma aplicar tecnologia de ponta sempre, seu tempo de vida útil acaba expirando mais pelo rápido avanço tecnológico do que por falhas no sistema. A Nasa vem juntando engenheiros e cientistas para desenvolver e lançar nanosatélites, provendo valiosas oportunidades para testar tecnologias emergentes e componentes comerciais econômicos que serão úteis em futuras missões espaciais. Usando satélites pequenos e muito pequenos a NASA pode testar novas tecnologias em escala menor em ambiente espacial e compreender melhor como o *hardware* sobrevive à radiação, temperatura e vácuo espaciais. Tais nanosatélites são projetados para um amplo espectro de

missões espaciais incluindo experimentos biológicos, testando propulsão avançada e tecnologias de comunicações [156]. Existe um projeto que foi desenvolvido e testado pela Goddard; e é parte do *New Millennium Program*, dirigido pela NASA pelo *agency's Jet Propulsion Laboratory*, Pasadena, Califórnia [156], que desenvolve e testa tecnologias de alto retorno que podem resultar em futuras capacidades para missões espaciais com custo e risco reduzidos. Nesse sentido também, foi criado o *NASA's Small Spacecraft Technology Initiative* (SSTI), um programa projetado para demonstrar tecnologias para redução de custo e tempo em missões civis e comerciais, servindo às necessidades da NASA de realizar missões espaciais mais frequentemente e aumentando consideravelmente a postura competitiva dos fabricantes espaciais americanos no mercado global. Também existem muito exemplos de satélites experimentais desenvolvidos com recursos privados e públicos e combinam experiências com tecnologia inovadora como uma nova forma de fazer negócios [155, 157]. Um grande número de constelações de pequenos satélites tem sido proposto para comunicações, sensoriamento remoto e missões *científicas* espaciais. Esses sistemas, sem dúvida nenhuma, irão estabelecer definitivamente os pequenos satélites no mercado.

Pequenos satélites no Brasil

No Brasil há projetos de quatro pequenos satélites: dois já operacionais de coleta de dados ambientais, SCD1 e SCD2, e dois planejados de sensoriamento remoto (imageamento), SSR1 (Amazônia 1) e SSR2 (Marsar), além da implantação da infraestrutura de solo. Os bem sucedidos satélites SCD1 e SCD2, lançados em 1993 e 1998, com a missão de captar e retransmitir dados ambientais de uma rede de PCDs (plataformas de coleta de dados ambientais) em todo o território nacional, apresentando desempenho bastante satisfatório. Paralelamente, foi criada uma importante infra-estrutura de solo, que inclui o LIT (Laboratório de Integração e Testes), o único no hemisfério sul; o Centro de Missão de Coleta de Dados; e o Centro de Rastreamento e Controle de Satélites. O sucesso do SCD1 e do SCD2 permitiu a expansão do Sistema Brasileiro de Coleta de Dados Ambientais, com aplicações em áreas como meteorologia, hidrologia, oceanografia, e planejamento agrícola [157].

Há também a linha CBERS (Satélite Sino-Brasileiro de Recursos Terrestres), acordo de cooperação tecnológica entre Brasil e China que prevê dois satélites de observação da Terra, com aplicação em gestão ambiental e agricultura. O *CBERS-1*, foi lançado em 1999, e operou até 2003. O *CBERS-2*, lançado em 2003 é maior distribuidor de imagens orbitais do mundo, e operou até janeiro de 2009 com sucesso. O *CBERS-2B*, foi lançado em setembro de 2007. Os demais *CBERS-3* e *4* estavam sendo desenvolvidos para possíveis

lançamentos em 2011 e 2014^{3,4}.

Nesse mesmo contexto, em 2007, foi adquirido para os Laboratórios Associados do INPE um Sistema de Microscopia de Força Atômica *Veeco Multimode* com estação de controle *Nanoscope V*, que permite observar os objetos em escala nanométrica. A nanotecnologia tem um papel fundamental na redução do peso, tamanho e consumo de energia desses pequenos satélites. A razão para se fabricar satélites miniaturizados é a enorme redução do custo do lançamento. Eles podem ser lançados em um grande número ou podem, inclusive, pegar carona em veículos lançadores de grande porte.

O satélite radar MAPSAR, parceria com a DLR (agência espacial da Alemanha), LLB (*Institute of Lightweight Structures - Alemanha*) e o INPE, também se encontra em análise e desenvolvimento de etapas. Já o Satélite GPM-BR inclui a participação brasileira em uma rede internacional de constelação de nove satélites para o monitoramento de chuvas, mudanças climáticas e alertas de desastres naturais. O Brasil deverá desenvolver um satélite específico para essa constelação, o GPM-BR (negociação com a NASA para acelerar o projeto de lançamento).

3 Os satélites artificiais produzidos pelo Inpe, <http://www.unesp.br/aci/jornal/231/suplec.php>

4 CBERS-2 encerra vida útil, <http://www.cbbers.inpe.br/noticias/index.php?cod=not155>

Apêndice B:

Criação automática de vetores binários

Para explorar a melhor codificação ROI nessas condições, foi criada uma forma de gerar vetores binários, dados alguns passos específicos. A heurística de busca em vizinhança, como a busca local é aplicada, a qual necessariamente parte de uma solução inicial viável (semente) que são vetores conhecidos provenientes das pesquisas no JPEG-2000. Logo, a melhor resposta é armazenada (depende de um limiar) para reiniciar todas as variáveis (novos vetores binários $vsROI$ e $vsBG$) e começar uma nova geração de soluções a partir da encontrada (elitismo). Como fatores de diversidade, se realiza a troca de *bits* entre $vsROI$ e $vsBG$ (pontos de cruzamento) e quando não se cumpre com mesma quantidade de *bits* (desejada por vetor) se aplica uma troca isolada em um mesmo vetor (mutação) que produz sobreposições entre vetores e diversidade parecida com o antecessor. A Fig. 7.1 mostra o fluxograma do algoritmo para gerar vetores binários.

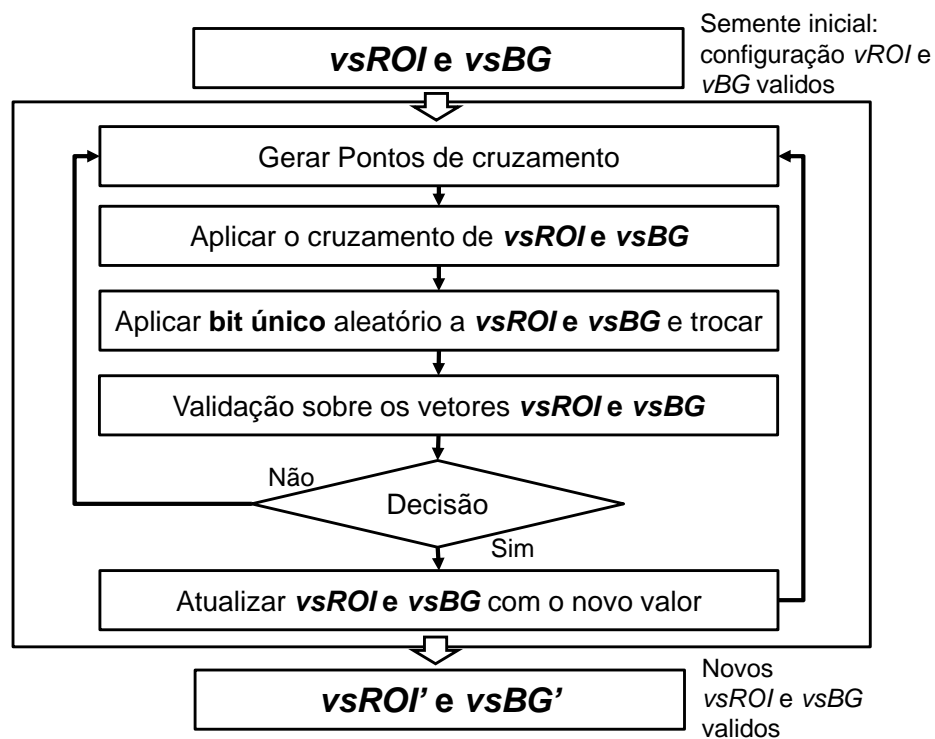


Fig. 7.1. Fluxograma do algoritmo de criação de vetores binários $vROI$ e vBG .

Os passos são observados como exemplo ilustrativo na Fig. 7.2. Primeiramente, os vetores contém um número constante de 1's, o vetor *vsROI* sempre deve ser maior e depois do cruzamento de *bits* deve ser conferido (caso contrário reiniciar o passo). Segundo, a variabilidade aleatória é feita trocando-se *bits* para que se cumpra o primeiro passo (se for necessário). Finalmente, é calculada a métrica (nesse caso o PSNR) e são salvos somente os que tenham um desempenho mínimo (limiar ~20dB e para até 1,5 bpp). Então, o processo se reinicia até atingir um número definido de iterações. Como é uma busca local, os vetores iniciais devem ser conhecidos em cada conjunto de iterações.

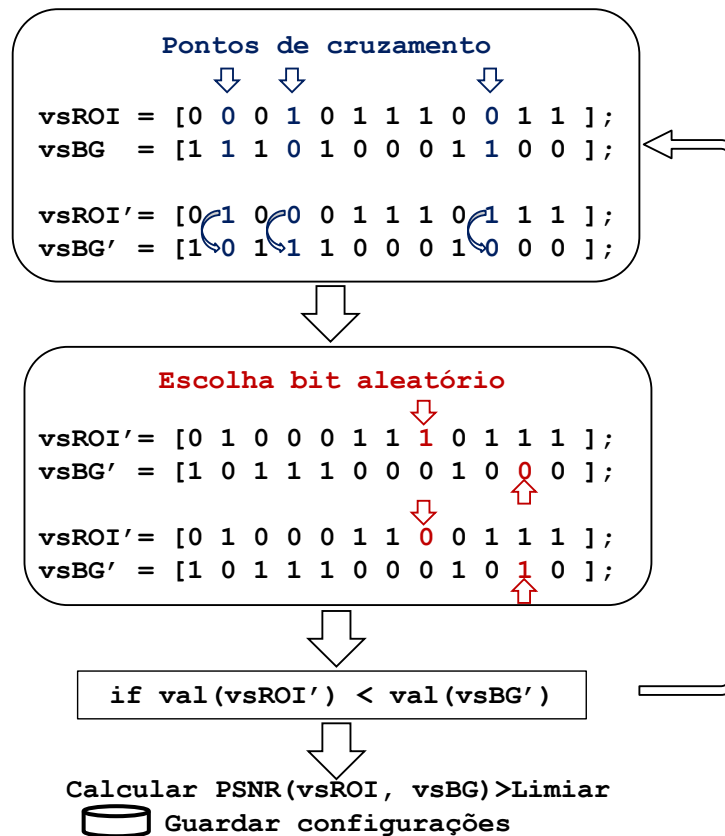


Fig. 7.2. Exemplo de como aplicar uma iteração do gerador de vetores binários no *vsROI* e *vsBG* e guardar os melhores resultados.

Apêndice C:

Aplicações da pesquisa na TV digital

C.1 TV digital: visão geral

A TV digital é um conjunto de tecnologias que proporciona uma transmissão e recepção de maior quantidade de conteúdo que a TV analógica em um mesmo intervalo de frequência. Na TV digital, a transmissão de áudio e de vídeo passa a ser feita por meio de sinais digitais que, codificados, permitem um uso mais eficiente do espectro eletromagnético, devido ao aumento da taxa de transmissão de dados na banda de frequências disponível [158].

Na TV digital é possível transmitir imagens de **melhor qualidade** viabilizando a televisão de alta definição HDTV (*High Definition Television*) ou obter **mais canais** (até 4) usando a mesma faixa de frequências utilizada por um canal analógico. Também apresenta algumas funcionalidades que permitem uma **interatividade** entre o telespectador e a emissora.

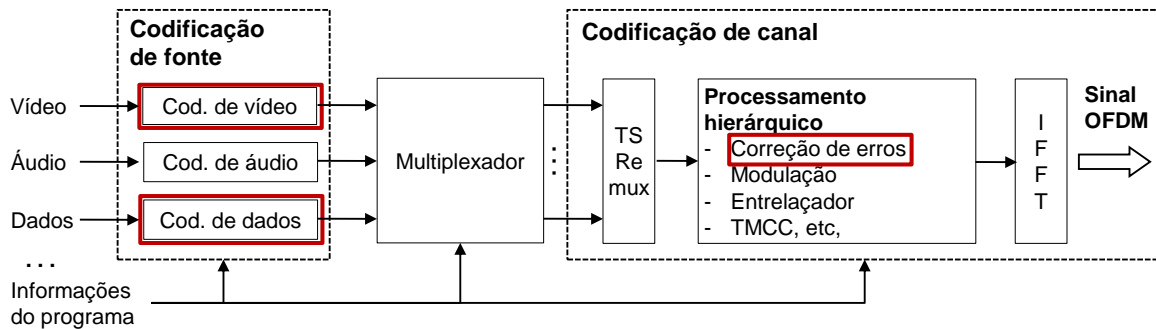


Fig. 7.3. Configuração do sistema ISDB-T, ressaltando as partes onde as pesquisas atuais podem ser aplicadas [159].

A TV digital utilizada no Japão é o padrão ISDB-T. Conforme se observa na Fig. 7.4, a configuração do sistema tem módulos gerais como a codificação de fonte, multiplexação MPEG-2 e a codificação de canal. Cada um desses blocos é responsável por uma tarefa especificada na norma e pode ser implementado de diversas formas.

As tecnologias que existem no mundo são sistemas de TV digital terrestre como o sistema americano (ATSC), o sistema europeu (DVB), e o sistema japonês (ISDB). Outros sistemas estão em desenvolvimento, mas o esquema geral na Fig. 7.4 se conserva. Por

exemplo o sistema DVB-T tem a segunda geração DVB-T2 onde se implementou melhoras utilizando-se esquemas de proteção de erros usando o LDPC (que já era utilizado no padrão europeu de satélite) e esquemas de modulação que no sistema inicial era de 64QAM para 256QAM (*Quadrature Amplitude Modulation*). Esses avanços trazem um ganho aproximado de 67% a mais na taxa de transmissão [160]. No sistema ISDB-T existem propostas para utilizar o codificador turbo convolucional TCC [161] e assim conseguir melhores esquemas de modulação e codificação de vídeo com ainda mais resolução.

Em outras frentes de pesquisa tem-se o codificador de vídeo Dirac. Esse codificador de vídeo de código aberto desenvolvido pela BBC oferece ferramentas de compressão de vídeo em geral desde resoluções progressivas QCIF (180x144) até HDTV (1920x1080). Apresenta a característica de ser de baixa complexidade e custo para compressão de imagens (utiliza uma baixa quantidade de módulos). Usa compressão com esquemas *wavelets* multi resolução, compensação de movimento e codificação aritmética. O padrão pretende ser competitivo em relação a outros codificadores, tais como o codificador H.264 de alta complexidade. A Fig. 7.4 mostra o desempenho do codificador Dirac, H.264 e outro codificador experimental como o Teora [162]. Observa-se que a qualidade em PSNR do Dirac é próxima a do H.264. Em termos subjetivos, MOS de Keimel [163] apresentou o Dirac como tendo resultados de boa qualidade.

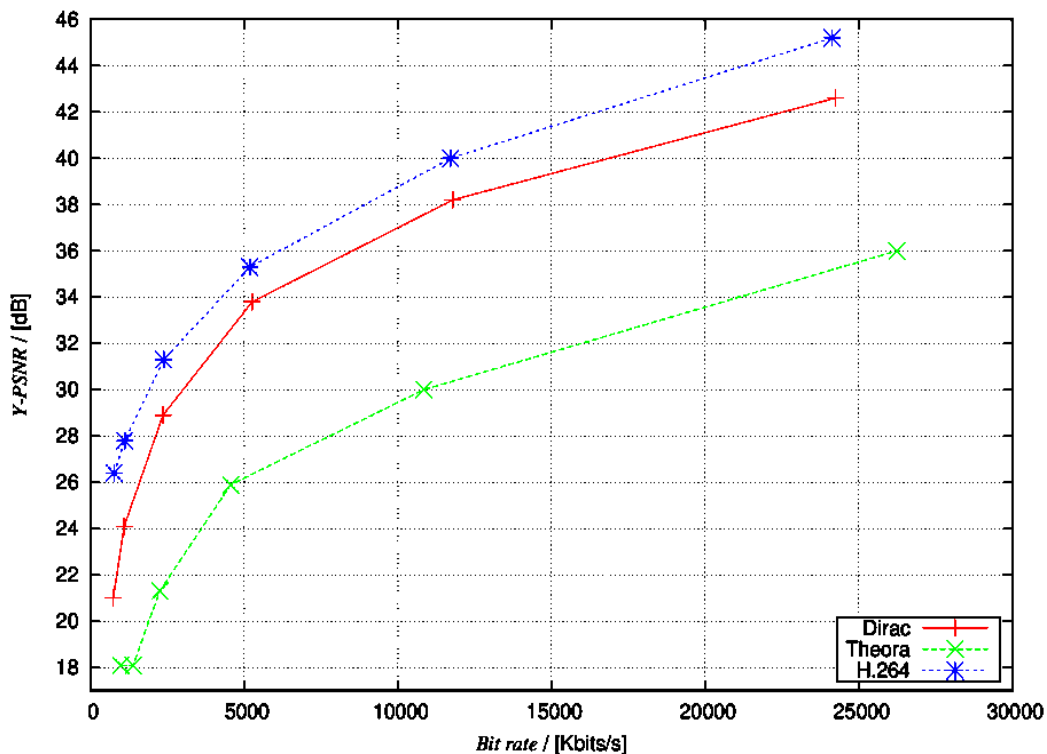


Fig. 7.4. Desempenho dos codificadores em PSNR e taxa de *bits* para uma sequência de vídeo em dimensão SD (720x480 *pixels*, 4:3) [162].

Receptores de TV

Com a introdução da tecnologia digital na radiodifusão de TV (TV digital terrestre), o usuário pode optar por uma das seguintes situações: a) Continuar a receber a TV aberta da forma atual utilizando a sua TV analógica (até 2016 quando ocorre o apagão analógico); b) Adquirir um conversor (*set top box*) que permite receber um sinal digital e convertê-lo para um formato de vídeo e áudio disponível em seu receptor de TV analógico ou, c) Adquirir uma TV fixa ou móvel nova que já incorpore o conversor [164].

Interatividade e padrão MPEG7

Ginga é o nome do *middleware* padrão do sistema brasileiro de TV digital terrestre (ISDB-TB). Essa camada de *software* proporciona bibliotecas de funções (formato NCL e Lua) que permitem o desenvolvimento rápido e fácil de aplicações. Ginga é de código aberto e reúne um conjunto de tecnologias e inovações brasileiras que o tornam a especificação de *middleware* mais avançada e, ao mesmo tempo, mais adequada à realidade do país (portal do *software* público brasileiro, <http://www.softwarepublico.gov.br>). A capacidade de interatividade do Ginga está embutida nas TVs produzidas no Brasil e estima-se que até 2015, 100% dos televisores produzidos terão o *middleware* baseado no Ginga.

Outra forma de realizar a interatividade é o mercado das TVs conectadas. Essa nova tecnologia está levantando um grande investimento no mundo e talvez seja o futuro da televisão no mundo. O modelo de tecnologia é compartilhar conteúdos e depende da *internet* para funcionar. No Brasil, se aponta como tecnologia promissora. O problema é que na atualidade somente 5,5% da população brasileira possui *internet*. Além disso, para um adequado funcionamento (banda larga), também existe uma falta de padronização. No entanto, a interatividade comumente funciona sobre um sistema operacional livre como o *android*. Deve-se ressaltar que a TV conectada e o *middleware* Ginga não são concorrentes, mas sim duas plataformas distintas, que podem conviver juntas em um único aparelho [165].

O MPEG-7 é um padrão ISO/IEC, ele foi aceito no ano de 2001, com o nome formal de "interface de descrições de conteúdos de multimídia" e tem como objetivo a descrição de conteúdos de mídia utilizando a anexação de metadados ao conteúdo de mídia. O MPEG-7 fornece um conjunto de ferramentas descritivas que podem ser utilizadas para a descrição de vários tipos de arquivos de multimídia. Ele trabalha em conjunto com os outros formatos MPEG, adicionando funcionalidades a eles [166].

O codificador MPEG-7 em vídeo, trabalha no vídeo com descritores com a característica 'forma'. Essa forma pode estar baseada em caixa de contorno, região, contorno ou descritor em 3D. Outros descritores para vídeo e áudio podem ser encontrados [167]. Logo a

inserção de metadados é obtida do conteúdo de mídia por extração manual ou automática [168]. Os metadados são armazenados pela geradora e são enviados pelo meio de transmissão (conjunto de descrições) para que os usuários possam interagir de acordo com essa informação (que seria controlada em parte pelo usuário através da aplicação da interatividade).

Esses conteúdos multimeios podem estar em grande parte no vídeo, dada a sua natureza. Para obter descritores é possível utilizar regiões de interesse no conteúdo visual e por exemplo, fazer um hipervínculo *web* no vídeo ou um hipervídeo [167, 169] entre outras aplicações. Assim, em conjunto, o MPEG-7 e regiões de interesse (ROI) podem ser aplicadas na codificação de fonte para ressaltar e priorizar essa informação no caso de transmissões de TV digital móvel onde a resolução é baixa e a taxa é limitada. Dessa forma, é possível aumentar a resolução (atualmente os dispositivos possuem relativamente altas resoluções em dispositivos móveis, por exemplo, celulares com TV digital de 800×480 *pixels*) permanecendo na mesma taxa e ainda assim ficar perto dessas resoluções maiores.

Também existe o MPEG-21 que é um *framework* aberto normativo para entrega e consumo multimídia para uso por todos os executores em uma cadeia de entrega e consumo. Essas considerações referem-se às possíveis aplicações dentre as aplicações no *middleware* e nos conteúdos do MPEG-7.

Implantação da TV digital no Brasil

A TV aberta, transmitida para os televisores existentes em 90% das residências brasileiras, utiliza canais analógicos com largura de banda de 6 MHz [41, 170, 171].

Em jun/06 o Brasil adotou o padrão japonês (ISDB) para a TV digital terrestre. O sistema brasileiro de televisão digital terrestre ISDB-T_B adota, como base, o padrão de sinais do ISDB-T e possibilita uma transmissão digital em alta definição (HDTV) e em definição padrão (SDTV). Permite também transmissão digital simultânea para recepção fixa, móvel e portátil bem como interatividade. A transmissão analógica continuará ocorrendo, simultaneamente com a digital, por um período de 10 anos até 29/06/2016. A partir de jul/2013 somente serão outorgados canais para a transmissão em tecnologia digital. Deverão ser consignados pelo menos quatro canais digitais para a exploração direta pela união federal como canal do poder executivo, canal de educação, canal de cultura e canal de cidadania [172].

Importância

A transmissão de TV digital em HDTV é a alavanca motriz de toda a TV digital. A importância desse tipo de transmissão é tal que é comparada a mudança da TV monocromática para a colorida. Comercialmente, a venda de TV de plasma ou LCD aumentam com a implantação da TV digital. Esse comportamento é natural no ser humano

e independe de classe social. Assim, os fabricantes de aparelhos eletrônicos sabem explorar muito bem esse fato e é possível abrir ainda mais mercados novos como serviços de interatividade [170].

C.2 Aplicação das pesquisas na TV digital

Pesquisas atuais como o presente trabalho podem ser aplicadas diretamente na configuração do sistema de TV digital como observado na Fig. 7.3. Nessa figura ressaltam-se os módulos que podem ser modificados para futuras implantações ou melhorias na TV digital.

Codificação de fonte

Codificação de vídeo

Um dos módulos mais complexos é a codificação de vídeo. No padrão brasileiro é codificado em H.264. Esse codificador tem uma complexidade alta e pode-se aplicar a proposta de utilizar um esquema de codificação usando o SPIHT com módulos que utilizem menor quantidade de memória com desempenho objetivo e subjetivo semelhante ao atual. Nesse contexto, a proposta do capítulo 3, adaptado para codificar/decodificar pode ser inserido no sistema de TV digital (recepção móvel) para diminuir os custos de armazenamento e complexidade. Em escala comercial o ganho pode ser considerável.

A proposta deste trabalho também pode ser aplicada no codificador de vídeo Dirac que utiliza um esquema *wavelet*. A diminuição do custo pode vir simplesmente em abaixar o custo da TWD desse codificador.

Codificação de vídeo e codificação de dados

Na maioria dos padrões de imagens estáticas, existe a possibilidade de se utilizar a codificação de região de interesse como no JPEG-2000 [13]. Em esquemas novos também é possível adaptar essa capacidade como se analisou no capítulo 4. A possível aplicação de utilizar uma codificação de ROI é uma vantagem para recepção móvel porque há limitantes de banda e resolução. É possível aplicar, mas existe o problema de encontrar qual região para se codificar. Nesse contexto, essas informações podem estar descritas segundo o padrão MPEG-7 e assim, o receptor pode, por exemplo, saber que existe uma ROI que tem melhor qualidade e que possivelmente tem uma outra informação multimídia relacionada. Assim, a operadora ou geradora desses vídeos seriam responsáveis para controlar essas codificações de ROI, armazenar no padrão MPEG-7 para que uma aplicação interativa possa tirar vantagens desse conteúdo prioritário.

Codificação de canal

Codificação de canal para proteção de erros

Como já observado, os sistemas atuais de TV digital, como o DVB-T2 utilizam LDPC e propostas indicam que é viável aplicar o turbo convolucional TCC no ISDB-T. De fato, os testes do capítulo 5 também mostram que a codificação de canal utilizando o LDPC e TCC para proteger erros é uma boa opção, mas devem ser aplicados dependendo da modulação e taxa do codificador a utilizar. O código turbo produto, descrito no capítulo 5, também pode ser aplicado nesses sistemas com a vantagem de se poder controlar o número de iterações e assim maximizar o tempo de carga da bateria, isso para receptores de TV digital móveis.