

DECOMPOSIÇÕES DIÁDICAS NÃO-LINEARES USANDO OPERADORES
MORFOLÓGICOS

Marcello Artimos Neves

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

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

Prof. Marcos Craizer, D.Sc.

Prof. Sergio Lima Netto, Ph.D.

Prof. Ralph da Costa Teixeira, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2006

NEVES, MARCELLO ARTIMOS

Decomposições Diádicas Não-Lineares
Usando Operadores Morfológicos [Rio de
Janeiro] 2006

XII, 94 p. 29,7 cm (COPPE/UFRJ,
M.Sc., Engenharia Elétrica, 2006)

Dissertação - Universidade Federal
do Rio de Janeiro, COPPE

1. Decomposições Diádicas

2. Lifting

3. Projeto de Decomposições Wavelets
Não-Lineares

4. Operadores Morfológicos Auto-Duais

5. Compressão de Imagens

I. COPPE/UFRJ II. Título (série)

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

DECOMPOSIÇÕES DIÁDICAS NÃO-LINEARES USANDO OPERADORES
MORFOLÓGICOS

Marcello Artimos Neves

Março/2006

Orientador: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Originalmente, a Decomposição em Multiresolução do tipo Wavelet é uma ferramenta linear. Contudo, reconhece-se atualmente que extensões não-lineares são possíveis; a técnica *lifting* é uma ferramenta valiosa para a construção de decomposições wavelets genéricas, inclusive não-lineares. Neste trabalho procedemos à formalização matemática de um método recentemente proposto para projeto de decomposições wavelet não-lineares baseadas em passos de *lifting* e operadores morfológicos. Além disso, propomos um procedimento que permite otimizá-lo para uma dada aplicação. É investigada a aplicação, nos passos de *lifting* das decomposições, de operadores morfológicos auto-duais, os quais vêm sendo utilizados com sucesso em filtragem morfológica, e possuem características desejáveis para as decomposições em questão. As wavelets morfológicas assim criadas são aplicadas em imagens que são submetidas a um processo de compressão baseado em codificação de subbandas. O desempenho de compressão das decomposições usando os operadores auto-duais é avaliado e comparado com o obtido com decomposições lineares amplamente usadas na literatura.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

NON-LINEAR DIADIC DECOMPOSITIONS USING MORPHOLOGIC
OPERATORS

Marcello Artimos Neves

March/2006

Advisor: Eduardo Antônio Barros da Silva

Department: Electric Engineering

In their original form, Multiresolution Wavelet Decompositions are linear tools. Nevertheless, non-linear extensions are currently recognized to be possible, where decompositions in lifting steps play an important role. In this work, we proceed to a mathematical formalization of a non-linear wavelet decomposition design method recently proposed, based on lifting steps with morphological operators. Furthermore, we propose a procedure in order to optimize it for a given application. We investigate the use of self-dual morphological operators in the lifting steps. Self-dual operators have been used with success in morphological filtering applications, and possess several useful characteristics for the non-linear decompositions investigated. The generated morphological wavelets are used to decompose images which are applied to a compression process based on subband coding. The compression performance of self-dual operators is evaluated and compared with the ones of well known linear decompositions.

Sumário

1	Introdução	1
2	Sistema axiomático para decomposição diádica de sinais	3
2.1	Decomposição Wavelet Acoplada	4
2.2	Decomposição Wavelet Desacoplada	6
2.3	Exemplos	9
3	O Esquema <i>Lifting</i>	12
3.1	<i>Lifting</i> para Predição (<i>Prediction Lifting</i>)	13
3.2	<i>Lifting</i> para Atualização (<i>Update Lifting</i>)	15
3.3	<i>Lifting</i> para Predição e Atualização (<i>Prediction-Update Lifting</i>)	17
3.4	<i>Lifting</i> no caso linear	19
4	Método para Projeto de Wavelets Não-Lineares	25
4.1	Procedimento para otimização da solução	31
5	Noções Preliminares em Morfologia Matemática	32
5.1	Morfologia em Reticulados Completos	33
5.2	Morfologia em Inf-Semireticulados Completos	47

6	Resultados	58
6.1	Decomposições Bidimensionais e EZW	59
6.2	Testes	65
6.3	Operadores modificados e novos testes	77
7	Conclusão	89
	Referências Bibliográficas	93

Lista de Figuras

2.1	Um estágio de decomposição wavelet acoplada.	4
2.2	Decomposição wavelet acoplada em 3 níveis: (a) análise; (b) síntese.	7
2.3	Um estágio de uma decomposição wavelet desacoplada.	8
2.4	Etapa de síntese de uma decomposição wavelet desacoplada em 3 níveis.	9
3.1	Um passo de <i>Lifting</i> para Predição: análise e síntese.	14
3.2	Um passo de <i>Lifting</i> para Atualização: análise e síntese.	16
3.3	Um passo de <i>Lifting</i> para Predição e Atualização: análise e síntese.	18
3.4	Um estágio de decomposição wavelet linear: análise e síntese.	19
3.5	Fatoração de filtros lineares em passos de <i>lifting</i> : análise (acima) e síntese (abaixo).	20
4.1	Esquema de decomposição do método de projeto: análise (acima) e síntese (abaixo).	26
5.1	Imagem f : Lena.	41
5.2	Aplicação de α_A às linhas de f	42
5.3	Aplicação de β_A às linhas de f	43
5.4	(a) Sinal original x ; (b) $\alpha_A\{x\}$; (c) $\beta_A\{x\}$; (d) $\gamma_A\{x\}$	45

5.5	Exemplo de dois sinais $x, y \in \text{Fun}(E, \mathcal{T})$ em que $x \preceq y$. Fonte da figura: [1].	50
5.6	Ínfimo entre dois sinais: $\wedge x, y$. Fonte da figura: [1].	51
5.7	Exemplo de aplicação de erosão (lado esquerdo) e abertura (lado direito). Repare que a erosão cria um “platô” na região onde o sinal de entrada x cruza o eixo horizontal. Esse platô garante a existência da dilatação e, por conseguinte, da abertura. Fonte da figura: [1].	54
6.1	Etapas do método EZW de compressão de imagens.	60
6.2	Um nível de decomposição bidimensional gerada a partir de uma decomposição unidimensional. Da esquerda para a direita: sinal original, as duas subbandas intermediárias geradas a partir da decomposição das linhas da imagem original, as quatro subbandas finais geradas a partir de decomposição das colunas das subbandas intermediárias.	62
6.3	Disposição das quatro subbandas resultantes de um nível de decomposição wavelet de modo a formar um sinal com a mesma dimensão do sinal original.	63
6.4	Dois níveis de decomposição wavelet bidimensional.	65
6.5	Imagem original: Lena, 512x512, 8bpp.	66
6.6	Três níveis de decomposição wavelet em subbandas utilizando o projeto 1.	69
6.7	Imagem recuperada após processo de compressão / decompressão, onde foram usados o projeto 1 e o operador γ_A . Características: 3 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.	70
6.8	Imagem recuperada após processo de compressão / decompressão, com o uso do projeto 1 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.	71

6.9	Imagem recuperada com o uso do projeto 1 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 5, taxa de 0,15bpp.	72
6.10	Imagem recuperada com o uso do projeto 3 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.	74
6.11	Imagem recuperada com o uso do projeto 3 e operador α_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.	76
6.12	Imagem recuperada com o uso do projeto 3 e operador κ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.	77
6.13	Imagem recuperada com o uso do projeto 3 e operador α_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,50bpp.	78
6.14	Imagem recuperada com o uso do projeto 3 e operador κ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,50bpp.	79
6.15	Correspondência entre as posições dos sinais: (a) x_0 , (b) $x_{1,i}$ e (c) $y_{1,i}$ e numa decomposição wavelet. Note que $x_{1,i}$ e $y_{1,i}$ são espacialmente desalinhados.	79
6.16	Correspondência entre as posições dos sinais: (a) x_0 , (b) $x_{1,i}$, (c) $\sigma_{t_n}^\downarrow \{x_{1,i}\}$, (d) $\sigma^\uparrow \sigma_{t_n}^\downarrow \{x_{1,i}\}$ e (e) $y_{1,i}$	81
6.17	Correspondência entre as posições dos sinais: (a) x_0 , (b) $y_{1,i}$, (c) $\lambda \rho_{t_n}^\downarrow \{y_{1,i}\}$, (d) $\rho^\uparrow \lambda \rho_{t_n}^\downarrow \{y_{1,i}\}$ e (e) $x_{1,i}$	82
6.18	Imagem recuperada com o uso do projeto 3, operador γ_A^{mod} , EE com 3 u.c., 5 níveis de decomposição e taxa de 0,15bpp.	85

6.19	Imagem recuperada com o uso do projeto 3, operador γ_A^{mod} , EE com 3 u.c., 5 níveis de decomposição e taxa de 0,50bpp.	86
6.20	Imagem recuperada após decomposição morfológica e linear, ambas com 5 níveis, e taxa de quantização de 0,15bpp. (a) Decomposição linear resultante do uso da wavelet Daubechies 9-7. (b) Decomposição morfológica do projeto 3, EE de comprimento 3 e operador γ_A^{mod}	88

Lista de Tabelas

3.1	Coeficientes dos filtros de síntese da wavelet Daubechies 9-7.	22
5.1	Comparação do valor médio após aplicação de diversos operadores morfológicos ao sinal x da figura 5.4(a).	46
6.1	Decomposições wavelets projetadas utilizando o método do capítulo 4.	65
6.2	Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 1, onde a operação γ_A foi usada nos passos de <i>lifting</i> . São apresentados resultados para diversos níveis de decomposição, EE com diversas unidades de comprimento (u.c.) e taxa de quantização / codificação de 0.15bpp.	68
6.3	Projeto 3: coeficientes resultantes da fatoração da wavelet Daubechies 9-7.	73
6.4	Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3, onde a operação γ_A foi usada nos passos de <i>lifting</i> .	73
6.5	Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3. São apresentados resultados para diversos operadores morfológicos, comprimentos de EE e taxas de quantização / codificação.	75
6.6	Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3. São apresentados resultados para os operadores morfológicos modificados γ_A^{mod} e κ_A^{mod} , EE com 2 e 3 u.c. e taxas de quantização / codificação de 0,15bpp e 0,50bpp.	84

6.7	Comparação entre a decomposição linear obtida com a wavelet Daubechies 9-7 e a decomposição morfológica obtida com o projeto 3, onde o operador utilizado foi o $\mathcal{L}_B\{.\}$ com EE de comprimento 3. . .	87
-----	---	----

Capítulo 1

Introdução

Em sua forma original, a Transformada Wavelet é conhecida como uma ferramenta linear [2, 3]. No entanto, atualmente reconhece-se que extensões não-lineares são possíveis [4, 5]. A decomposição em passos de *lifting*, introduzida por Daubechies e Sweldens [5], é uma técnica valiosa para a construção de Transformadas Wavelets não-lineares.

Em [4], Cardoso apresentou um método para projeto de Transformadas Wavelets não-lineares, com a utilização de passos de *lifting* e operadores morfológicos [6–8]. Neste trabalho, começamos por formalizar tal método utilizando, para isto, a notação e o sistema axiomático desenvolvidos por Heijmans e Goutsias em [9, 10]. Em seguida, é proposta uma melhoria para este método, no sentido de otimizá-lo para uma dada aplicação. Além disso, operadores morfológicos auto-duais são testados e os resultados de sua utilização são confrontados com os obtidos por Cardoso e Artimos em [4] e [11].

Este trabalho é dividido como segue. O capítulo 2 apresenta sucintamente o sistema axiomático desenvolvido por Heijmans e Goutsias para decomposição de sinais em multiresolução do tipo wavelet [10](existem outros tipo de decomposição de sinais em multiresolução, onde a decomposição do tipo piramidal é um exemplo [9]), que contempla tanto as lineares quanto as não-lineares.

No capítulo 3 apresentamos a técnica *lifting*, peça chave para a construção de decomposições wavelets não-lineares. A filosofia desta técnica é encadear passos

de operações simples para a construção de decomposições complexas; com ela, qualquer que seja o operador escolhido em cada um desses passos, é sempre possível a reconstrução perfeita.

No capítulo 4, descrevemos um método para projeto de decomposições wavelet não-lineares utilizando passos de *lifting* com operadores morfológicos, com suporte dado pelo sistema axiomático apresentado no capítulo 2. Esse método é primeiramente descrito em sua versão original, como em [4, 11], para, então, serem incorporadas melhorias que permitem otimizá-lo para uma dada aplicação. Já que as características construtivas do método de projeto impõem que o comportamento freqüencial da wavelet não-linear gerada seja semelhante ao de algumas wavelets lineares [4, 11], utilizamos, para as melhorias citadas, ferramentas lineares de análise freqüencial.

Como em cada passo de *lifting* utilizamos operadores morfológicos, dedicamos o capítulo 5 para uma apresentação básica desses operadores. Uma ênfase especial é dada aos que possuem a característica de auto-dualidade, muito útil em filtragem morfológica de sinais. Os resultados obtidos com a utilização de operadores auto-duais são dispostos no capítulo 6, bem como as comparações com os obtidos por Artimos [11] e Cardoso [4].

Finalmente, o capítulo 7 traz conclusões e propostas para trabalhos futuros.

Capítulo 2

Sistema axiomático para decomposição diádica de sinais

Neste capítulo, apresentamos sucintamente o sistema axiomático desenvolvido por Goutsias e Heimans [10] que engloba todos os tipos de decomposição diádica de sinais.

Uma wavelet pode ser construída a partir de uma decomposição diádica recursiva; para isto, somente uma das duas bandas da decomposição diádica é novamente dividida em duas outras e assim por diante. Como neste trabalho não estamos interessados em quaisquer decomposições diádicas, só falaremos, a partir de agora, em decomposições wavelets.

A seção 2.1 introduz a definição de uma decomposição wavelet acoplada, onde a etapa de síntese é formada por um único operador (acoplado), cujos argumentos são os sinais de aproximação e de detalhe. Na seção 2.2, apresentamos a definição de uma wavelet desacoplada, que é um caso particular da anterior, onde o operador da etapa de síntese é constituído da adição de dois operadores (desacoplados) que agem, cada um deles, sobre um único argumento: o sinal de aproximação e o de detalhe, respectivamente. Uma apresentação mais detalhada dos conceitos presentes neste capítulo, seguida de inúmeros exemplos esclarecedores, pode ser encontrada em [9, 10].

2.1 Decomposição Wavelet Acoplada

Considere conjuntos (ou espaços) V_j e W_j , onde j pode variar sobre um conjunto finito ou infinito de índices. Aqui, à medida que j cresce, a quantidade de informação é reduzida, ou seja, o espaço V_j (W_j) é mais “rico” em informações que V_{j+1} (W_{j+1}). Nós nos referimos a V_j como o espaço de sinais de aproximação no nível j e W_j como o espaço de detalhes no nível j . Chamamos de “etapa de análise” uma decomposição de um sinal na direção de crescimento de j , por meio de operadores de análise de sinais $\psi_j^\uparrow : V_j \rightarrow V_{j+1}$ e operadores de análise de detalhes $\omega_j^\uparrow : V_j \rightarrow W_{j+1}$. Já que os operadores ψ_j^\uparrow e ω_j^\uparrow são projetados para reduzir a quantidade de informação do sinal sobre o qual eles são aplicados, eles geralmente não são inversíveis. Por outro lado, a “etapa de síntese” procede na direção de decréscimo de j por meio de operadores de síntese $\Psi_j^\downarrow : V_{j+1} \times W_{j+1} \rightarrow V_j$. Isto está ilustrado na figura 2.1.

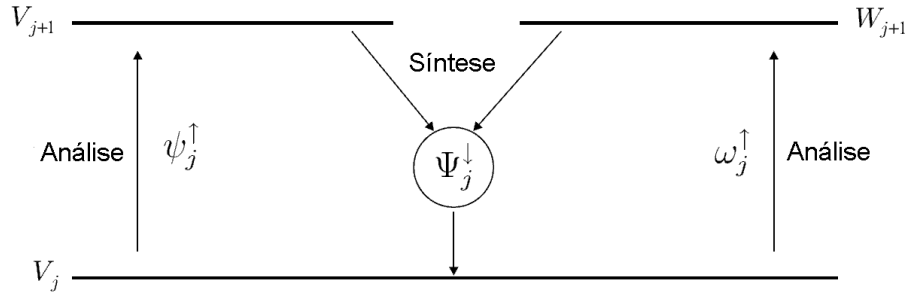


Figura 2.1: Um estágio de decomposição wavelet acoplada.

Para que o esquema de decomposição anterior possa gerar uma representação completa de sinais, no sentido de que os mapeamentos $(\psi_j^\uparrow, \omega_j^\uparrow) : V_j \rightarrow V_{j+1} \times W_{j+1}$ e $\Psi_j^\downarrow : V_{j+1} \times W_{j+1} \rightarrow V_j$ sejam inversos um do outro, é necessário que as seguintes condições sejam satisfeitas:

$$\Psi_j^\downarrow\{(\psi_j^\uparrow\{x\}, \omega_j^\uparrow\{x\})\} = x, \forall x \in V_j, \quad (2.1)$$

que é chamada de **condição de reconstrução perfeita**, e

$$\begin{cases} \psi_j^\uparrow \Psi_j^\downarrow \{(x, y)\} = x, \forall x \in V_{j+1} \text{ e } y \in W_{j+1} \\ \omega_j^\uparrow \Psi_j^\downarrow \{(x, y)\} = y, \forall x \in V_{j+1} \text{ e } y \in W_{j+1} \end{cases} \quad (2.2)$$

Defina agora o mapeamento $\Psi_j^\uparrow : V_j \rightarrow V_{j+1} \times W_{j+1}$ como na equação 2.3 em seguida:

$$\Psi_j^\uparrow \{x\} = (\psi_j^\uparrow \{x\}, \omega_j^\uparrow \{x\}) \quad (2.3)$$

As duas condições acima descritas nas equações 2.1 e 2.2 garantem que a decomposição seja não-redundante.

A condição 2.1 implica que o mapeamento $\Psi_j^\uparrow : V_j \rightarrow V_{j+1} \times W_{j+1}$, definido em 2.3, seja uma operação injetora (cada elemento do conjunto contradomínio $V_{j+1} \times W_{j+1}$ se relaciona com no máximo um elemento do conjunto domínio V_j), enquanto Ψ_j^\downarrow seja sobrejetora. Para demonstrar, considere x_1 e $x_2 \in V_j$, tais que $\Psi_j^\uparrow \{x_1\} = \Psi_j^\uparrow \{x_2\}$; pela condição 2.1, tem-se que $\Psi_j^\downarrow \{\Psi_j^\uparrow \{x_1\}\} = x_1$ e $\Psi_j^\downarrow \{\Psi_j^\uparrow \{x_2\}\} = x_2$. Como $\Psi_j^\uparrow \{x_1\} = \Psi_j^\uparrow \{x_2\}$, conclui-se que $x_1 = x_2$. Logo Ψ_j^\uparrow deve ser injetora. Para mostrar que Ψ_j^\downarrow é sobrejetora, basta perceber que todo elemento x pertencente ao contradomínio V_j é relacionado a algum elemento do domínio V_j , fazendo com que o conjunto imagem seja igual ao conjunto contradomínio. Logo Ψ_j^\downarrow é uma operação sobrejetora.

Por outro lado, a condição 2.2 implica que Ψ_j^\uparrow seja sobrejetora e que Ψ_j^\downarrow seja injetora. A demonstração é análoga ao caso anterior.

Além disso, pode-se mostrar que, se a condição 2.1 é assegurada e Ψ_j^\uparrow é sobrejetora (ou Ψ_j^\downarrow é injetora), então 2.2 é também assegurada. Finalmente, se a condição

Em outras palavras, o fato da decomposição ser não-redundante significa que os sinais $x \in V_{j+1}$ e $y \in W_{j+1}$ não possuem interseção, embora sejam suficientes para reconstruir, a partir de Ψ_j^\uparrow , o sinal no nível j que os gerou.

Agora, dado um sinal $x_0 \in V_0$, considere o seguinte esquema recursivo de análise:

$$x_0 \rightarrow \{x_1, y_1\} \rightarrow \{x_2, y_2, y_1\} \rightarrow \cdots \rightarrow \{x_k, y_k, y_{k-1}, \dots, y_1\} \rightarrow \cdots \quad (2.4)$$

onde cada etapa de análise é dada por

$$\begin{cases} x_{j+1} = \psi_j^\uparrow \{x_j\} \in V_{j+1}, j \geq 0 \\ y_{j+1} = \omega_j^\uparrow \{x_j\} \in W_{j+1}, j \geq 0 \end{cases} \quad (2.5)$$

É fácil perceber que o sinal original x_0 pode ser perfeitamente reconstruído a partir de x_k e y_1, y_2, \dots, y_k por meio do seguinte esquema recursivo de síntese:

$$x_j = \Psi_j^\downarrow \{(x_{j+1}, y_{j+1})\}, j = k-1, k-2, \dots, 0, \quad (2.6)$$

o que mostra que a decomposição dada por 2.4 e 2.5 é inversível. O esquema completo de representação de sinais governado pelas equações 2.1 a 2.6 é chamado de **decomposição wavelet acoplada**. Diagramas de blocos ilustrando este esquema para o caso em que $k = 3$, ou seja, em 3 níveis de decomposição, são mostrados na figura 2.2.

2.2 Decomposição Wavelet Desacoplada

Um caso particular de interesse é quando existe uma operação binária $\dot{+}$ em V_j , que chamaremos de “adição” (note que $\dot{+}$ pode depender de j , caso em que precisaríamos escrever $\dot{+}_j$), e operadores $\psi_j^\downarrow : V_{j+1} \rightarrow V_j$ e $\omega_j^\downarrow : W_{j+1} \rightarrow V_j$ tais que:

$$\Psi_j^\downarrow \{(x, y)\} = \psi_j^\downarrow \{x\} \dot{+} \omega_j^\downarrow \{y\}, x \in V_{j+1} \text{ e } y \in W_{j+1} \quad (2.7)$$

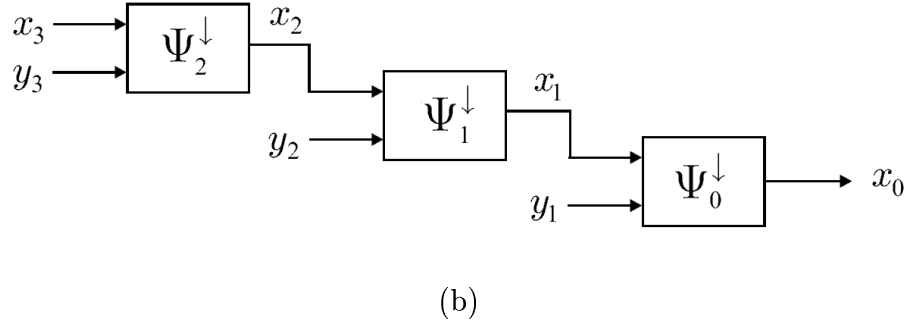
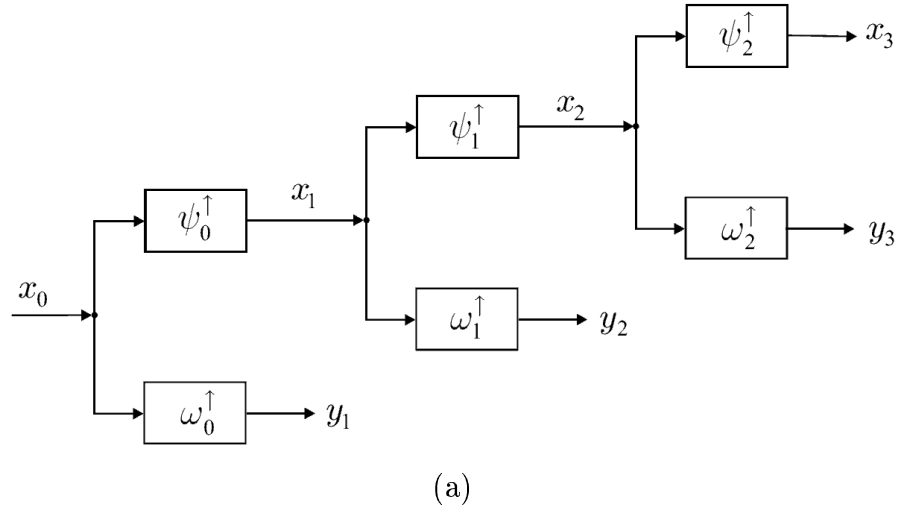


Figura 2.2: Decomposição wavelet acoplada em 3 níveis: (a) análise; (b) síntese.

Nos referimos a ψ_j^\downarrow como operador de síntese de sinais de aproximação e a ω_j^\downarrow como operador de síntese de detalhes, respectivamente. Assim sendo, as condições 2.1 e 2.2 podem ser reescritas como:

$$\psi_j^\downarrow \psi_j^\uparrow \{x\} + \omega_j^\downarrow \omega_j^\uparrow \{x\} = x, \text{ se } x \in V_j \quad (2.8)$$

$$\psi_j^\uparrow \{ \psi_j^\downarrow \{x\} + \omega_j^\downarrow \{y\} \} = x, \text{ se } x \in V_{j+1} \text{ e } y \in W_{j+1} \quad (2.9)$$

$$\omega_j^\uparrow \{ \psi_j^\downarrow \{x\} + \omega_j^\downarrow \{y\} \} = y, \text{ se } x \in V_{j+1} \text{ e } y \in W_{j+1} \quad (2.10)$$

Nos referimos ao esquema para representação de sinais governado pelas equações 2.8, 2.9 e 2.10 como **decomposição wavelet desacoplada**. Um estágio deste esquema de decomposição é ilustrado na figura 2.3.

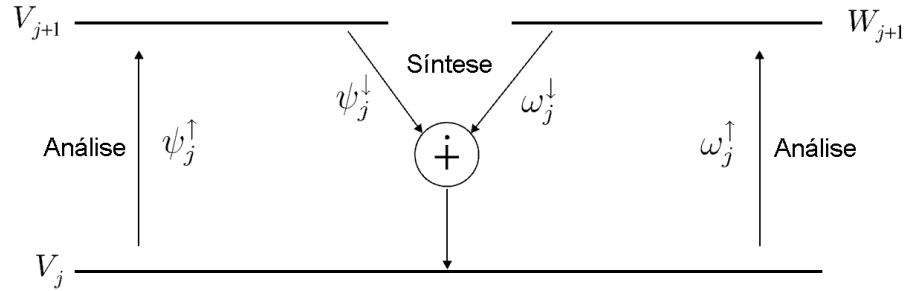


Figura 2.3: Um estágio de uma decomposição wavelet desacoplada.

Agora, para um sinal $x_0 \in V_0$ e o esquema recursivo de análise apresentado em 2.4 e 2.5, x_0 pode ser perfeitamente reconstruído a partir de x_k e y_1, y_2, \dots, y_k por meio do seguinte esquema recursivo de síntese:

$$x_j = \psi_j^\downarrow \{x_{j+1}\} + \omega_j^\downarrow \{y_{j+1}\}, \quad j = k-1, k-2, \dots, 0. \quad (2.11)$$

Assim como anteriormente, um sinal x_j no nível j é reconstruído utilizando-se informação que só está disponível no nível $j+1$. Primeiramente, o sinal x_{j+1} é mapeado através da operação ψ_j^\downarrow de volta ao nível j , gerando uma aproximação $\hat{x}_j = \psi_j^\downarrow \{x_{j+1}\}$ de x_j ; de forma análoga, o sinal de detalhe y_{j+1} é mapeado de volta ao nível j através da operação ω_j^\downarrow , gerando o sinal de detalhe $e_j = \omega_j^\downarrow \{y_{j+1}\}$ no nível j ; finalmente, os resultados são combinados por meio do operador de adição $+$.

A equação 2.7 diz respeito somente à estrutura de uma etapa de síntese. Um diagrama de blocos ilustrando essa etapa, para o caso $k=3$, é mostrado na figura 2.4; a etapa de análise é a mesma da figura 2.2(a).

Vejamos alguns exemplos na seção que segue.

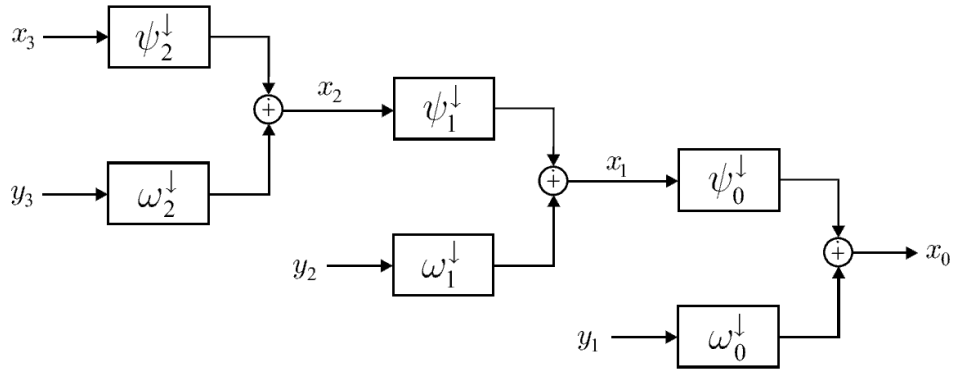


Figura 2.4: Etapa de síntese de uma decomposição wavelet desacoplada em 3 níveis.

2.3 Exemplos

Nos exemplos que seguem, consideramos somente um nível de decomposição, isto é, decomposições envolvendo somente os conjuntos V_0, V_1, W_1 . Por simplicidade, suprimimos os subíndices $j = 0$ nos operadores de análise e síntese.

Exemplo 1 Neste exemplo, apresentamos a decomposição em componentes polifásicas de ordem 2, também denominada *lazy wavelet*, muito utilizada na comunidade de processamento de sinais.

A decomposição em componentes polifásicas, ou *lazy wavelet*, é utilizada no capítulo 4, onde descrevemos um método para decomposições diádicas de sinais.

Na *lazy wavelet*, a operação de análise ψ^\uparrow deve reter os elementos do sinal de entrada x situados em posições pares, enquanto a operação ω^\uparrow deve reter os situados em posições ímpares. Assim, temos:

$$\psi^\uparrow\{x\}(n) = x(2n), \quad n \in \mathbb{Z} \quad (2.12)$$

$$\omega^\uparrow\{x\}(n) = x(2n + 1), \quad n \in \mathbb{Z} \quad (2.13)$$

Os operadores de síntese ψ^\downarrow e ω^\downarrow são tais que, quando adicionados por meio da operação $\dot{+}$, recuperam o sinal original. Assim, sendo $\dot{+}$ a adição padrão, temos:

$$\begin{cases} \psi^\downarrow\{x\}(2n) = x(n) \\ \psi^\downarrow\{x\}(2n+1) = 0 \end{cases} \quad (2.14)$$

$$\begin{cases} \omega^\downarrow\{y\}(2n) = 0 \\ \omega^\downarrow\{y\}(2n+1) = y(n) \end{cases} \quad (2.15)$$

Exemplo 2 Vejamos um exemplo numérico de decomposição em componentes polifásicas.

Seja $x_0(n) = n$, ou seja, $x_0 = [0; 1; 2; 3; 4; 5; \dots]$, para $n \geq 0$. Pela equação 2.12 encontra-se:

$$x_1 = \psi^\uparrow\{x_0\}(n) = x_0(2n) = 2n, \text{ ou seja, } x_1 = [0; 2; 4; \dots], \text{ para } n \geq 0.$$

Da mesma forma, pela equação 2.13, encontra-se:

$$y_1(n) = \omega^\uparrow\{x_0\}(n) = x_0(2n+1) = 2n+1, \text{ ou seja } y_1 = [1; 3; 5; \dots], \text{ para } n \geq 0.$$

Na síntese, pela equação 2.14, tem-se:

$$\psi^\downarrow\{x_1\}(2n) = x_1(n) \text{ e } \psi^\downarrow\{x_1\}(2n+1) = 0, \text{ ou seja,}$$

$$\psi^\downarrow\{x_1\} = [0; 0; 2; 0; 4; 0; \dots], \text{ } n \geq 0.$$

Pela equação 2.15, chega-se a:

$$\omega^\downarrow\{y_1\}(2n) = 0 \text{ e } \psi^\downarrow\{y_1\}(2n + 1) = y_1(n) , \text{ ou seja,}$$

$$\omega^\downarrow\{y_1\} = [0; 1; 0; 3; 0; 5; 0; \dots], n \geq 0.$$

Finalmente, com o operador de síntese $\Psi^\downarrow\{(x_1, y_1)\}$ dado pela adição padrão dos operadores parciais $\psi^\downarrow\{x_1\}$ e $\omega^\downarrow\{y_1\}$, consegue-se a reconstrução perfeita:

$$\begin{aligned} \Psi^\downarrow\{(x_1, y_1)\} &= \psi^\downarrow\{x_1\} + \omega^\downarrow\{y_1\} = \\ &= [0; 0; 2; 0; 4; 0; \dots] + [0; 1; 0; 3; 0; 5; 0; \dots] = \\ &= [0; 1; 2; 3; 4; 5; \dots] = \\ &= x_0 \end{aligned} \tag{2.16}$$

Capítulo 3

O Esquema *Lifting*

Uma importante proposta de Sweldens para construção, de forma simples e flexível, de novas decomposições wavelet, possivelmente não-lineares, a partir de outras já existentes foi a técnica denominada *lifting* [5]. Ela compreende uma dada transformada wavelet seguida por um número arbitrário de dois tipos de passos de operações simples: o passo de predição e o de atualização, a que chamamos passos de *lifting*. No de atualização, o sinal de aproximação da wavelet original é modificado a partir de informação da banda de detalhes; no outro tipo, o sinal de detalhes é modificado a partir de informação contida na banda de aproximação. Esses passos podem ser invertidos e, por isso, a propriedade de reconstrução perfeita é garantida.

Assim, com a técnica *lifting*, é possível modificar os operadores de análise e síntese de modo a construir wavelets com “melhores características” que a original; aqui, a expressão “melhores características” pode ser interpretada de diferentes maneiras, dependendo da aplicação. Por exemplo, no caso de aplicação em compressão de imagens, pode ser requerido que o sinal de detalhes tenha energia reduzida, então podemos modificar o operador de síntese no sentido de que sua saída tenha baixa energia.

O esquema *lifting* é uma ferramenta poderosa e genérica, apesar de sua simplicidade; uma de suas qualidades, que é de especial interesse para este trabalho, é a possibilidade de construir decomposições wavelets não-lineares.

Como dissemos, em seu trabalho, Sweldens apresentou duas modalidades de

passos *lifting*:

- *prediction lifting* ou *lifting* para predição, que na análise modifica o operador ω^\uparrow e na síntese modifica o operador Ψ^\downarrow , no caso acoplado, ou ψ^\downarrow , no caso desacoplado;
- *update lifting* ou *lifting* para atualização, que na análise modifica o operador ψ^\uparrow e na síntese modifica o operador Ψ^\downarrow , no caso acoplado, ou ω^\downarrow , no caso desacoplado.

Tratamos essas duas modalidades separadamente nas próximas seções.

3.1 *Lifting* para Predição (*Prediction Lifting*)

Considere um nível de uma decomposição wavelet governada pelos operadores de análise $\psi^\uparrow : V_0 \rightarrow V_1$ e $\omega^\uparrow : V_0 \rightarrow W_1$ e o operador de síntese $\Psi^\downarrow : V_1 \times W_1 \rightarrow V_0$ (suprimimos o subíndice $j = 0$ por conveniência), tais que as condições 2.1 e 2.2 são satisfeitas. Considere também um operador de predição qualquer $\pi : V_1 \rightarrow W_1$. Sendo assim, um passo de ***lifting* para predição** é definido como:

$$y'_1 = y_1 - \pi\{x_1\} \quad (3.1)$$

onde $x_1 = \psi^\uparrow\{x_0\}$, $y_1 = \omega^\uparrow\{x_0\}$, $x_0 \in V_0$ e y'_1 é o novo sinal de detalhes após o passo de *lifting* para predição.

Esta definição leva à etapa de análise ilustrada na figura 3.1.

É fácil perceber que o sinal x_0 pode ser perfeitamente reconstruído a partir de x_1 e y'_1 , uma vez que:

$$\Psi^\downarrow\{(x_1, y'_1 + \pi\{x_1\})\} = \Psi^\downarrow\{(x_1, y_1)\} = x_0 \quad (3.2)$$

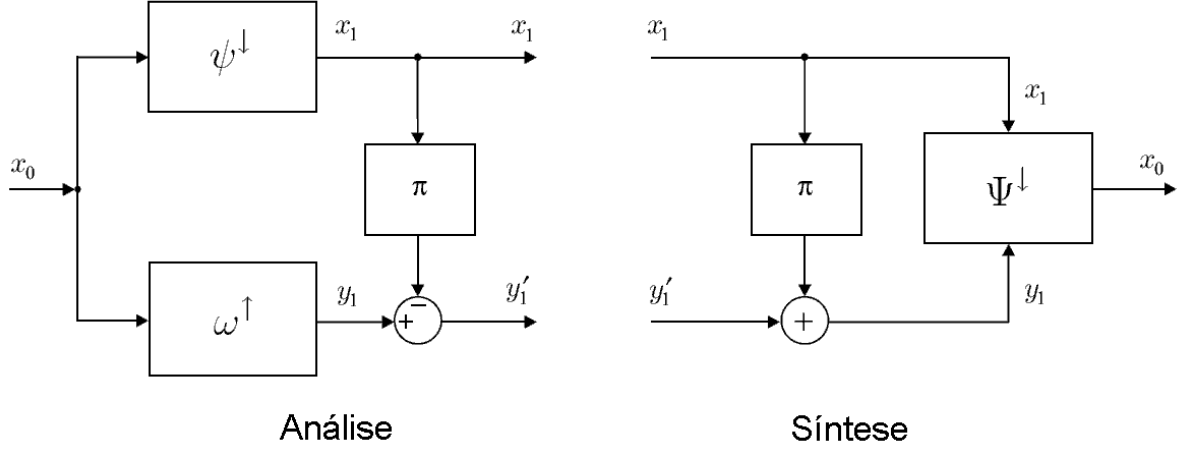


Figura 3.1: Um passo de *Lifting* para Predição: análise e síntese.

Isto leva à etapa de síntese ilustrada na figura 3.1.

Com a introdução do passo de *lifting* para predição, os novos operadores de análise ψ_p^\uparrow e ω_p^\uparrow e síntese Ψ_p^\downarrow são dados por:

$$\psi_p^\uparrow\{x\} = \psi^\uparrow\{x\}, x \in V_0 \quad (3.3)$$

$$\omega_p^\uparrow\{x\} = \omega^\uparrow\{x\} - \pi\psi^\uparrow\{x\}, x \in V_0 \quad (3.4)$$

$$\Psi_p^\downarrow\{(x, y)\} = \Psi^\downarrow\{(x, y + \pi\{x\})\}, x \in V_1 \text{ e } y \in W_1 \quad (3.5)$$

Para mostrar que o esquema formado após a introdução do operador de *lifting* continua sendo uma decomposição wavelet, devemos verificar que os operadores ψ_p^\uparrow , ω_p^\uparrow e Ψ_p^\downarrow satisfazem as condições 2.1 e 2.2. Para $x \in V_0$, temos (os superíndices entre parênteses acima dos sinais de igualdade são as equações usadas para o desenvolvimento da demonstração):

$$\begin{aligned}
\Psi_p^\downarrow\{(\psi_p^\uparrow\{x\}, \omega_p^\uparrow\{x\})\} & \stackrel{(3.5)}{=} \Psi^\downarrow\{(\psi_p^\uparrow\{x\}, \omega_p^\uparrow\{x\} + \pi\psi_p^\uparrow\{x\})\} = \\
& \stackrel{(3.3 \text{ e } 3.4)}{=} \Psi^\downarrow\{(\psi^\uparrow\{x\}, \omega^\uparrow\{x\} - \pi\psi^\uparrow\{x\} + \pi\psi^\uparrow\{x\})\} = \\
& = \Psi^\downarrow\{(\psi^\uparrow\{x\}, \omega^\uparrow\{x\})\} = \\
& \stackrel{(2.1)}{=} x
\end{aligned} \tag{3.6}$$

Isto mostra que os operadores ψ_p^\uparrow , ω_p^\uparrow e Ψ_p^\downarrow satisfazem a condição 2.1. Além disso, com $x \in V_1$ e $y \in W_1$ temos:

$$\begin{aligned}
\psi_p^\uparrow\{\Psi_p^\downarrow\{(x, y)\}\} & \stackrel{(3.3)}{=} \psi^\uparrow\{\Psi_p^\downarrow\{(x, y)\}\} = \\
& \stackrel{(3.5)}{=} \psi^\uparrow\{\Psi^\downarrow\{(x, y + \pi x)\}\} = \\
& \stackrel{(2.2)}{=} x
\end{aligned} \tag{3.7}$$

Finalmente, temos:

$$\begin{aligned}
\omega_p^\uparrow\{\Psi_p^\downarrow\{(x, y)\}\} & \stackrel{(3.4)}{=} \omega^\uparrow\Psi_p^\downarrow\{(x, y)\} - \pi\psi^\uparrow\Psi_p^\downarrow\{(x, y)\} = \\
& \stackrel{(3.5)}{=} \omega^\uparrow\Psi_p^\downarrow\{(x, y + \pi\{x\})\} - \pi\psi^\uparrow\Psi_p^\downarrow\{(x, y)\} = \\
& \stackrel{(2.2)}{=} y + \pi\{x\} - \pi\psi^\uparrow\Psi_p^\downarrow\{(x, y)\} = \\
& \stackrel{(3.7)}{=} y + \pi\{x\} - \pi\{x\} = \\
& = y
\end{aligned} \tag{3.8}$$

As igualdades 3.7 e 3.8 mostram que os novos operadores de análise satisfazem a condição 2.2, o que conclui a demonstração.

3.2 *Lifting* para Atualização (*Update Lifting*)

Analogamente ao que fizemos na seção anterior, podemos modificar o sinal de aproximação $x_1 \in V_1$ gerando o sinal atualizado (*updated*) $x'_1 \in V_1$ utilizando

informação contida em $y_1 \in W_1$. Podemos realizar tal tarefa criando um operador $\lambda : W_1 \rightarrow V_1$ e definindo a operação de *lifting* para atualização como segue:

$$x'_1 = x_1 - \lambda\{y_1\} \quad (3.9)$$

onde $x_1 = \psi^\uparrow\{x_0\}$, $y_1 = \omega^\uparrow\{x_0\}$, $x_0 \in V_0$ e x'_1 é o novo sinal de aproximação após o passo de *lifting* para atualização.

O passo de atualização dado por 3.9 é ilustrado na etapa de análise da figura 3.2.

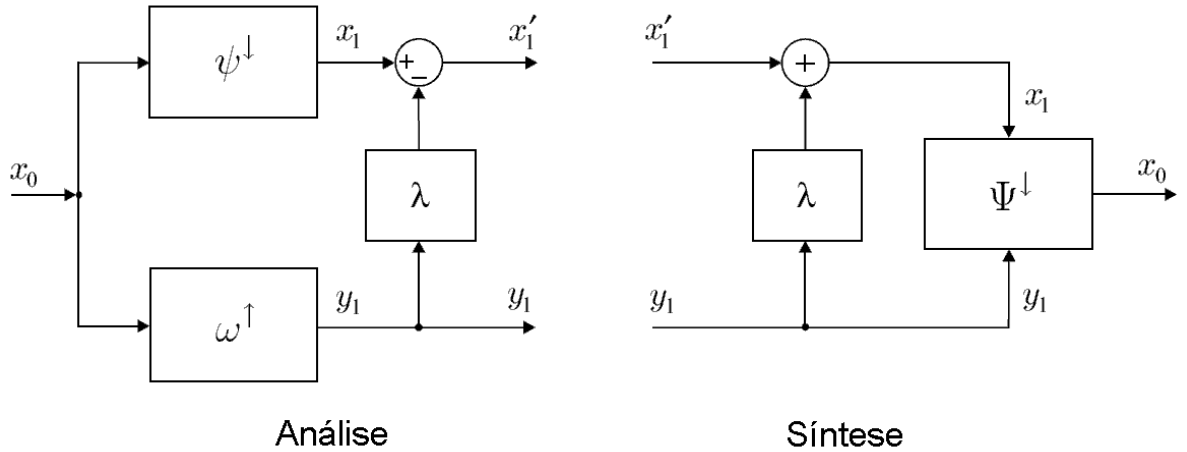


Figura 3.2: Um passo de *Lifting* para Atualização: análise e síntese.

É evidente que a reconstrução perfeita é possível; basta que a modificação causada pela introdução do passo de *lifting* seja desfeita:

$$\begin{aligned} x_1 &= x'_1 + \lambda\{y_1\} \\ x_0 &= \Psi^\downarrow\{(x_1, y_1)\} = \Psi^\downarrow\{(x'_1 + \lambda\{y_1\}, y_1)\} \end{aligned} \quad (3.10)$$

Isto leva à etapa de síntese ilustrada na figura 3.1.

Os novos operadores de análise e síntese são dados por:

$$\psi_u^\uparrow\{x\} = \psi^\uparrow\{x\} - \lambda\omega^\uparrow\{x\}, x \in V_0 \quad (3.11)$$

$$\omega_u^\uparrow\{x\} = \omega^\uparrow\{x\}, x \in V_0 \quad (3.12)$$

$$\Psi_u^\downarrow\{(x, y)\} = \Psi^\downarrow\{(x + \lambda\{y\}, y)\}, x \in V_1 \text{ e } y \in W_1 \quad (3.13)$$

Da mesma forma que no caso do *lifting* para predição, pode-se mostrar que os operadores ψ_u^\uparrow , ω_u^\uparrow e Ψ_u^\downarrow satisfazem as condições 2.1 e 2.2, definindo, então, um esquema de decomposição wavelet.

3.3 *Lifting* para Predição e Atualização (*Prediction-Update Lifting*)

Um passo completo de *lifting* é formado pelo operador de predição π seguido pelo de atualização λ ; esta composição é dada por:

$$\begin{aligned} y'_1 &= y_1 - \pi\{x_1\} \\ x'_1 &= x_1 - \lambda\{y'_1\} \end{aligned} \quad (3.14)$$

e é ilustrada na figura 3.3

A reconstrução perfeita de x_0 a partir de x'_1 e y'_1 é conseguida como segue:

$$\begin{aligned} x_1 &= x'_1 + \lambda\{y'_1\} \\ y_1 &= y'_1 + \pi\{x_1\} = y'_1 + \pi\{x'_1 + \lambda\{y'_1\}\} \\ x_0 &= \Psi^\downarrow\{(x_1, y_1)\} = \Psi^\downarrow\{x'_1 + \lambda\{y'_1\}, y'_1 + \pi\{x'_1 + \lambda\{y'_1\}\}\} \end{aligned} \quad (3.15)$$

Esta composição leva aos operadores modificados de análise e síntese, ψ_{pu}^\uparrow , ω_{pu}^\uparrow e Ψ_{pu}^\downarrow , como nas equações que seguem:

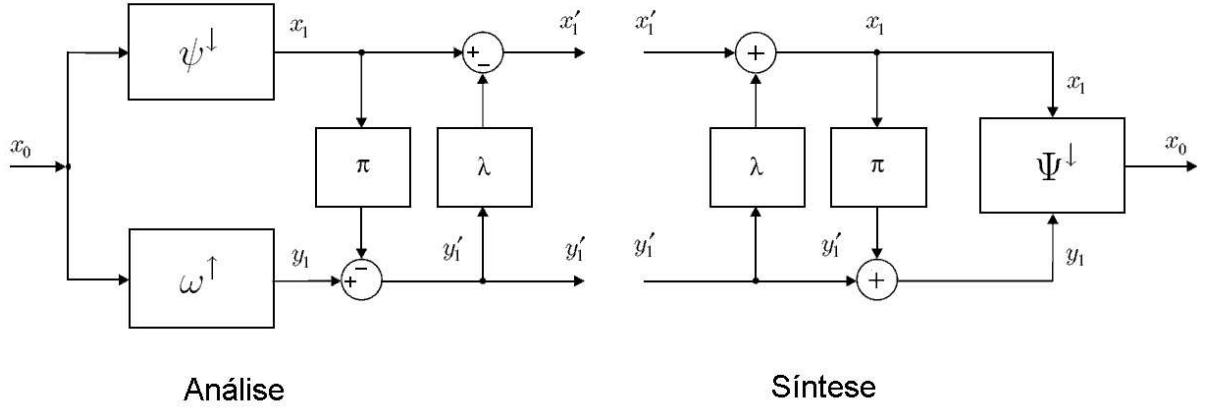


Figura 3.3: Um passo de *Lifting* para Predição e Atualização: análise e síntese.

$$\psi_{pu}^{\uparrow}\{x\} = \psi^{\uparrow}\{x\} - \lambda\{\omega^{\uparrow}\{x\} - \pi\psi^{\uparrow}\{x\}\}, x \in V_0 \quad (3.16)$$

$$\omega_{pu}^{\uparrow}\{x\} = \omega^{\uparrow}\{x\} - \pi\psi^{\uparrow}\{x\}, x \in V_0 \quad (3.17)$$

$$\Psi_{pu}^{\downarrow}\{(x, y)\} = \Psi^{\downarrow}\{(x + \lambda\{y\}, y + \pi\{x + \lambda\{y\}\})\}, x \in V_1 \text{ e } y \in W_1 \quad (3.18)$$

Para mostrar que este esquema define uma decomposição wavelet, basta verificar que os operadores ψ_{pu}^{\uparrow} , ω_{pu}^{\uparrow} e Ψ_{pu}^{\downarrow} satisfazem as condições 2.1 e 2.2.

Também é possível construir um esquema formado por um passo de atualização, com operador $\lambda : V_1 \rightarrow W_1$, seguido por um passo de predição, com $\pi : W_1 \rightarrow V_1$, sendo denominado *lifting* para atualização e predição (*update-prediction lifting*).

Até o momento, mostramos como podemos criar operadores ψ_{pu}^{\uparrow} , ω_{pu}^{\uparrow} e Ψ_{pu}^{\downarrow} utilizando, no máximo, um passo completo de *lifting*, mas note que podemos inserir tantos passos quanto forem necessários para que a decomposição apresente as características requeridas a uma dada aplicação.

É importante salientar que para que seja garantida a propriedade de reconstrução perfeita da wavelet original, em todos os esquemas apresentados nas seções 3.1, 3.2 e 3.3, não é feita nenhuma suposição acerca dos operadores π e λ , ou seja, eles podem ser quaisquer. Por isso, a ferramenta *lifting* é usada neste trabalho como subsídio para construção de wavelets não-lineares.

3.4 *Lifting* no caso linear

A figura 3.4 mostra um nível de uma decomposição wavelet linear, contemplando tanto a etapa de análise como a de síntese. Na análise, $\tilde{h}(n)$ é passa-baixa e $\tilde{g}(n)$ é passa-alta. O símbolo de subamostragem $\downarrow 2$ é equivalente ao operador ψ^\uparrow dado pela equação 2.12. Na síntese, o símbolo de superamostragem $\uparrow 2$ é equivalente ao operador ψ^\downarrow dado pela equação 2.14. Além disso, os filtros $h(n)$ e $g(n)$ são, respectivamente, passa-baixa e passa-alta.

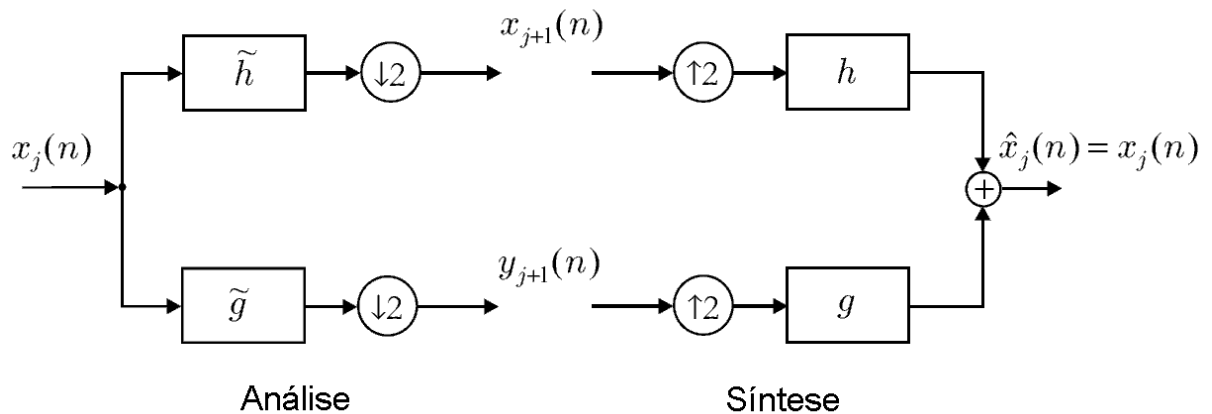


Figura 3.4: Um estágio de decomposição wavelet linear: análise e síntese.

Em se tratando de wavelets lineares, caso os filtros de síntese das bandas de aproximação e de detalhes satisfaçam a condição de complementariedade [3, 5], Daubechies e Sweldens mostraram que é possível fatorá-los em passos de *lifting* utilizando o algoritmo de Euclides para fatoração de polinômios de Laurent [5].

Em todos os casos de filtros lineares, a fatoração em passos de *lifting* resulta,

na análise, (a) numa decomposição em componentes polifásicas (ou *lazy wavelet*) descrita pelas equações 2.12 e 2.13 (ver exemplo 1 da seção 2.3), (b) seguida de operações $\pi_i : V_i \rightarrow W_i$ e $\lambda_i : W_i \rightarrow V_i$, com $i \in \{1, 2, \dots, I\}$, onde I depende da ordem dos filtros lineares a serem fatorados e, por fim, (c) a atuação de uma constante multiplicativa K no resultado da banda de aproximação e de uma constante $\frac{1}{K}$ no resultado da banda de detalhes. A síntese é obtida de forma direta, através da aplicação dos passos de *lifting* invertidos, como visto nas seções anteriores. A figura 3.5 ilustra essa fatoração, onde denotamos por $\psi_{\text{LW}}^\uparrow$, $\omega_{\text{LW}}^\uparrow$, $\psi_{\text{LW}}^\downarrow$ e $\omega_{\text{LW}}^\downarrow$ os operadores de análise e síntese da *lazy wavelet*, definidos respectivamente pelas equações 2.12, 2.13, 2.14 e 2.15.

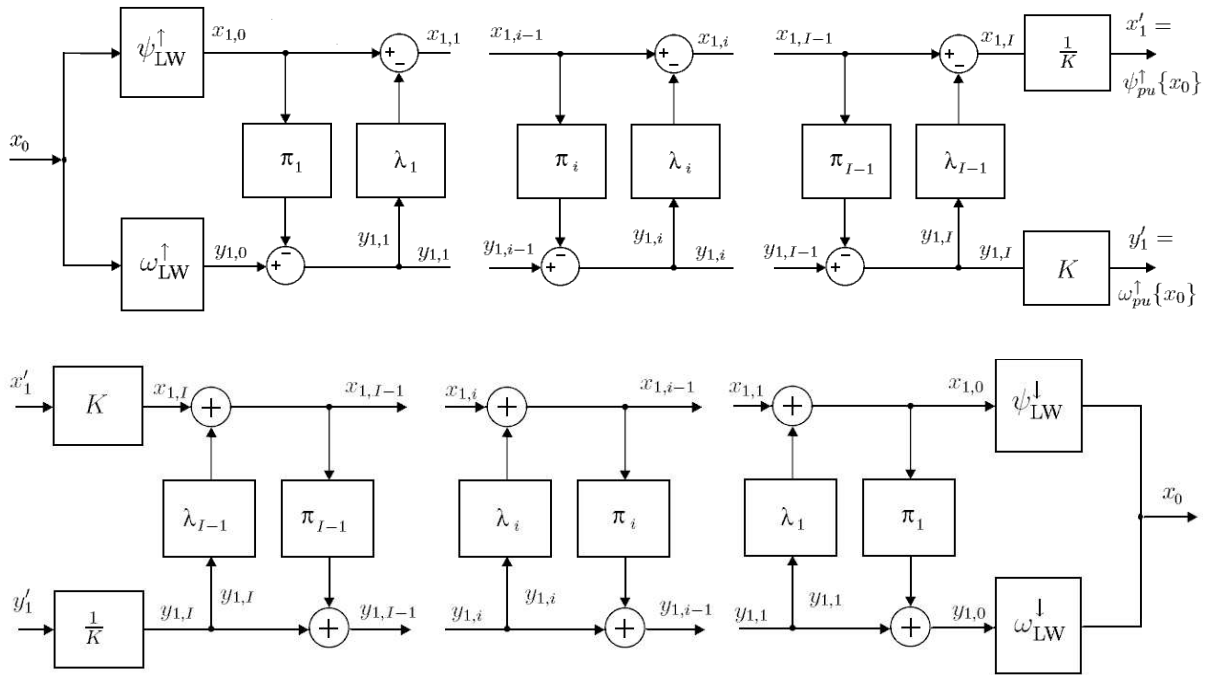


Figura 3.5: Fatoração de filtros lineares em passos de *lifting*: análise (acima) e síntese (abaixo).

Em seguida, vemos dois exemplos do resultado desta fatoração quando aplicados à wavelet Haar e à Daubechies 9-7.

Exemplo 1 (Wavelet Haar) Primeiramente, consideramos a decomposição wavelet de Haar, em que temos os seguintes filtros de síntese:

$$\begin{aligned}
H(z) &= 1 + z^{-1} \\
G(z) &= -\frac{1}{2} + \frac{1}{2}z^{-1}
\end{aligned} \tag{3.19}$$

A fatoração resulta nas operações de atualização e predição conforme 3.20:

$$\begin{aligned}
\pi_1 &= 0 \\
\lambda_1 &= 1 \\
\pi_2 &= -1/2 \\
\lambda_2 &= 0 \\
K &= 1
\end{aligned} \tag{3.20}$$

ou seja,

$$\begin{aligned}
x_{1,0} &= \psi^\uparrow\{x_0\} \\
y_{1,0} &= \omega^\uparrow\{x_0\} \\
x_{1,1} &= x_{1,0} - \pi_1\{y_{1,0}\} = x_{1,0} \\
y_{1,1} &= y_{1,0} - \lambda_1\{x_{1,1}\} = y_{1,0} - x_{1,1} = y_{1,0} - x_{1,0} \\
x_{1,2} &= x_{1,1} - \pi_2\{y_{1,1}\} = x_{1,1} + \frac{1}{2}y_{1,1} = x_{1,0} + \frac{1}{2}y_{1,0} - \frac{1}{2}x_{1,0} = \frac{1}{2}x_{1,0} + \frac{1}{2}y_{1,0} \\
y_{1,2} &= y_{1,1} - \lambda_2\{x_{1,2}\} \\
x'_1 &= x_{1,2} \\
y'_1 &= y_{1,2}
\end{aligned} \tag{3.21}$$

onde

- x_0 é o sinal de entrada a ser decomposto;
- ψ^\uparrow e ω^\uparrow são dados pelas equações 2.12 e 2.13, respectivamente, isto é, definem uma decomposição em componentes polifásicas;
- $x_{1,i}$ é um sinal que pertence ao espaço de sinais de aproximação V_1 (primeiro nível de decomposição wavelet) e já sofreu i atualizações pelos operadores do tipo λ (i -ésimo passo de *lifting*);

- $y_{1,i}$ é um sinal que pertence ao espaço de sinais de detalhes W_1 e já sofreu i modificações com operadores π de *lifting* para predição;
- x'_1 e y'_1 são as saídas das bandas de aproximação e de detalhes, respectivamente.

Exemplo 2 (Wavelet Daubechies 9-7) Agora, consideramos a decomposição wavelet Daubechies 9-7, cujos coeficientes dos filtros de síntese são dados na tabela 3.1.

$H(z)$	$G(z)$
+0,0377692	+0,0644393
-0,0238126	-0,0406275
-0,110991	-0,418373
+0,377596	-0,788613
+0,852079	-0,418373
+0,377596	-0,0406275
-0,110991	+0,0644393
-0,0238126	
+0,0377692	

Tabela 3.1: Coeficientes dos filtros de síntese da wavelet Daubechies 9-7.

A fatoração resulta nas operações de atualização e predição conforme 3.22:

$$\begin{aligned}
\pi_1(z) &= 0 \\
\lambda_1(z) &= +3,172 \left(\frac{1+z^{-1}}{2} \right) \\
\pi_2(z) &= +0,106 \left(\frac{1+z}{2} \right) \\
\lambda_2(z) &= -1,766 \left(\frac{1+z^{-1}}{2} \right) \\
\pi_3(z) &= -0,886 \left(\frac{1+z}{2} \right) \\
\lambda_3(z) &= 0 \\
K &= \frac{1}{1,149}
\end{aligned} \tag{3.22}$$

ou seja,

$$\begin{aligned}
x_{1,0} &= \psi^\uparrow \{x_0\} \\
y_{1,0} &= \omega^\uparrow \{x_0\} \\
x_{1,1} &= x_{1,0} - \pi_1 \{y_{1,0}\} \\
y_{1,1} &= y_{1,0} - \lambda_1 \{x_{1,1}\} \\
x_{1,2} &= x_{1,1} - \pi_2 \{y_{1,1}\} \\
y_{1,2} &= y_{1,1} - \lambda_2 \{x_{1,2}\} \\
x_{1,3} &= x_{1,2} - \pi_2 \{y_{1,2}\} \\
y_{1,3} &= y_{1,2} - \lambda_2 \{x_{1,3}\} \\
x'_1 &= \frac{1}{K} x_{1,3} \\
y'_1 &= K y_{1,3}
\end{aligned} \tag{3.23}$$

Note de que todas as operações $\lambda_i(z)$ são do tipo $mt_i \cdot \left(\frac{1+z^{-1}}{2} \right)$ e todas as operações $\pi_i(z)$ são do tipo $ms_i \cdot \left(\frac{1+z}{2} \right)$; ambas são passa-baixas (suavizadoras), diferindo apenas no atraso ou avanço que cada uma gera, ou seja z^{-1} ou z . Repare que as operações $\lambda_i(z)$ tentam gerar uma estimativa de uma amostra da banda de detalhes situada na posição $2n - 1$ (do sinal original) a partir de uma interpolação linear entre amostras da banda de aproximação situadas nas posições $2n$ e $2n - 2$; por outro lado as operações $\pi_i(z)$ tentam refinar as amostras da banda de aproximação situadas nas posições $2n$ a partir de uma interpolação linear entre amostras da

banda de detalhes situadas em $2n-1$ e $2n+1$. Os atrasos e avanços dessas operações podem ser explicados se tivermos em mente que boas estimativas de uma dada amostra são feitas utilizando-se vizinhos espaciais (ou temporais); desse modo, quando uma dessas operações age no sentido de modificar uma amostra localizada numa determinada posição (ou instante) k , elas o fazem utilizando os vizinhos localizados nos instantes (ou posições) $k-1$ e $k+1$.

Vale comentar que diversos pares de filtros lineares geram, quando fatorados, operações $\pi_i(z)$ e $\lambda_i(z)$ como descrito acima. Essa observação será útil no capítulo 4, onde apresentamos um método de projeto para decomposições wavelets não-lineares.

Capítulo 4

Método para Projeto de Wavelets Não-Lineares

O método que apresentamos neste capítulo utiliza a ferramenta *lifting* descrita no capítulo 3. Este foi inicialmente proposto por Cardoso ([4], [11]) e agora é descrito com o sistema axiomático apresentado no capítulo 2. Posteriormente, algumas modificações para o método são introduzidas com o intuito de adequá-lo melhor a uma dada aplicação.

Considere uma decomposição wavelet de um único nível (a definição encontra-se no capítulo 2) governada por uma *lazy* wavelet (decomposição em componentes polifásicas) seguida de um número arbitrário $I \in \mathbb{N}$ de passos de *lifting* do tipo *prediction-update* $(\pi_1, \lambda_1, \pi_2, \lambda_2, \dots, \pi_i, \lambda_i, \dots, \pi_I, \lambda_I)$. Denotando por $\psi_{\text{LW}}^\uparrow$, $\omega_{\text{LW}}^\uparrow$, $\psi_{\text{LW}}^\downarrow$ e $\omega_{\text{LW}}^\downarrow$ os operadores de análise e síntese da *lazy* wavelet, definidos respectivamente pelas equações 2.12, 2.13, 2.14 e 2.15, temos:

$$\begin{cases} x_{1,0} = \psi_{\text{LW}}^\uparrow\{x_0\}, x_0 \in V_0 \\ y_{1,0} = \omega_{\text{LW}}^\uparrow\{x_0\}, x_0 \in V_0 \\ y_{1,i} = y_{1,i-1} - \pi_i\{x_{1,i-1}\}, i \in \{1, 2, \dots, I\} \\ x_{1,i} = x_{1,i-1} - \lambda_i\{y_{1,i}\}, i \in \{1, 2, \dots, I\} \end{cases} \quad (4.1)$$

$$\begin{cases} x'_1 = \psi_{pu}^\uparrow\{x_0\} = \frac{1}{K} \cdot x_{1,I}, K \in \mathbb{R} \text{ e } x'_1 \in V_1 \\ y'_1 = \omega_{pu}^\uparrow\{x_0\} = K \cdot y_{1,I}, K \in \mathbb{R} \text{ e } y'_1 \in W_1 \end{cases} \quad (4.2)$$

Uma observação quanto à notação: o sinal $x_{j,i}$ pertence ao espaço de sinais de aproximação V_j (j -ésimo nível de decomposição wavelet) e já sofreu i atualizações pelos operadores do tipo λ (i -ésimo passo de *lifting*). Da mesma forma, o sinal $y_{j,i}$ pertence ao espaço de sinais de detalhes W_j (j -ésimo nível de decomposição wavelet) e já sofreu i modificações com operadores π de *lifting* para predição (i -ésimo passo de *lifting*).

A figura 4.1 ilustra este esquema de decomposição.

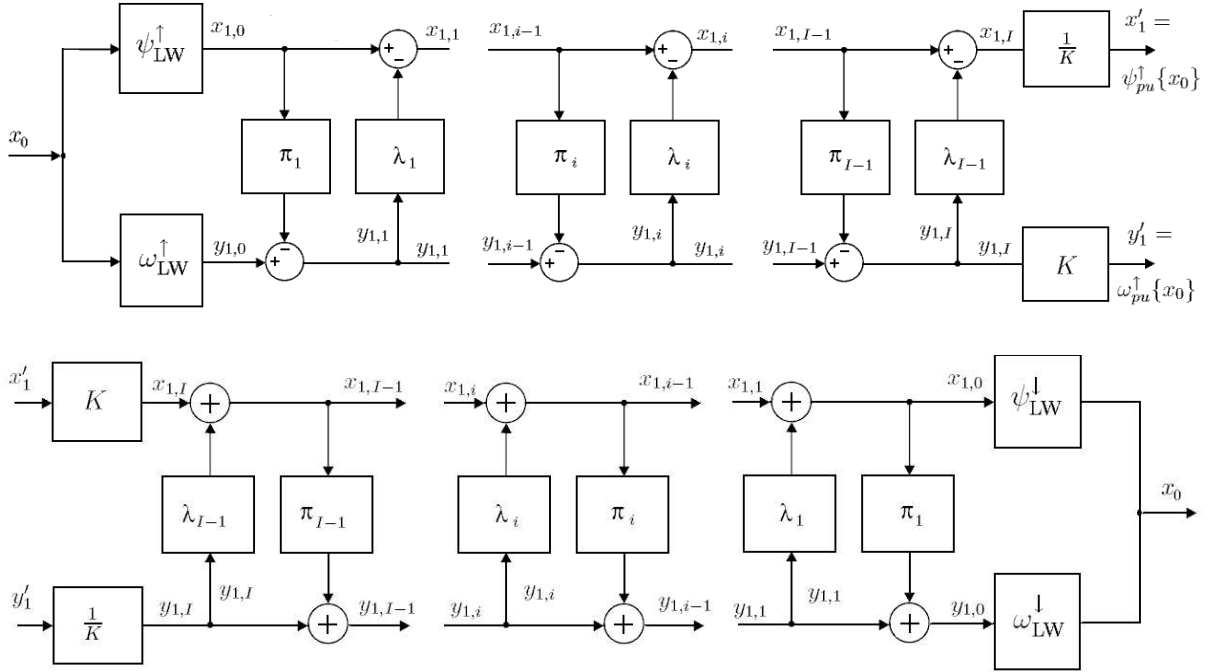


Figura 4.1: Esquema de decomposição do método de projeto: análise (acima) e síntese (abaixo).

Observe que para projetar mais de um nível de decomposição ($J > 1$), basta projetar o primeiro nível com o método e utilizar a decomposição resultante para gerar os demais níveis requeridos decompondo a banda de aproximação de um nível em duas novas do próximo nível. Ou seja, basta fazer realizar o seguinte esquema recursivo de análise:

$$x_0 \rightarrow \{x_1, y_1\} \rightarrow \{x_2, y_2, y_1\} \rightarrow \cdots \rightarrow \{x_k, y_k, y_{k-1}, \dots, y_1\} \rightarrow \cdots \quad (4.3)$$

onde

$$\begin{cases} x'_{j+1} = \psi_{pu}^\uparrow \{x'_j\} \in V_{j+1}, j \geq 0 \\ y'_{j+1} = \omega_{pu}^\uparrow \{x'_j\} \in W_{j+1}, j \geq 0 \end{cases} \quad (4.4)$$

Agora, sejam π_i e λ_i definidos como na equação que segue:

$$\begin{aligned} \pi_i &= cs_i \cdot \pi, i \in \{1, 2, \dots, I\} \text{ e } cs_i \in \mathbb{R} \\ \lambda_i &= ct_i \cdot \lambda, i \in \{1, 2, \dots, I\} \text{ e } ct_i \in \mathbb{R} \end{aligned} \quad (4.5)$$

Na equação 4.5, π e λ são operadores que devem possuir as seguintes duas características:

1. ser uma operação suavizadora que significa, na linguagem de processamento de sinais, ser um filtro passa-baixa;
2. conservar o valor médio do sinal sobre o qual são aplicados, ou seja, $\sum_{n=-\infty}^{+\infty} x(n) = \sum_{n=-\infty}^{+\infty} \pi\{x(n)\} = \sum_{n=-\infty}^{+\infty} \lambda\{x(n)\}$.

Observe que se π ou λ é não-linear, então toda a decomposição é também não-linear.

Neste ponto, cabe uma nota importante. Em [5], Sweldens e Daubechies desenvolveram um algoritmo para fatoração de um par de filtros lineares $(\pi^\uparrow, \omega^\uparrow)$ - que definem uma decomposição wavelet - em passos completos de *lifting*. A aplicação de tal algoritmo a inúmeros pares de filtros lineares freqüentemente resulta em passos de *lifting* cujas operações, tanto de predição quanto de atualização, são filtros passa-baixa com ganho unitário na freqüência zero, ou seja, são operações suavizadoras

que conservam o valor médio do sinal sobre o qual são aplicadas. Veja o exemplo 2 da seção 3.4.

Utilizando o esquema de decomposição wavelet descrito pelas equações 4.1 e 4.2, o método de projeto consiste nas três etapas abaixo:

1. encontrar valores para as constantes $cs_1, ct_1, cs_2, ct_2, \dots, cs_i, ct_i, \dots, cs_I, ct_I$ que satisfaçam as seguintes equações:

$$y'_1(n) = 0, \text{ para } x_0(n) = 1, n \in \mathbb{Z}; \quad (4.6)$$

$$x'_1(n) = 0, \text{ para } x_0(n) = (-1)^n, n \in \mathbb{Z}; \quad (4.7)$$

2. encontrar o valor de K que satisfaça a seguinte igualdade:

$$x'_1(n) = \sqrt{2}, \text{ para } x_0(n) = 1, n \in \mathbb{Z} \quad (4.8)$$

3. escolher as operações morfológicas π e λ (e os respectivos elementos estruturantes) levando-se em consideração as características de suavização e conservação do valor médio citadas anteriormente.

Na linguagem de processamento de sinais, diz-se que a equação 4.6 impõe que a relação entre $y'_1 \in W_1$ e $x_0 \in V_0$ da wavelet projetada seja um filtro passa-alta (saída nula para entrada na frequência zero) e que a relação entre $x'_1 \in V_1$ e $x_0 \in V_0$ seja um filtro passa-baixa (saída nula para entrada de frequência π).

Além disso, a equação 4.6 significa que a decomposição wavelet projetada possui um momento nulo (*vanishing moment*) [12].

A equação 4.8 corresponde à propriedade de conservação de energia das wavelets lineares, ou seja, é uma maneira de garantir que a saída da banda passa-baixa

x'_1 não cresça indefinidamente a medida que aumentamos o número J de níveis de decomposição [11].

Note que o sistema de equações formado por 4.6, 4.7 e 4.8 pode ser solucionado com algoritmos de programação numérica.

Exemplo 1 Neste exemplo, vamos projetar uma decomposição wavelet usando o método descrito anteriormente. Para isso, considere que a decomposição é realizada com somente um passo completo de *lifting*, ou seja, as incógnitas são cs_1 , ct_1 e k .

Começamos por escrever as equações que descrevem a decomposição, como na equação 4.1:

$$\begin{cases} x_{1,0} = \psi_{\text{LW}}^\uparrow\{x_0\}, x_0 \in V_0 \\ y_{1,0} = \omega_{\text{LW}}^\uparrow\{x_0\}, x_0 \in V_0 \\ y_{1,1} = y_{1,0} - \pi_1\{x_{1,0}\} \\ x_{1,1} = x_{1,0} - \lambda_1\{y_{1,1}\} = x_{1,0} - \lambda_1\{y_{1,0} - \pi_1\{x_{1,0}\}\} \end{cases} \quad (4.9)$$

Agora, considere uma entrada $x_0(n) = 1, \forall n$.

$$\begin{cases} x_{1,0}(n) = \psi_{\text{LW}}^\uparrow\{x_0\}(n) = x_0(2n) = 1 \\ y_{1,0}(n) = \omega_{\text{LW}}^\uparrow\{x_0\}(n) = x_0(2n+1) = 1 \\ y_{1,1}(n) = y_{1,0}(n) - \pi_1\{x_{1,0}\}(n) \Rightarrow y_{1,1}(n) = 1 - cs_1 \\ x_{1,1}(n) = x_{1,0}(n) - \lambda_1\{y_{1,1} - \pi_1\{x_{1,0}\}\}(n) \Rightarrow x_{1,1}(n) = 1 - ct_1(1 - cs_1) \end{cases} \quad (4.10)$$

Nas igualdades da equação 4.10 acima, foi necessário aplicar os operadores $\pi_1 = cs_1 \cdot \pi$ e $\lambda_1 = ct_1 \cdot \lambda$ (ver equação 4.5) a sinais constantes (seqüência de 1's). Como as operações π e λ devem ter as propriedades de suavização e conservação do valor médio, quando elas são aplicadas a um sinal constante, elas se comportam como uma função identidade. Ou seja, se $x(n) = 1$, então $\pi_1(n) = cs_1$ e $\lambda_1(n) = ct_1$.

Agora, considere uma entrada $x_0(n) = (-1)^n$. Dessa forma, temos:

$$\begin{cases} x_{1,0}(n) = \psi_{\text{LW}}^\uparrow\{x_0\}(n) = x_0(2n) = 1 \\ y_{1,0}(n) = \omega_{\text{LW}}^\uparrow\{x_0\}(n) = x_0(2n+1) = -1 \\ y_{1,1}(n) = y_{1,0}(n) - \pi_1\{x_{1,0}\}(n) \Rightarrow y_{1,1}(n) = -1 - \text{cs}_1 \\ x_{1,1}(n) = x_{1,0}(n) - \lambda_1\{y_{1,0} - \pi_1\{x_{1,0}\}\}(n) \Rightarrow x_{1,1}(n) = 1 - \text{ct}_1(-1 - \text{cs}_1) \end{cases} \quad (4.11)$$

Começamos impondo que a decomposição satisfaça a equação 4.6. Então, na equação 4.10 devemos fazer $y_{1,1}(n) = 0$, o que implica em

$$1 - \text{cs}_1 = 0 \Rightarrow \text{cs}_1 = 1 \quad (4.12)$$

Em seguida, impomos que a decomposição satisfaça a equação 4.7. Então, na equação 4.11 devemos fazer $x_{1,1}(n) = 0$, o que implica em

$$1 - \text{ct}_1(-1 - \text{cs}_1) = 0 \Rightarrow 1 - \text{ct}_1(-1 - 1) = 0 \Rightarrow \text{ct}_1 = -\frac{1}{2} \quad (4.13)$$

Agora que encontramos o valor das constantes cs_1 e ct_1 , resta encontrar o valor de K . Para isto, devemos fazer, na equação 4.6, com que $x_{1,1}(n) \cdot \frac{1}{K} = \sqrt{2}$, que implica em:

$$[1 - \text{ct}_1(1 - \text{cs}_1)] \cdot \frac{1}{K} = \sqrt{2} \Rightarrow [1 + \frac{1}{2}(1 - 1)] \cdot \frac{1}{K} = \sqrt{2} \Rightarrow K = \frac{1}{\sqrt{2}} \quad (4.14)$$

Caso $I > 1$, ou seja, caso tenhamos mais de um passo de *lifting*, teremos que satisfazer duas igualdades (equações 4.6 e 4.7) utilizando mais de duas incógnitas ($\text{cs}_1, \text{ct}_1, \text{cs}_2, \text{ct}_2, \dots, \text{cs}_i, \text{ct}_i, \dots, \text{cs}_I, \text{ct}_I$), o que faz com que o sistema de equações seja sobredeterminado.

Na próxima seção, sugerimos um procedimento que visa modificar o método de projeto de modo a restringir o campo de busca e otimizar a solução para uma dada aplicação.

4.1 Procedimento para otimização da solução

Com o intuito de restringir o espaço de busca e otimizar a solução para uma dada aplicação, uma modificação pode ser introduzida no método de projeto original descrito anteriormente. A modificação consiste em substituir a primeira etapa do método de projeto pela resolução do seguinte problema de otimização:

$$\begin{aligned}
 & \min_{\bar{C}} \quad \Phi\{(x_0, x'_1, y'_1)\} \\
 & \text{sujeito a} \tag{4.15} \\
 & \quad y'_1(n) = 0, \text{ para } x_0(n) = 1, \forall n \in \mathbb{Z} \\
 & \quad x'_1(n) = 0, \text{ para } x_0(n) = (-1)^n, n \in \mathbb{Z} \\
 & \quad x'_1(n) = \sqrt{2}, \text{ para } x_0(n) = 1, n \in \mathbb{Z}
 \end{aligned}$$

onde $\bar{C} = (cs_1, ct_1, cs_2, ct_2, \dots, cs_i, ct_i, \dots, cs_I, ct_I, K)$ e Φ é uma função objetivo relacionando os sinais x_0 , x'_1 e y'_1 .

Note que, enquanto no caso anterior a primeira etapa do método de projeto consiste apenas em solucionar as equações 4.6 e 4.7, no caso atual estas equações são as restrições de um problema de otimização. Esse fato restringe o espaço de busca - ou seja, toda solução deste método de projeto modificado (atual) é também solução do método original - e, deste modo, a solução então encontrada pode ser mais “vantajosa” para a aplicação em questão, bastando para isso que a função objetivo que se quer minimizar seja convenientemente escolhida.

Capítulo 5

Noções Preliminares em Morfologia Matemática

Este capítulo é dedicado à apresentação dos conceitos básicos de morfologia matemática e à definição dos operadores morfológicos utilizados neste trabalho. Para uma compreensão mais detalhada do assunto, recomendamos [1, 6–8, 13].

Começamos por apresentar operadores definidos em um reticulado completo (*complete lattice* [1,7,9]), para, após, apresentamos outros definidos sob o paradigma não convencional dos inf-semireticulados completos (*complete inf-semilattice* [1,13])

No capítulo 4, quando da descrição do método de projeto de decomposições wavelets não-lineares, vimos que os operadores morfológicos usados nos passos de *lifting* devem possuir duas características: a de suavização e a de conservação do valor médio do sinal sobre o qual eles atuam.

Contemplamos neste capítulo alguns operadores morfológicos construídos sob a ótica da auto-dualidade, pois esta propriedade é interessante por levar, aproximadamente, à conservação do valor médio. Vemos que os reticulados completos são conjuntos sobre os quais existe uma limitação para a construção de operadores auto-duais. O mesmo não acontece com os inf-semireticulados completos, pois estes últimos formam uma base para uma morfologia intrinsecamente auto-dual [1, 13]. O preço a ser pago é termos que abandonar o paradigma dos reticulados completos, que é largamente aceito desde as últimas décadas.

5.1 Morfologia em Reticulados Completos

Nesta seção, apresentamos a morfologia matemática em sua forma clássica, ou seja, definida sobre reticulados completos.

Como já mencionado, esse paradigma impõe limitações para a construção de operadores intrinsecamente auto-duais, isto é, que agem da mesma forma sobre o picos e vales dos sinais, conservando, aproximadamente, o valor médio dos sinais sobre os quais eles atuam.

Na verdade, os operadores definidos em reticulados completos aparecem em pares, onde os integrantes de cada par são duais entre si, como por exemplo a erosão-dilatação e a abertura-fechamento; o efeito é que os operadores dos pares atuam de forma complementar sobre os picos e vales de sinais: enquanto um deles reduz os picos, diminuindo assim o valor médio, o outro faz o justamente o oposto, ou seja, reduz os vales aumentando o valor médio.

Contudo, é possível, com a combinação inteligente de operadores simples, definirmos outros com a propriedade da auto-dualidade. Apresentamos alguns destes no final desta seção.

O leitor interessado pode encontrar uma discussão mais detalhada deste assunto e a prova dos resultados em [7].

Definição 1 (*Reticulado Completo*) Um conjunto \mathcal{L} com uma ordenação parcial \leq é chamado de reticulado completo se todo subconjunto $\mathcal{K} \subseteq \mathcal{L}$ tem um supremo $\bigvee \mathcal{K}$ e um ínfimo $\bigwedge \mathcal{K}$.

Por supremo entendemos menor limite superior, isto é, menor elemento de \mathcal{L} maior ou igual a qualquer elemento de \mathcal{K} . Analogamente, ínfimo quer dizer maior limite inferior, isto é, maior elemento de \mathcal{L} que ainda é menor ou igual a qualquer elemento de \mathcal{K} .

O supremo de um reticulado completo \mathcal{L} é representado por \top e o ínfimo por \perp , ou seja $\bigvee \mathcal{L} = \top$ e $\bigwedge \mathcal{L} = \perp$.

Definição 2 (Adjunção) Considere dois reticulados completos \mathcal{L} e \mathcal{M} , e dois operadores $\varepsilon : \mathcal{L} \rightarrow \mathcal{M}$ e $\delta : \mathcal{M} \rightarrow \mathcal{L}$. Dizemos que o par (ε, δ) constitui uma adjunção (*adjunction* [1, 7, 9]) entre \mathcal{L} e \mathcal{M} se

$$\delta\{y\} \leq x \Leftrightarrow y \leq \varepsilon\{x\}, \quad x \in \mathcal{L} \text{ e } y \in \mathcal{M} \quad (5.1)$$

A equivalência em 5.1 é chamada de relação de adjunção e constitui a peça-chave para a teoria de reticulados completos da morfologia matemática.

Proposição 1 Se o par (ε, δ) , onde $\varepsilon : \mathcal{L} \rightarrow \mathcal{M}$ e $\delta : \mathcal{M} \rightarrow \mathcal{L}$, constitui uma adjunção, ou seja, se ε e δ satisfazem a relação de equivalência em 5.1, então o operador ε satisfaz a seguinte propriedade:

$$\varepsilon\{\bigwedge_{i \in I} x_i\} = \bigwedge_{i \in I} \varepsilon\{x_i\} \quad (5.2)$$

para qualquer família de sinais $\{x_i \mid i \in I\} \subseteq \mathcal{L}$.

Além disso, δ satisfaz a seguinte propriedade dual:

$$\delta\{\bigvee_{i \in I} y_i\} = \bigvee_{i \in I} \delta\{y_i\} \quad (5.3)$$

para qualquer família de sinais $\{y_i \mid i \in I\} \subseteq \mathcal{M}$.

Definição 3 (Erosão e Dilatação) Um operador $\varepsilon : \mathcal{L} \rightarrow \mathcal{M}$ é chamado de

erosão se ele satisfaz a equação 5.2. E um operador $\delta : \mathcal{M} \rightarrow \mathcal{L}$ é chamado de dilatação se ele satisfaz a equação 5.3.

Das definições 2 e 3 e proposição 1, temos a seguinte proposição.

Proposição 2 Se o par (ε, δ) forma uma adjunção entre os reticulados completos \mathcal{L} e \mathcal{M} , então ε é uma erosão e δ é uma dilatação.

Os reticulados completos que usamos neste trabalho são conjuntos de sinais, e não apenas conjuntos de valores. Definimos o reticulado completo $\text{Fun}(E, \mathcal{T})$ como o conjunto de sinais com domínio E e imagem \mathcal{T} . Se \mathcal{T} tem uma ordenação parcial \leq e $x, y \in \text{Fun}(E, \mathcal{T})$, temos:

$$x \leq y, \text{ se } x(p) \leq y(p), \forall p \in E$$

Definição 4 (*Operador Crescente*) Um operador $\varepsilon : \mathcal{L} \rightarrow \mathcal{M}$ é dito crescente se para quaisquer $x_1, x_2 \in \mathcal{L}$ tal que $x_1 > x_2$, tem-se $\varepsilon\{x_1\} > \varepsilon\{x_2\}$.

Agora considere $x_1, x_2 \in \mathcal{L}$, tal que $x_1 > x_2$. Então, dado um operador ε qualquer (não necessariamente uma erosão), temos $\varepsilon\{ \bigwedge_{i \in \{1,2\}} x_i \} = \varepsilon\{x_2\}$. Se ε é uma erosão, pela equação 5.2 temos $\varepsilon\{x_2\} = \bigwedge_{i \in \{1,2\}} \varepsilon\{x_i\}$. Esta última igualdade diz que o mínimo entre $\varepsilon\{x_1\}$ e $\varepsilon\{x_2\}$ (lado direito da igualdade) é $\varepsilon\{x_2\}$ (lado esquerdo da igualdade). Assim, acabamos de mostrar que se ε é uma erosão e $x_1 > x_2$, então $\varepsilon\{x_1\} > \varepsilon\{x_2\}$; logo ε é um operador crescente.

De forma análoga, temos que a dilatação também é um operador crescente.

Dois mapeamentos morfológicos em $\text{Fun}(\mathbb{Z}^d, \mathcal{T})$ (ou seja, de $\text{Fun}(\mathbb{Z}^d, \mathcal{T})$ em $\text{Fun}(\mathbb{Z}^d, \mathcal{T})$) que são a base para a construção de alguns operadores que usaremos neste trabalho, são a dilatação plana e a erosão plana, dados por:

$$\delta_A\{x\}(n) = \bigvee_{k \in A} x(n - k), \quad x \in \text{Fun}(\mathbb{Z}^d, \mathcal{T}) \quad (5.4)$$

$$\varepsilon_A\{y\}(n) = \bigwedge_{k \in A} y(n + k), \quad y \in \text{Fun}(\mathbb{Z}^d, \mathcal{T}) \quad (5.5)$$

Nas equações 5.4 e 5.5 acima, $A \subseteq \mathbb{Z}^d$ é um dado conjunto conhecido como elemento estruturante.

Note que $(\varepsilon_A, \delta_A)$ constitui uma adjunção à luz da definição 2.

É fácil notar, pelas equações 5.4 e 5.5, que se $0 \in A$, então:

$$\delta_A\{x\}(n) \geq x(n), \quad \forall n \quad (5.6)$$

$$\varepsilon_A\{y\}(n) \leq y(n), \quad \forall n \quad (5.7)$$

A equação 5.6 significa que δ_A é um operador extensivo, enquanto a equação 5.7 significa que ε_A é anti-extensivo. Note que um operador extensivo aumenta o valor médio do sinal sobre o qual ele atua, enquanto o anti-extensivo diminui.

Exemplo 1 Considere um sinal $x \in \text{Fun}\{\mathbb{Z}, \mathbb{Z}\}$ definido por

$$\begin{cases} x(n) = n, \text{ para } 0 \leq n \leq 2 \\ x(n) = 5 - n, \text{ para } 2 \leq n \leq 5 \end{cases}$$

Então, $x = [0; 1; 2; 2; 1; 0]$.

Seja $A = \{0, 1\} \subset \mathbb{Z}$. Considere também a condição de contorno $x(-1) = 0$.

Então, pela equação 5.4, temos:

$$\begin{aligned}\delta_A\{x\}(0) &= \bigvee_{k \in A} x(0 - k) = \max\{x(0); x(-1)\} = 0 \\ \delta_A\{x\}(1) &= \bigvee_{k \in A} x(1 - k) = \max\{x(1); x(0)\} = 1 \\ \delta_A\{x\}(2) &= \bigvee_{k \in A} x(2 - k) = \max\{x(2); x(1)\} = 2 \\ \delta_A\{x\}(3) &= \bigvee_{k \in A} x(3 - k) = \max\{x(3); x(2)\} = 2 \\ \delta_A\{x\}(4) &= \bigvee_{k \in A} x(4 - k) = \max\{x(4); x(3)\} = 2 \\ \delta_A\{x\}(5) &= \bigvee_{k \in A} x(5 - k) = \max\{x(5); x(4)\} = 1\end{aligned}$$

Ou seja, $\delta_A\{x\} = [0; 1; 2; 2; 2; 1]$.

Vemos que $\delta_A\{x\} \geq x$, o que já era de se esperar pelo que vimos anteriormente.

Denotamos $\text{id}_{\mathcal{L}}$ a operação identidade em um reticulado completo \mathcal{L} , ou simplesmente id quando nenhuma confusão possa ocorrer.

Definição 5 (Negação de sinais) Uma operação $v : \mathcal{L} \rightarrow \mathcal{L}$ é uma negação se ela satisfaz as seguintes condições:

- v é bijetora;
- v reverte uma ordenação, isto é, $x \leq y \Rightarrow v(y) \leq v(x)$, para $x, y \in \mathcal{L}$;
- $v^2 = \text{id}$, o operador identidade.

Para todo $x \in \text{Fun}(E, \mathcal{T})$, exemplos de negação são dados nas equações 5.8 e 5.9:

$$v\{x\} = -x, \text{ se } \mathcal{T} = \mathbb{R} \quad (5.8)$$

$$v\{x\} = N - 1 - x, \text{ se } \mathcal{T} = \{0, 1, \dots, N - 1\} \quad (5.9)$$

Definição 6 (*Mapeamento de Negação de operadores*) Sejam \mathcal{L} e \mathcal{M} dois reticulados completos com os operadores de negação $v_{\mathcal{L}}$ e $v_{\mathcal{M}}$, respectivamente. Dado um operador $\sigma : \mathcal{L} \rightarrow \mathcal{M}$, o mapeamento de negação $\sigma^* : \mathcal{L} \rightarrow \mathcal{M}$ é definido por

$$\sigma^* = v_{\mathcal{M}}\sigma v_{\mathcal{L}} \quad (5.10)$$

Se $v = v_{\mathcal{L}} = v_{\mathcal{M}}$ é o operador de negação dado pela equação 5.9, pode-se mostrar que as operações δ_A e ε_A , definidas respectivamente em 5.4 e 5.5, satisfazem as seguintes igualdades [6, 8]:

$$\begin{aligned} \delta_A^* &= \varepsilon_A \\ \varepsilon_A^* &= \delta_A \end{aligned} \quad (5.11)$$

Definição 7 (*Dualidade entre operadores*) Dois operadores $\sigma : \mathcal{L} \rightarrow \mathcal{M}$ e $\varphi : \mathcal{L} \rightarrow \mathcal{M}$ são duais entre si se eles satisfazem a seguinte relação

$$\sigma^* = \varphi \text{ e } \varphi^* = \sigma \quad (5.12)$$

Concluimos, por 5.11 e 5.12, que as operações δ_A e ε_A são duais entre si. Então, é óbvio que se $(\varepsilon_A, \delta_A)$ constitui uma adjunção, o mesmo acontece com o par

$(\delta_A^*, \varepsilon_A^*)$. Essa característica é válida para qualquer par de operadores que forma uma adjunção, como na proposição que segue.

Proposição 3 Se o par (ε, δ) forma uma adjunção entre \mathcal{L} e \mathcal{M} , então $(\delta^*, \varepsilon^*)$ também forma.

Agora apresentamos a definição de filtros morfológicos.

Definição 8 (*Filtros Morfológicos, Abertura, Fechamento*) Seja σ um mapeamento em um reticulado completo \mathcal{L} . Considere as seguintes definições:

1. σ é idempotente, se $\sigma^2 = \sigma$;
2. Se σ é crescente e idempotente, então σ é um filtro morfológico;
3. Um filtro morfológico σ que satisfaz $\sigma \leq \text{id}$ (σ é anti-extensivo) é chamado de *abertura*;
4. Um filtro morfológico σ que satisfaz $\sigma \geq \text{id}$ (σ é extensivo) é chamado de *fechamento*.

Proposição 4 Seja o par (ε, δ) uma adjunção entre os reticulados completos \mathcal{L} e \mathcal{M} . Então, à luz da definição 8, o operador $\alpha : \mathcal{M} \rightarrow \mathcal{M}$ definido por $\alpha = \delta\varepsilon$ é uma abertura em \mathcal{M} e o operador $\beta : \mathcal{L} \rightarrow \mathcal{L}$ definido por $\beta = \varepsilon\delta$ é um fechamento em \mathcal{L} .

Vimos anteriormente que o par $(\varepsilon_A, \delta_A)$ dado por 5.5 e 5.4 constitui uma adjunção em $\text{Fun}(\mathbb{Z}, \mathcal{T})$. Então, pela proposição 4, conclui-se que os mapeamentos $\alpha_A : \text{Fun}(\mathbb{Z}, \mathcal{T}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{T})$ e $\beta_A : \text{Fun}(\mathbb{Z}, \mathcal{T}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{T})$, dados pelas equações 5.13 e 5.14 são, respectivamente, uma abertura e um fechamento. Dizemos que

α_A é a abertura pelo elemento estruturante A e β é o fechamento pelo elemento estruturante A .

$$\alpha_A = \delta_A \varepsilon_A \quad (5.13)$$

$$\beta_A = \varepsilon_A \delta_A \quad (5.14)$$

Proposição 5 Os operadores α_A e β_A definidos, respectivamente, por 5.13 e 5.14 são duais entre si (com relação ao operador de negação dado pela equação 5.9), ou seja,

$$\begin{aligned} \alpha_A^* &= \beta_A \\ \beta_A^* &= \alpha_A \end{aligned} \quad (5.15)$$

Exemplo 2 Seja f uma imagem digital cujas linhas pertencem ao conjunto $\text{Fun}(\mathbb{Z}, \mathcal{T})$, onde $\mathcal{T} = \{0; 1; \dots; 255\}$. Considere $A = \{-2; -1; 0; 1; 2\} \subset \mathbb{Z}$.

As figuras 5.1, 5.2, 5.3 ilustram, respectivamente:

- a imagem f ;
- a imagem resultante da aplicação da operação abertura $\alpha_A = \delta_A \varepsilon_A$ às linhas de f ;
- a imagem resultante da aplicação da operação fechamento $\beta_A = \varepsilon_A \delta_A$ às linhas de f .

Vemos que a aplicação da abertura deixa a imagem mais escura, pois $\alpha_A \leq \text{id}$ (anti-extensiva). Por outro lado, vemos que a aplicação do fechamento deixa a imagem mais clara, pois $\beta_A \geq \text{id}$ (extensiva). Lembre que quanto menor o valor de um *pixel*, mais escuro ele é, e vice-versa.



Figura 5.1: Imagem f : Lena.

Como visto no capítulo 4, neste trabalho estamos interessados em operadores que possuem as seguintes propriedades:

- ser suavizador;
- conservar o valor médio dos sinais sobre os quais eles atuam.

Os operadores δ_A e ε_A não satisfazem nenhuma dessas duas propriedades. Já a abertura $\alpha_A = \delta_A \varepsilon_A$ e o fechamento $\beta = \varepsilon_A \delta_A$, ambos filtros morfológicos, são operações suavizadoras. Porém, elas não conservam o valor médio do sinal sobre o qual elas atuam, já que $\alpha_A \leq \text{id}$ e $\beta \geq \text{id}$. Contudo, o mapeamento definido pela média aritmética entre esses dois filtros conservam, aproximadamente, o valor médio.

Definição 9 (Operador γ_A) Sejam α_A e β_A governados pelas equações 5.13 e 5.14. O operador $\gamma_A : \text{Fun}(\mathbb{Z}, \mathcal{L}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{L})$ é definido como segue



Figura 5.2: Aplicação de α_A às linhas de f .

$$\gamma_A = \frac{\alpha_A + \beta_A}{2} \tag{5.16}$$

Definição 10 (*Auto-dualidade de um operador*) Um operador $\sigma : \mathcal{L} \rightarrow \mathcal{L}$ é dito auto-dual se

$$\sigma^* = \sigma \tag{5.17}$$

Proposição 6 O operador γ_A definido pela equação 5.16 é auto-dual com relação ao operador de negação dado pela equação 5.9, ou seja, $\gamma_A(x) = N - 1 - \gamma_A(N - 1 - x)$, $x \in \text{Fun}(E, \{0, 1, \dots, N - 1\})$.



Figura 5.3: Aplicação de β_A às linhas de f .

Esta proposição não é encontrada na literatura, por isso é necessário prová-la.

Prova Vimos na proposição 5, equação 5.15, que as operações α_A e β_A são duais entre si com relação ao operador de negação dado por 5.9. Sendo assim, temos:

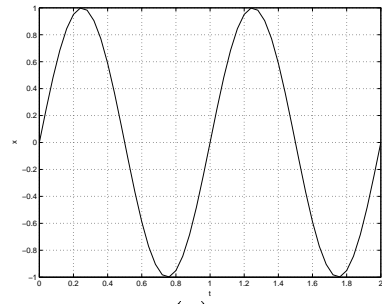
$$\begin{aligned}
\gamma_A^*\{x\} &= v\gamma_A\{vx\} = \\
&= N - 1 - \gamma_A\{vx\} = \\
&= N - 1 - \left(\frac{\alpha_A\{vx\} + \beta_A\{vx\}}{2}\right) = \\
&= \frac{2N - 2 - \alpha_A\{vx\} - \beta_A\{vx\}}{2} = \\
&= \frac{(N - 1 - \alpha\{vx\}) + (N - 1 - \beta\{vx\})}{2} = \\
&= \frac{(v\alpha\{vx\}) + (v\beta\{vx\})}{2} = \\
&= \frac{\alpha^*\{x\} + \beta^*\{x\}}{2} = \\
&= \frac{\beta\{x\} + \alpha\{x\}}{2} = \\
&= \gamma_A\{x\}
\end{aligned} \tag{5.18}$$

Exemplo 3 Neste exemplo, comparamos a aplicação dos operadores α_A , β_A e γ_A , onde $A = \{-2; -1; 0; 1; 2\}$, a um sinal unidimensional $x \in \text{Fun}(\mathbb{Z}, \mathbb{R})$. A figura 5.4 mostra:

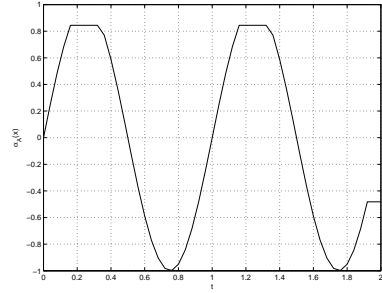
- (a) o sinal original x ;
- (b) a abertura de x , ou seja, $\alpha_A\{x\}$;
- (c) o fechamento de x , ou seja, $\beta_A\{x\}$;
- (d) $\gamma_A\{x\}$.

Note que o operador γ aproximadamente conserva o valor médio de x .

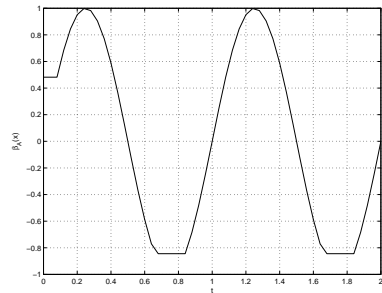
Quando aplicados em imagens, os operadores auto-duais possuem a característica de processar o *background* e o *foreground* da mesma forma, ao contrário dos que não são auto-duais. Assim, eles tendem a conservar o valor médio do sinal sobre o qual eles são aplicados.



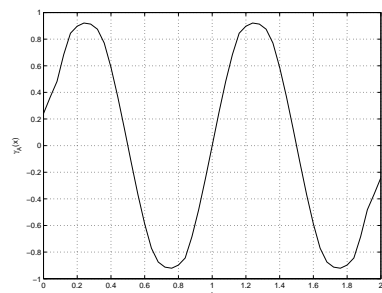
(a)



(b)



(c)



(d)

Figura 5.4: (a) Sinal original x ; (b) $\alpha_A\{x\}$; (c) $\beta_A\{x\}$; (d) $\gamma_A\{x\}$.

Exemplo 4 A tabela 5.1 apresenta os valores médios do sinal x da figura 5.4(a) e após aplicação de diversos operadores morfológicos. Em todos os casos, $A = \{-1; 0; 1\}$.

Tabela 5.1: Comparação do valor médio após aplicação de diversos operadores morfológicos ao sinal x da figura 5.4(a).

sinal	valor médio
sinal original x	$-1.9945 \cdot 10^{-017}$
após aplicação de δ_A	0.3021
após aplicação de ε_A	-0.3021
após aplicação de α_A	-0.0320
após aplicação de β_A	0.0320
após aplicação de γ_A	$-4.9525 \cdot 10^{-017}$

Veja na tabela que o operador γ_A aproximadamente conserva o valor médio do sinal x .

Em seguida, vemos outros exemplos de operadores auto-duais num reticulado completo que usamos neste trabalho.

Definição 11 (*Operador toggle contrast*) O operador *toggle contrast* $\kappa_A : \mathcal{L} \rightarrow \mathcal{L}$ é definido por:

$$\kappa_A = \begin{cases} \beta_A, & \text{se } \beta_A - \text{id} < \text{id} - \alpha_A \\ \alpha_A, & \text{nos demais casos} \end{cases} \quad (5.19)$$

Definição 12 (*Operador morphological centre*) Considere dois operadores morfológicos em \mathcal{L} definidos por $m_A = \alpha_A \beta_A$ e $n_A = \beta_A \alpha_A$.

O operador *morphological centre* $\theta_A : \mathcal{L} \rightarrow \mathcal{L}$ é definido por:

$$\theta_A = (\text{id} \vee (m_A \wedge n_A)) \wedge (m_A \vee n_A) \quad (5.20)$$

onde $x \vee y$ é equivalente a $\vee\{x, y\}$ e $x \wedge y$ é equivalente a $\wedge\{x, y\}$.

Proposição 7 O operador *toggle-contrast* $\kappa : \mathcal{L} \rightarrow \mathcal{L}$, definido pela equação 5.19 é auto-dual [6].

Proposição 8 O operador *morphological centre* $\theta : \mathcal{L} \rightarrow \mathcal{L}$, definido pela equação 5.20 é auto-dual [6].

Na próxima seção, estudamos outros operadores que são intrinsecamente auto-duais.

5.2 Morfologia em Inf-Semireticulados Completos

Uma consequência importante do fato de definirmos a teoria clássica da morfologia matemática sob a ótica dos reticulados completos é que os operadores assim definidos aparecem em pares, auto-duais entre si, como por exemplo os pares erosão-dilatação e abertura-fechamento.

O efeito causado por este fato é que, dados dois operadores duais entre si pertencentes a um par, um deles é anti-extensivo enquanto o outro extensivo.

Vimos anteriormente que o método de projeto descrito no capítulo 4 requer que os operadores usados nos passos de *lifting* sejam suavizadores e conservem o valor médio de seu argumento.

Os operadores anti-extensivos e os extensivos definidos em reticulados completos, ainda que sejam suavizadores como, por exemplo, a abertura e o fechamento, não conservam o valor médio. Isto acontece, pois eles atuam de forma diferente nos picos (*foreground*) e vales (*background*) dos sinais. Os operadores auto-duais, como vimos na seção anterior, agem da mesma forma sobre os picos e vales. Isto faz

com que eles aproximadamente conservem o valor médio. Por isso a motivação de estudarmos operadores auto-duais.

Nesta seção, ao contrário do que acontece com os reticulados completos, apresentamos um paradigma em que os operadores morfológicos definidos são, por natureza, auto-duais. Esse novo paradigma é criado abandonando-se a ordenação parcial \leq dos reticulados completos e substituindo-a por uma nova ordenação \preceq . Com isto, pode-se gerar um novo sistema axiomático a que chamamos inf-semireticulado completo (*complete inf-semilattice* ou *cisl* [1, 13]). O preço a ser pago é que este sistema não é mais compatível com o dos reticulados completos, este último largamente aceito desde a última década. O nome inf-semireticulado completo vem do fato de, neste sistema, não ser mais possível falar em supremo de uma coleção de sinais, como falávamos nos reticulados completos.

Não fazemos uma apresentação exaustiva deste assunto, mas expomos tão somente os principais resultados que são úteis a este trabalho. O leitor interessado pode encontrar uma discussão mais detalhada e as provas dos resultados em Heijmans e Kresch [1].

Na seção anterior, definimos os operadores em reticulados completos, que possuem uma ordenação parcial \leq . Sob este paradigma, o mapeamento $x \rightarrow -x$, $x \in \text{Fun}(E, \mathbb{R})$ reverte a ordenação parcial, ou seja:

$$x \leq y \Rightarrow -y \leq -x, \text{ para } x, y \in \text{Fun}(E, \mathbb{R}).$$

Da mesma forma, o mapeamento $x \rightarrow N-1-x$, $x \in \text{Fun}(E, \{0, 1, \dots, N-1\})$ também reverte a ordenação parcial, ou seja:

$$x \leq y \Rightarrow N-1-y \leq N-1-x, \text{ para } x, y \in \text{Fun}(E, \{0, 1, \dots, N-1\}).$$

Como dissemos, o passo-chave em direção a uma morfologia auto-dual é a substituição de \leq por outra ordenação parcial, como na equação 5.21

$$x \preceq y, \text{ se } 0 \leq x \leq y \text{ ou } y \leq x \leq 0, \text{ onde } x, y \in \mathcal{T}. \quad (5.21)$$

Nesta nova ordenação parcial, dois valores são comparáveis se ambos possuem o mesmo sinal ou se, pelo menos, um deles é zero. Note que $x \preceq y$ significa que os valores x e y possuem o mesmo sinal e x é mais próximo de zero que y .

Definição 11 (*Inf-Semireticulado Completo ou cisl*) Um conjunto \mathcal{L} com uma ordenação parcial \preceq é chamado de inf-semireticulado completo (*complete inf-semilattice* ou *cisl* [1, 13]) se todo subconjunto $\mathcal{K} \subseteq \mathcal{L}$ tem um ínfimo $\wedge \mathcal{K}$.

Seja $\mathcal{K} = \{x_i | i \in I\}$. O ínfimo $\wedge \mathcal{K}$ é dado por:

$$\wedge_{i \in I} x_i = \begin{cases} \bigwedge_{i \in I} x_i, & \text{se todo } x_i \geq 0 \\ \bigvee_{i \in I} x_i, & \text{se todo } x_i \leq 0 \\ 0, & \text{nos demais casos} \end{cases} \quad (5.22)$$

Note que todo inf-semireticulado completo tem no mínimo um elemento, que é zero neste caso. No entanto, o supremo \vee em geral não existe [13]. Na verdade, só podemos falar em supremo de um conjunto de valores caso todos eles apresentem o mesmo sinal; neste caso, o supremo é o valor mais afastado de zero.

A próxima proposição mostra porque esta nova alternativa para ordenação parcial é de especial interesse em relação à propriedade de auto-dualidade.

Proposição 7 Sejam dois valores quaisquer $x, y \in \mathcal{K}$. O mapeamento $x \rightarrow -x$ em \mathcal{K} não reverte a ordenação parcial \preceq , ou seja:

$$x \preceq y \Leftrightarrow -x \preceq -y$$

Da mesma forma que na seção anterior, $\text{Fun}(E, \mathcal{T})$ representa o conjunto de todos os sinais com domínio em E e imagem \mathcal{T} . Se \mathcal{T} tem uma ordenação parcial \preceq e $x, y \in \text{Fun}(E, \mathcal{T})$, temos:

$$x \preceq y, \text{ se } x(p) \preceq y(p) \forall p \in E$$

A figura 5.5 ilustra dois sinais $x, y \in \text{Fun}(E, \mathcal{T})$ em que $x \preceq y$. A figura 5.6 mostra, na linha escura, o ínfimo entre dois sinais x, y em um inf-semireticulado completo.

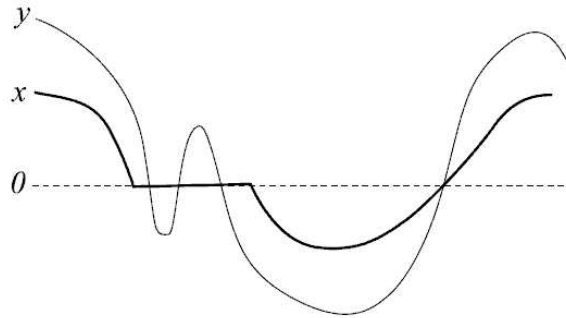


Figura 5.5: Exemplo de dois sinais $x, y \in \text{Fun}(E, \mathcal{T})$ em que $x \preceq y$. Fonte da figura: [1].

Definição 12 (Adjunção em *cisl's*) Considere dois inf-semireticulados completos (*cisl's*) \mathcal{L} e \mathcal{M} , e dois operadores $\varepsilon' : \mathcal{L} \rightarrow \mathcal{M}$ e $\delta' : \mathcal{M} \rightarrow \mathcal{L}$. Dizemos que o par (ε', δ') constitui uma adjunção entre \mathcal{L} e \mathcal{M} se

$$\delta'\{y\} \preceq x \Leftrightarrow y \preceq \varepsilon'\{x\}, x \in \mathcal{L} \text{ e } y \in \mathcal{M} \quad (5.23)$$

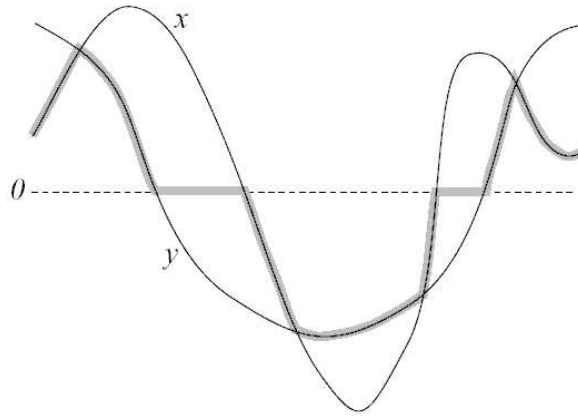


Figura 5.6: Ínfimo entre dois sinais: $\wedge x, y$. Fonte da figura: [1].

Proposição 8 (Erosão e Dilatação em *cisl's*) Um operador $\varepsilon' : \mathcal{L} \rightarrow \mathcal{M}$ é chamado de erosão se ele satisfaz a equação 5.24:

$$\varepsilon'\{\wedge_{i \in I} x_i\} = \wedge_{i \in I} \varepsilon'\{x_i\} \quad (5.24)$$

para qualquer família não vazia de sinais $\{x_i \mid i \in I\} \subseteq \mathcal{L}$.

Definamos o conjunto $\mathcal{M}[\varepsilon']$ por:

$$\mathcal{M}[\varepsilon'] = \{y \in \mathcal{M} \mid \exists x \in \mathcal{L} : y \preceq \varepsilon\{x\}\} \quad (5.25)$$

Se (ε', δ') constitui uma adjunção entre \mathcal{L} e \mathcal{M} , e ε é dada por 5.24, então o operador $\delta' : \mathcal{M}[\varepsilon'] \rightarrow \mathcal{L}$ dado pela equação 5.26 é uma dilatação.

$$\delta'\{y\} = \wedge \{x \in \mathcal{L} \mid y \preceq \varepsilon'\{x\}\}, y \in \mathcal{M} \quad (5.26)$$

Note que o domínio da dilatação δ' é limitado aos sinais que são menores ou iguais, no sentido \preceq , aos elementos do conjunto imagem de ε' . Sendo assim, se

(ε', δ') constitui um *cisl*, a dilatação δ' sempre pode ser aplicada no resultado da aplicação de ε' a um sinal qualquer.

Definição 13 (*Operador Crescente*) Um operador $\varepsilon' : \mathcal{L} \rightarrow \mathcal{M}$ é dito crescente (com relação à ordenação \preceq) se para quaisquer $x_1, x_2 \in \mathcal{L}$ tal que $x_1 \succeq x_2$, tem-se $\varepsilon'\{x_1\} \succeq \varepsilon'\{x_2\}$.

Dois mapeamentos morfológicos em $\text{Fun}(\mathbb{Z}^d, \mathcal{T})$ que usamos neste trabalho são a dilatação plana e a erosão plana em *cisl*'s, dados por:

$$\delta'_A\{x\}(n) = \vee_{k \in A} x(n - k), \quad x \in \text{Fun}(\mathbb{Z}^d, \mathcal{T}) \quad (5.27)$$

$$\varepsilon'_A\{y\}(n) = \wedge_{k \in A} y(n + k), \quad y \in \text{Fun}(\mathbb{Z}^d, \mathcal{T}) \quad (5.28)$$

Pode-se mostrar que os operadores definidos anteriormente nas equações 5.27 e 5.28 são crescentes. Além disso, o par $(\varepsilon'_A, \delta'_A)$ constitui uma adjunção à luz da definição 12 e seus operadores satisfazem as equações 5.24 e 5.26 [13].

Exemplo 5 Considere um sinal $x \in \text{Fun}\{\mathbb{Z}, \mathbb{Z}\}$ definido por

$$x(n) = n - 2, \text{ para } n \in \mathbb{Z} \text{ tal que } 0 \leq n \leq 4$$

Então, $x = [-2; -1; 0; 1; 2]$.

Seja $A = \{0; 1\} \subset \mathbb{Z}$. Considere também a condição de contorno $x(-1) = 0$.

Então, pela equação 5.27, temos:

$$\begin{aligned}
\delta'_A\{x\}(0) &= \Upsilon_{k \in A} x(0 - k) = \Upsilon\{x(-1); x(0)\} = -2 \\
\delta'_A\{x\}(1) &= \Upsilon_{k \in A} x(1 - k) = \Upsilon\{x(0); x(1)\} = -2 \\
\delta'_A\{x\}(2) &= \Upsilon_{k \in A} x(2 - k) = \Upsilon\{x(1); x(2)\} = -1 \\
\delta'_A\{x\}(3) &= \Upsilon_{k \in A} x(3 - k) = \Upsilon\{x(2); x(3)\} = 1 \\
\delta'_A\{x\}(4) &= \Upsilon_{k \in A} x(4 - k) = \Upsilon\{x(3); x(4)\} = 2
\end{aligned}$$

Ou seja, $\delta'_A\{x\} = [-2; -2; -1; 1; 2]$.

Note que $\delta'_A\{x\} \succeq x$.

Considere agora que $A = \{-1; 0; 1\}$ e as condições de contorno são $x(-1) = 0$ e $x(5) = 0$. Nessa nova condição, suponha que queremos calcular o valor da dilatação em $n = 2$. Então:

$$\delta'_A\{x\}(2) = \Upsilon_{k \in A} x(2 - k) = \Upsilon\{x(1); x(2); x\{3\}\} = \Upsilon\{-1; 0; 1\}^\# \quad (5.29)$$

Neste caso a dilatação δ'_A não é definida, pois o supremo de um conjunto de valores com sinais opostos não existe.

Definição 14 (Abertura) Um operador $\alpha : \mathcal{L} \rightarrow \mathcal{L}$ em um inf-semireticulado completo \mathcal{L} é chamado abertura se ele satisfaz as seguintes propriedades:

1. α é crescente;
2. α é idempotente, ou seja, $\alpha^2 = \alpha$;
3. α é anti-extensivo, ou seja, $\alpha\{x\} \preceq x$, $x \in \mathcal{L}$.

Como α é crescente e idempotente, então ele é um filtro morfológico.

Proposição 9 Seja o par (ε', δ') uma adjunção entre os inf-semireticulados completos \mathcal{L} e \mathcal{M} . Então, à luz da definição 8, o operador $\alpha' : \mathcal{L} \rightarrow \mathcal{L}$ definido por $\alpha' = \delta'\varepsilon'$ é uma abertura em \mathcal{L} [13].

Note que o operador $\alpha'_A : \text{Fun}(\mathbb{Z}, \mathcal{T}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{T})$, definido por $\alpha'_A = \delta'_A \varepsilon'_A$, onde δ'_A e ε'_A são dados respectivamente pelas equações 5.27 e 5.28, é uma abertura no inf-semireticulado completo $\text{Fun}(\mathbb{Z}, \mathcal{T})$. Na abertura, a dilatação existe, pois ela é calculada após aplicação prévia da uma erosão. A operação fechamento (dilatação seguida de erosão) não será usada neste trabalho, pois ela nem sempre existe, uma vez que o seu domínio é limitado, assim como o da dilatação.

A figura 5.7 mostra um exemplo de aplicação da erosão ε'_A (lado esquerdo) e da abertura $\alpha'_A = \delta'_A \varepsilon'_A$ (lado direito), onde $A = \{-3; -2; -1; 0; 1; 2; 3\}$ [13]. Repare que a erosão cria um “platô” na região onde o sinal de entrada x cruza o eixo horizontal. Esse platô garante a existência da dilatação e, por conseguinte, da abertura.

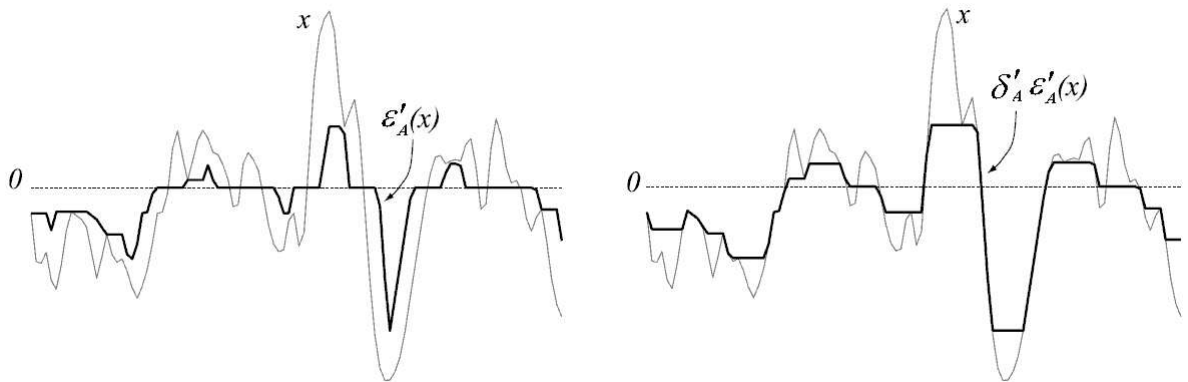


Figura 5.7: Exemplo de aplicação de erosão (lado esquerdo) e abertura (lado direito). Repare que a erosão cria um “platô” na região onde o sinal de entrada x cruza o eixo horizontal. Esse platô garante a existência da dilatação e, por conseguinte, da abertura. Fonte da figura: [1].

Definição 15 (*Negação de operadores*) Dado um operador $\sigma : \mathcal{L} \rightarrow \mathcal{M}$, o mapeamento de negação $\sigma^* : \mathcal{L} \rightarrow \mathcal{M}$ é definido por

$$\sigma^*(x) = -\sigma(-x), x \in \mathcal{L} \quad (5.30)$$

Definição 16 (*Auto-dualidade de um operador*) Um operador $\sigma : \mathcal{L} \rightarrow \mathcal{L}$ é dito auto-dual se

$$\sigma^* = \sigma \quad (5.31)$$

ou seja, $\sigma\{x\} = -\sigma\{-x\}$, $x \in \mathcal{L}$.

Proposição 10 A erosão ε'_A definida pela equação 5.28 é uma operação auto-dual. [13].

Proposição 11 A dilatação δ'_A definida pela equação 5.27 é uma operação auto-dual. [13].

Proposição 12 Considere a adjunção formada pelo par $(\varepsilon'_A, \delta'_A)$. A abertura definida por $\alpha' = \delta'_A \varepsilon'_A$ é uma operação auto-dual. [13].

Nesta seção, definimos auto-dualidade fazendo uso do mapeamento de negação $x \rightarrow -x$. Porém, neste trabalho aplicamos operadores morfológicos nas

linhas de imagens digitais, que são sinais pertencentes a $\text{Fun}(\mathbb{Z}, \mathcal{T})$, onde $\mathcal{T} = \{0, 1, \dots, N - 1\}$. Neste caso, dada uma linha x de uma imagem, não faz sentido falar em $-x$, pois este operador não pode ser definido em \mathcal{T} .

Então, à primeira vista, não é possível aplicar os operadores auto-duais apresentados nesta seção (ε'_A , δ'_A e α'_A) a sinais com imagem em $\mathcal{T}\{0, 1, \dots, N - 1\}$. Contudo, a próxima proposição fornece o subsídio necessário para que os resultados desta seção possam ser aplicados de forma apropriada nesse trabalho.

Proposição 13 Seja x um sinal definido em $\text{Fun}(E, \mathcal{T})$, onde $\mathcal{T} = \{0, 1, \dots, N - 1\}$. Considere também $\xi : \text{Fun}(E, \mathcal{T}') \rightarrow \text{Fun}(E, \mathcal{T}')$, onde $\mathcal{T}' = \{-\frac{N-1}{2}, -\frac{N-3}{2}, \dots, 0, \dots, \frac{N-3}{2}, \frac{N-1}{2}\}$, um operador auto-dual no sentido da definição 16, ou seja, $\xi\{x\} = -\xi\{-x\}$, $x \in \mathcal{T}'$. Então o mapeamento $\eta : \text{Fun}(E, \mathcal{T}) \rightarrow \text{Fun}(E, \mathcal{T})$ definido pela equação 5.32 abaixo:

$$\eta\{x\} = \xi\left\{x - \frac{N-1}{2}\right\} + \frac{N-1}{2} \quad (5.32)$$

é uma operação auto-dual no sentido $\eta\{x\} = N - 1 - \eta\{N - 1 - x\}$.

Exemplo 6 Seja x um sinal definido em $\text{Fun}\{\mathbb{Z}, \mathcal{T}\}$, onde $\mathcal{T} = \{0; 1; \dots; 255\}$. Então, podemos aplicar a erosão definida na equação 5.28 da seguinte forma:

$$\eta_A\{x\} = \varepsilon'_A\left\{x - \frac{255}{2}\right\} + \frac{255}{2} \quad (5.33)$$

Apesar ε'_A ser definida em $\text{Fun}(\mathbb{Z}, \mathcal{T}')$, onde $\mathcal{T}' = \{-\frac{255}{2}; -\frac{253}{2}; \dots; 0; \dots; \frac{253}{2}; \frac{255}{2}\}$, o domínio de η_A continua sendo o mesmo de x , ou seja, $\text{Fun}\{\mathbb{Z}, \mathcal{T}\}$.

Uma operação η com a propriedade de auto-dualidade mencionada na proposição 13 age da mesma forma no *background* e *foreground* dos sinais sobre os quais

é aplicada, o que tende a conservar o seu valor médio.

Capítulo 6

Resultados

Diversas são as aplicações das decomposições em multiresoluções: codificação de imagens, vídeo e áudio baseada em subbandas [14]; as modulações digitais e as técnicas de múltiplo acesso realizadas num sistema de comunicação digital [2, 3]; a análise de sinais para remoção de ruídos; a análise estatística; etc.

Neste trabalho, aplicamos as decomposições wavelets projetadas com o método descrito no capítulo 4 em imagens, para posterior aplicação em um algoritmo de codificação de imagens baseado em subbandas. Assim, começamos a primeira seção deste capítulo descrevendo sucintamente o método de compressão de imagens chamado EZW, que possui três etapas: uma decomposição wavelet, uma quantização, e uma codificação baseada em subbandas. Em seguida, para um completo entendimento da aplicação em questão, mostramos como realizar decomposições wavelets bidimensionais de imagens utilizando as decomposições unidimensionais projetadas como o método apresentado neste trabalho. Na segunda seção, apresentamos resultados de testes das decomposições projetadas, quando usadas para implementar a etapa da transformação wavelet do método EZW. São analisadas as influências de diversos parâmetros nos resultados: o número de níveis de decomposição, o comprimento do elemento estruturante usado nas operações morfológicas, algumas soluções apresentadas pelo método de projeto descrito no capítulo 4, e os operadores morfológicos usados nos passos de *lifting*. Em todos os casos, observamos que quanto maior o elemento estruturante da operação morfológica, piores os resultados dos tes-

tes. Por isso, apresentamos um artifício para que possamos modificar uma imagem de modo que a utilização de um elemento estruturante de determinado tamanho na imagem modificada funcione como se estivesse sendo utilizado um de tamanho menor na imagem original. Esse artifício é agregado aos operadores morfológicos e, então, novos testes são realizados. Os melhores resultados são comparados ao da wavelet linear Daubechies 9-7, amplamente usada na literatura.

6.1 Decomposições Bidimensionais e EZW

Neste trabalho, o método de projeto descrito no capítulo 4 é utilizado para gerar decomposições não-lineares em multiresolução do tipo wavelet para que sejam aplicadas em imagens, com o objetivo de prepará-las para posterior aplicação do método EZW [15] (*Embedded Zerotree Wavelet*) de compressão de imagens baseado em codificação de subbandas. O método EZW foi desenvolvido para aproveitar as características presentes em imagens resultantes de uma decomposição linear em multiresolução obtida com uma DWT (*Discrete Wavelet Transform*), com o objetivo de codificá-las eficientemente segundo a ótica do compromisso entre taxa e distorção [16]. Como vários outros algoritmos de compressão de imagens, o EZW pode ser dividido em três etapas: a transformação - que reduz a redundância presente no conjunto de dados original, fazendo com que a imagem transformada apresente alta concentração de energia, isto é, poucos elementos detendo grande parte da energia -, a quantização - que reduz a faixa dinâmica e a precisão numérica dos dados transformados -, e a codificação - que representa o conjunto quantizado de uma forma mais compacta, atribuindo códigos de comprimento grande aos elementos menos prováveis e de comprimento pequeno aos mais prováveis [14, 17, 18]. Note que a quantização impõe perdas, já que reduz a precisão numérica dos dados. Assim, parte da compressão deve-se à quantização e outra parte à codificação. A figura 6.1 ilustra as etapas do método EZW; nela a etapa da transformação wavelet, que faz o mapeamento entre a imagem original x_0 e as subbandas x_{2LL} , x_{2LH} , x_{2HL} , x_{2HH} , x_{1LH} , x_{1HL} e x_{1HH} , é discutida em seguida.

O que fizemos nos testes foi substituir a etapa de transformação do EZW,

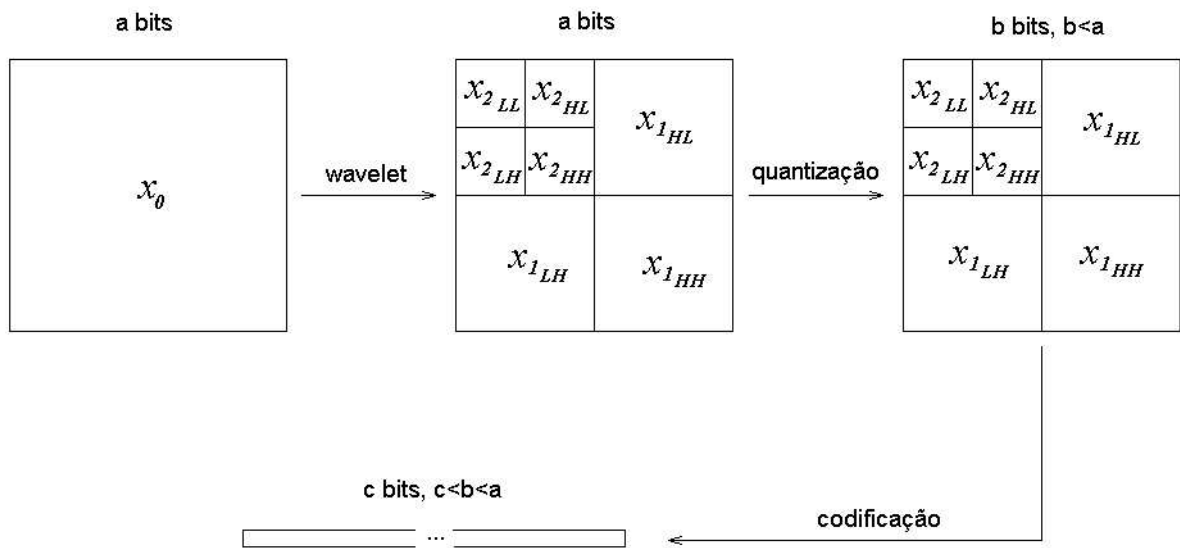


Figura 6.1: Etapas do método EZW de compressão de imagens.

inicialmente realizada por uma DWT, que é uma transformação linear, por uma decomposição wavelet morfológica (não-linear); as etapas de quantização e codificação foram mantidas inalteradas. Os resultados das decomposições morfológicas foram considerados tanto melhores quanto menor a distorção entre a imagem original e a recuperada após processo de compressão / decompressão com EZW a uma dada taxa de bits por símbolo; essa distorção é calculada por uma medida objetiva chamada PSNR (*Peak Signal to Noise Ratio*), que relaciona a energia do pico da imagem original com a energia média da distorção causada pelo processo de compressão / decompressão.

O fato de modificarmos somente a etapa de transformação do EZW foi feito com a intenção de analisarmos a influência da utilização das decomposições wavelet morfológicas projetadas neste trabalho em detrimento às wavelets lineares. Contudo, as etapas de quantização e codificação do algoritmo EZW original foram projetadas para que fossem utilizadas com uma DWT, ou seja, com uma wavelet linear. Assim, uma análise mais justa poderia ser feita se substituíssemos, também, as etapas de quantização e de codificação por outras otimizadas para serem usadas com as decomposições wavelet morfológicas. Isto pode ser objeto de estudo de trabalhos futuros.

Quando filtramos linearmente um sinal bidimensional $x(m, n)$, como uma imagem, precisamos de um filtro cuja resposta ao impulso $h(m, n)$ também seja bidimensional. Porém, uma classe importante de filtros bidimensionais é aquela dos filtros separáveis, ou seja, aqueles que podem ser separados numa filtragem unidimensional das linhas seguida de uma filtragem unidimensional das colunas. No caso linear, isto é equivalente a termos $h(m, n) = h_1(m)h_2(n)$, onde $h_1(m)$ é a resposta ao impulso de um filtro (unidimensional) linear que deve ser aplicado nas linhas e $h_2(n)$ a resposta ao impulso que deve ser aplicado nas colunas.

Da mesma forma que nos filtros lineares, podemos realizar uma decomposição bidimensional com as wavelets unidimensionais não-lineares geradas neste trabalho. Primeiramente, basta decompor as linhas da imagem teste, gerando duas subbandas intermediárias com as seguintes características:

- uma delas resultante de filtragem passa-baixa (*low-pass*) nas linhas (sinal de aproximação), denominada de x_L (“L” de *low-pass*);
- outra resultante de filtragem passa-alta (*high-pass*) nas linhas, denominada de x_H (“H” de *high-pass*);

Posteriormente, decomparamos as colunas destas subbandas intermediárias x_L e x_H , gerando quatro subbandas finais (duas subbandas a partir de cada subbanda intermediária) com as seguintes características:

- uma delas sendo uma aproximação da imagem de teste (passa-baixa nas linhas e nas colunas), e é denominada por x_{LL} ;
- outra tendo somente detalhes horizontais (passa-baixa nas linhas e passa-alta nas colunas), sendo denominada de x_{LH} ;
- uma terceira tendo somente detalhes verticais (passa-alta nas linhas e passa-baixa nas colunas), denominada por x_{HL} ;
- a última tendo detalhes diagonais (passa-alta nas linhas e colunas), ou seja, que não são nem verticais nem horizontais, sendo denominada x_{HH} .

Neste esquema de decomposição, se o sinal de teste tem M linhas e N colunas, então as subbandas intermediárias têm, cada uma, dimensão $M \times \frac{N}{2}$ - devido à subamostragem nas linhas - e as quatro subbandas finais têm dimensões $\frac{M}{2} \times \frac{N}{2}$. A figura 6.2 ilustra esse esquema de decomposição.

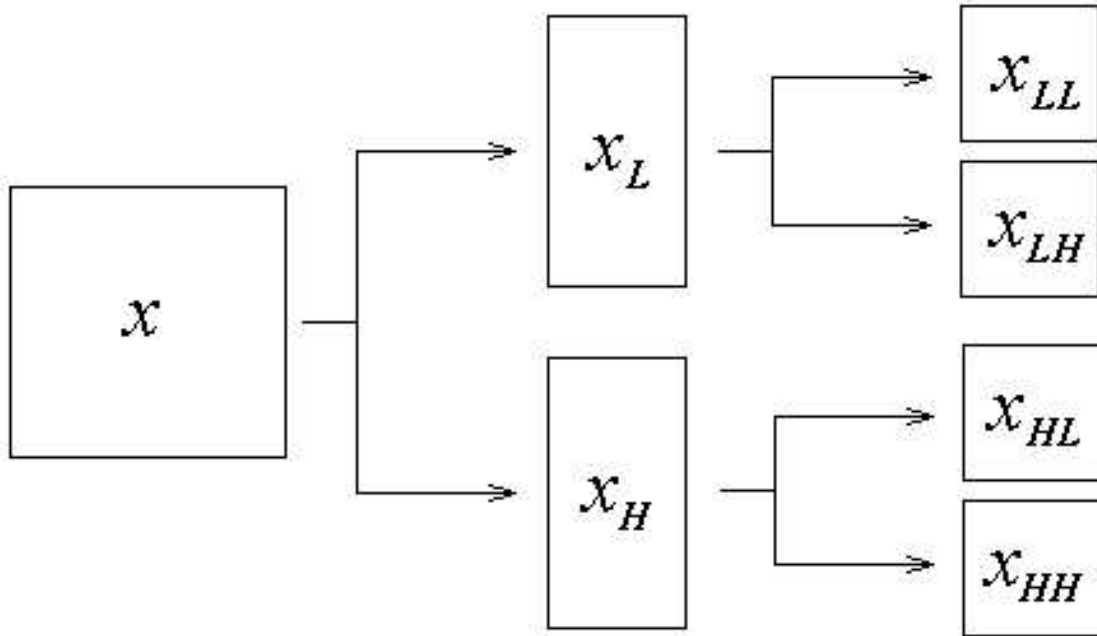


Figura 6.2: Um nível de decomposição bidimensional gerada a partir de uma decomposição unidimensional. Da esquerda para a direita: sinal original, as duas subbandas intermediárias geradas a partir da decomposição das linhas da imagem original, as quatro subbandas finais geradas a partir de decomposição das colunas das subbandas intermediárias.

A figura 6.3 mostra como as quatro subbandas resultantes de uma decomposição bidimensional podem ser dispostas de modo a formar uma imagem com a mesma dimensão da original.

Exemplo 1 Para um melhor entendimento deste esquema de decomposição bidimensional separável, considere uma decomposição unidimensional governada apenas por uma *lazy wavelet* (uma decomposição em componentes polifásicas), onde os operadores de análise ψ_{LW}^\dagger e ω_{LW}^\dagger são dados pela equação 2.12. Considere o sinal bidimensional $x(m, n)$ a ser decomposto dado por:

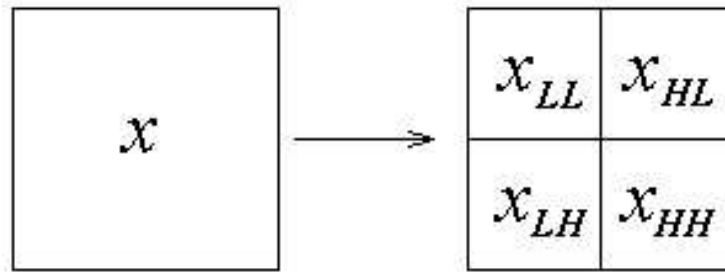


Figura 6.3: Disposição das quatro subbandas resultantes de um nível de decomposição wavelet de modo a formar um sinal com a mesma dimensão do sinal original.

$$x = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad (6.1)$$

Após a aplicação da wavelet às linhas, obtemos as seguintes subbandas intermediárias:

$$x_L = \begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \\ 13 & 15 \end{bmatrix} \quad (6.2)$$

$$x_H = \begin{bmatrix} 2 & 4 \\ 6 & 8 \\ 10 & 12 \\ 14 & 16 \end{bmatrix} \quad (6.3)$$

Como explicado anteriormente, a componente x_L é resultante de filtragem passa-baixa nas linhas (subbanda de aproximação) e x_H é resultante de filtragem passa-alta nas linhas (subbanda de detalhes).

Em seguida, a subbanda intermediária x_L é novamente decomposta, gerando

os sinais x_{LL} , resultante de filtragem passa-baixa nas linhas e nas colunas, e x_{LH} , resultante de filtragem passa-baixa nas linhas e passa-alta nas colunas. Já a subbanda intermediária x_H é decomposta gerando os sinais x_{HL} , resultante de filtragem passa-alta nas linhas e passa-baixa nas colunas, e x_{HH} , resultante de filtragem passa-alta nas linhas e colunas.

$$x_{LL} = \begin{bmatrix} 1 & 3 \\ 9 & 11 \end{bmatrix} \quad (6.4)$$

$$x_{LH} = \begin{bmatrix} 5 & 7 \\ 13 & 15 \end{bmatrix} \quad (6.5)$$

$$x_{HL} = \begin{bmatrix} 2 & 4 \\ 10 & 12 \end{bmatrix} \quad (6.6)$$

$$x_{HH} = \begin{bmatrix} 6 & 8 \\ 14 & 16 \end{bmatrix} \quad (6.7)$$

Para realizar uma decomposição bidimensional em mais de um nível, a subbanda de aproximação x_{jLL} de um nível deve ser decomposta em novas quatro subbandas: x_{j+1LL} , x_{j+1LH} , x_{j+1HL} e x_{j+1HH} . A figura 6.4 mostra uma decomposição wavelet bidimensional em dois níveis.

Agora que foram apresentados o princípio de funcionamento do EZW e o esquema de decomposição bidimensional separável, passamos à seção de testes.

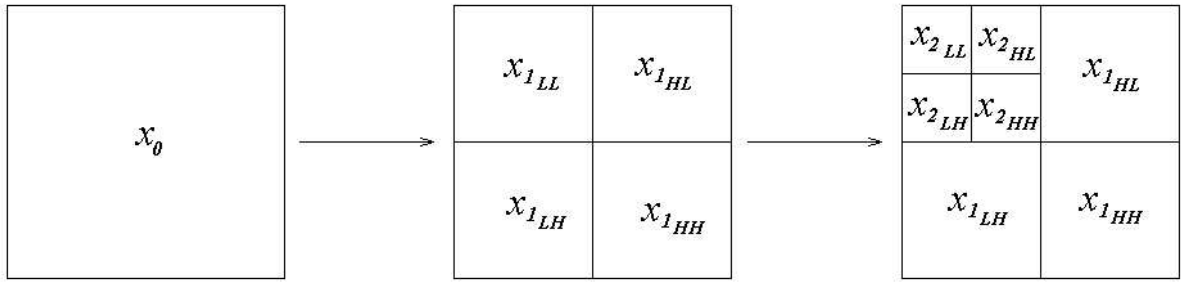


Figura 6.4: Dois níveis de decomposição wavelet bidimensional.

6.2 Testes

Usando o método descrito no capítulo 4, foram projetadas duas decomposições diferentes [4], cujos coeficientes resultantes estão dispostos na tabela 6.1. Chamamos os dois conjuntos de coeficientes de projeto 1 e projeto 2, respectivamente. Repare que os projetos apresentam duas soluções diferentes para as equações 4.6, 4.7 e 4.6, ambas contemplando 2 passos de *lifting*.

Tabela 6.1: Decomposições wavelets projetadas utilizando o método do capítulo 4.

	ms_1	mt_1	ms_2	mt_2	K
projeto 1	0, 0	1, 0	-0, 5	0, 0	$1/\sqrt{2}$
projeto 2	0, 0	1, 1	-0, 476190	-0, 1050	0, 742462

Para averiguar a qualidade das decomposições wavelets projetadas, as aplicamos à imagem teste da figura 6.5, cujas dimensões são 512x512 *pixels* e cada elemento (*pixel*) é representado por um inteiro sem sinal de 8 bits.

Em cada passo de *lifting*, começamos utilizando o operador morfológico γ_A (definido na seção 5.1) com elementos estruturantes (EE) unidimensionais de comprimentos 2, 3, 4 e 5 *pixels* (o uso do EE unidimensional é necessário, pois, em se tratando de filtragem bidimensional separável, os filtros aplicados nas linhas e colunas devem ser unidimensionais). São realizados testes com 3, 4, 5 e 6 níveis de decomposição.

Como dito anteriormente, as decomposições wavelets morfológicas são usadas para realizar a etapa de transformação do algoritmo de compressão de imagens EZW [15], em substituição à DWT (*Discrete Wavelet Transform*) originalmente usada neste algoritmo. As imagens transformadas são quantizadas com taxas de 0,15bpp (*bits por pixel*) e 0,50bpp, para, então, serem reconstruídas (na verdade, queremos dizer que as taxas de 0,15bpp e 0,50bpp são conseguidas após aplicação da quantização e codificação EZW, pois somente com a quantização não é possível atingir tais taxas). As imagens reconstruídas são comparadas à original, onde utilizamos a PSNR (*Peak Signal to Noise Ratio*) como medida objetiva de qualidade.



Figura 6.5: Imagem original: Lena, 512x512, 8bpp.

A tabela 6.2 mostra resultados da aplicação do EZW à imagem da figura 6.5, onde a etapa de transformação é realizada pela decomposição morfológica resultante do projeto 1. A operação γ_A definida na equação 5.16 foi usada nos passos de *lifting*. São apresentados resultados para diversos níveis de decomposição, EE com diversas

unidades de comprimento (u.c.) e taxa de quantização / codificação de 0.15bpp. Valores de PSNR para o EZW tradicional, ou seja, com a etapa de transformação realizada por uma DWT (linear), podem ser encontrados na última coluna da tabela 6.7.

Dois fatos importantes podem ser observados na tabela 6.2. Primeiramente, que poucos níveis de decomposição resultam em PSNR pequena; isso acontece, pois poucos níveis implicam em baixa concentração de energia na imagem transformada, o que prejudica a eficiência do quantizador/codificador EZW. Por outro lado, se o número de níveis de decomposição cresce indefinidamente, há uma concentração enorme de energia em poucos elementos da imagem transformada; com isso, quando um desses coeficientes com muita energia é quantizado, há uma grande perda de informação, o que, em última instância, também prejudica a PSNR da imagem recuperada. Assim, há um número ótimo de níveis de decomposição que maximiza a PSNR, que, em nosso caso, é 5 ou 6. Por isso, a partir de agora todos os testes que seguem são realizados com 5 níveis de decomposição. Em segundo lugar, pode-se observar que, para todos os níveis de decomposição, o aumento no comprimento do elemento estruturante implica em diminuição da PSNR. Isto acontece, pois, com elementos estruturantes de comprimento menor, as estimativas geradas para uma dada amostra em cada passo de *lifting* da decomposição são mais fiéis que com o uso de elementos estruturantes grandes, já que os primeiros implicam em usar vizinhos mais próximos para as estimativas.

As figuras 6.6, 6.7, 6.8 e 6.9 foram obtidas com o uso do projeto 1. A primeira delas mostra o resultado da decomposição bidimensional separável utilizando-se 3 níveis de decomposição e EE de comprimento 2, ou seja, mostra a imagem que passará pelas etapas de quantização e codificação. Esta imagem (a da figura 6.6) foi, então, comprimida utilizando o EZW e recuperada com uma taxa de 0,15bpp; o resultado é mostrado na figura 6.7. A terceira apresenta a imagem recuperada após a decomposição em 5 níveis, com EE de comprimento 2 e quantização / codificação de 0,15bpp. A última mostra a imagem recuperada após decomposição em 5 níveis, com EE de comprimento 5 e quantização / codificação de 0,15bpp.

Os resultados obtidos com o uso do projeto 2 foram semelhantes aos encon-

Tabela 6.2: Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 1, onde a operação γ_A foi usada nos passos de *lifting*. São apresentados resultados para diversos níveis de decomposição, EE com diversas unidades de comprimento (u.c.) e taxa de quantização / codificação de 0.15bpp.

projeto	níveis decomp	EE (u.c.)	taxa (bpp)	PSNR (dB)
1	3	2	0,15	25.5429
1	3	3	0,15	25.1130
1	3	4	0,15	24.6950
1	3	5	0,15	24.2177
1	4	2	0,15	26.8918
1	4	3	0,15	26.2708
1	4	4	0,15	25.6588
1	4	5	0,15	25.0485
1	5	2	0,15	27.2051
1	5	3	0,15	26.5024
1	5	4	0,15	25.8176
1	5	5	0,15	25.1734
1	6	2	0,15	27.2037
1	6	3	0,15	26.4968
1	6	4	0,15	25.9094
1	6	5	0,15	25.3295

tramos com o projeto 1, porém ligeiramente piores, motivo pelo qual entendemos não ser necessário exibi-los.

Podemos ver que o uso do EE de tamanho 5 faz com que a imagem recuperada apresente alguns artefatos (riscos horizontais e verticais) que não são presentes quando usamos o de tamanho 2; estes defeitos são equivalentes ao efeito de *ringing* apresentado no caso linear.

Apesar disso, todas as imagens apresentam efeito de blocos, fato que contribui

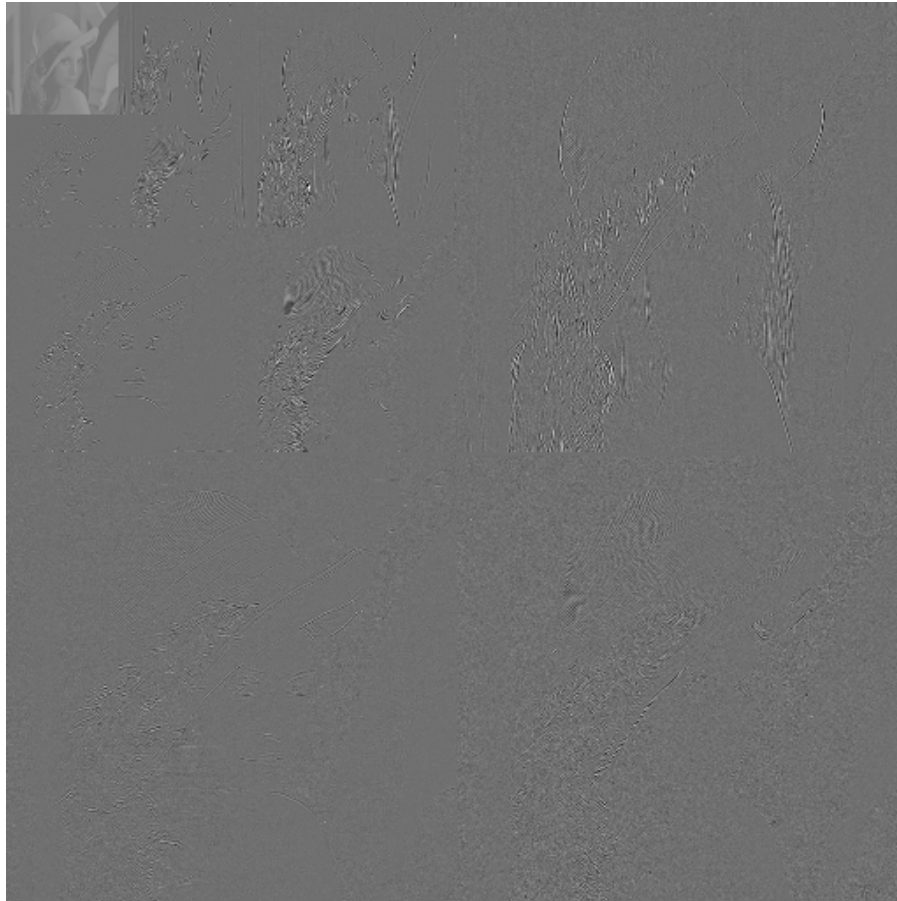


Figura 6.6: Três níveis de decomposição wavelet em subbandas utilizando o projeto 1.

para que nenhuma imagem recuperada após compressão / decompressão apresente boa qualidade, o que pode ser comprovado pelos baixos valores de PSNR (veja última coluna da tabela 6.7). Três são as possíveis causas:

1. Os coeficientes gerados pelo método de projeto podem não definir uma decomposição wavelet com boa qualidade para a aplicação em questão;
2. O operador γ_A pode não ser adequado, sendo necessários testes com outros operadores;
3. Por fim, os elementos estruturantes, mesmo os com 2 unidades de comprimento, podem ainda ser grandes demais para gerarem bons resultados.

Para tentar solucionar o problema da baixa qualidade das imagens recuperadas, atacamos a primeira das possíveis causas, ou seja, a solução para o método de



Figura 6.7: Imagem recuperada após processo de compressão / decompressão, onde foram usados o projeto 1 e o operador γ_A . Características: 3 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.

projeto. Para encontrarmos uma solução melhor, realizamos as seguintes modificações:

- 1) estabelecemos uma decomposição com 3 passos de *lifting*, ao invés de 2; e
- 2) utilizamos nos passos de *lifting* as operações

$$\begin{aligned}\Lambda(e^{j\omega}) &= \frac{1+e^{-j\omega}}{2} \text{ e} \\ \Pi(e^{j\omega}) &= \frac{1+e^{j\omega}}{2}\end{aligned}\tag{6.8}$$

e procedemos à minimização da seguinte função objetivo (ver capítulo 4):



Figura 6.8: Imagem recuperada após processo de compressão / decompressão, com o uso do projeto 1 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.

$$\Phi\{(x_0, x'_1, y'_1)\} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left[\left| \frac{X'_1(e^{j\omega})}{X_0(e^{j\omega})} \right|^2 + \left(\left| \frac{Y'_1(e^0)}{X_0(e^0)} \right| - \left| \frac{Y'_1(e^{j\omega})}{X_0(e^{j\omega})} \right| \right)^2 \right] d\omega \quad (6.9)$$

onde $X(e^{j\omega})$ é a transformada de Fourier de $x(n)$.

Cabe notar que λ e π só assumiram os mapeamentos da equação 6.8 para fins do projeto dos coeficientes de cada passo de *lifting* da decomposição. Caso contrário, não faria sentido criar uma função objetivo baseada em transformadas de Fourier. Após o projeto, λ e π assumem uma das operações morfológicas apresentadas no capítulo 5.



Figura 6.9: Imagem recuperada com o uso do projeto 1 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 5, taxa de 0,15bpp.

Essa função objetivo visa minimizar a energia da banda de rejeição, o que pode ser interessante para a aplicação em questão.

Observamos que os coeficientes resultantes da fatoração da wavelet linear Daubechies 9-7 em passos de *lifting* (veja exemplo 2 da seção 3.4, equação 3.22) satisfazem as equações de projeto 4.2 e 4.6 e minimizam a função objetivo dada por 6.9. Escolhemos usar esse conjunto de coeficientes, o qual denominamos de projeto 3 e dispomos na tabela 6.3, pois os resultados da aplicação das decomposições wavelets não-lineares, onde os passos de *lifting* são dados pelo produto de tais coeficientes pelos operadores morfológicos apresentados neste trabalho, são comparados aos obtidos com a utilização de transformada wavelet linear Daubechies 9-7.

Da mesma forma que fizemos com o projeto 1, dispomos na tabela 6.4 alguns

Tabela 6.3: Projeto 3: coeficientes resultantes da fatoração da wavelet Daubechies 9-7.

	ms_1	mt_1	ms_2	mt_2	ms_3	K
projeto 2	0,0	3,172	+0.106	-1,766	-0,886	1/1,149

resultados obtidos com o projeto 3. Os resultados foram ligeiramente melhores. Como era de se esperar, quanto maior o comprimento do EE, menor é a PSNR.

A figura 6.10 apresenta a imagem recuperada após processo de compressão / decompressão a uma taxa de 0.15bpp, onde foi usada uma decomposição em 5 níveis baseada nos coeficientes do projeto 3 e operador γ_A com EE de comprimento 2. Veja que, mesmo com o novo projeto, a imagem recuperada apresenta efeito de blocos.

Tabela 6.4: Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3, onde a operação γ_A foi usada nos passos de *lifting*.

projeto	níveis decomp	EE (u.c.)	taxa (bpp)	PSNR (dB)
1	5	2	0,15	27.2075
1	5	3	0,15	26.6609
1	5	4	0,15	26.0660
1	5	5	0,15	25.5113

Como vimos, a nova solução de projeto não trouxe melhorias notáveis nem para os valores de PSNR nem para o problema dos efeitos de blocos apresentados pelas imagens recuperadas após compressão / decompressão.

Por isso, passamos a explorar a segunda causa possível para tais problemas, ou seja, a falta de habilidade do operador γ_A em gerar boas decomposições para nossa aplicação. A tabela 6.5 mostra os resultados de testes com os operadores abaixo, para as taxas de quantização / codificação de 0,15bpp e 0,50bpp:

1. α_A (abertura padrão), definido na equação 5.13;



Figura 6.10: Imagem recuperada com o uso do projeto 3 e operador γ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.

2. γ_A (média aritmética entre abertura e fechamento), definido na equação 5.16;
3. κ_A (*toggle contrast*), definido na equação 5.19;
4. θ_A (*morphological centre*), definido na equação 5.20;
5. α'_A (abertura auto-dual em um *cisl*), definido por $\alpha'_A = \delta'_A \varepsilon'_A$, onde δ'_A e ε'_A são dados respectivamente pelas equações 5.27 e 5.28.

Observe na tabela 6.5 que os piores resultados, em todos os cenários de simulação, são atribuídos ao uso do operador α_A , que, apesar de ser suavizador, não conserva o valor médio do sinal sobre o qual ele atua. Note que a abertura em um inf-semireticulado completo α'_A , auto-dual por natureza, apresentou resultados

Tabela 6.5: Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3. São apresentados resultados para diversos operadores morfológicos, comprimentos de EE e taxas de quantização / codificação.

operador	níveis decomp	EE (u.c.)	taxa (bpp)	PSNR (dB)
α_A	5	2	0,15	24.5014
γ_A	5	2	0,15	27.2075
κ_A	5	2	0,15	28.1485
θ_A	5	2	0,15	24.9853
α'_A	5	2	0,15	24.9118
α_A	5	3	0,15	22.5808
γ_A	5	3	0,15	26.6609
κ_A	5	3	0,15	26.5264
θ_A	5	3	0,15	24.4434
α'_A	5	3	0,15	23.5356
α_A	5	2	0,50	28.7648
γ_A	5	2	0,50	32.0856
κ_A	5	2	0,50	33.1922
θ_A	5	2	0,50	29.1578
α'_A	5	2	0,50	29.1422
α_A	5	3	0,50	26.9994
γ_A	5	3	0,50	31.6463
κ_A	5	3	0,50	30.7221
θ_A	5	3	0,50	28.6997
α'_A	5	3	0,50	28.0895

melhores que a abertura padrão α_A em todos os casos. Por outro lado, os melhores resultados foram conseguidos com o uso do operador κ_A .

Ilustramos em figuras apenas o pior e o melhor caso, com o uso de elemento estruturante com 2 u.c., para as taxa de codificação/quantização de 0,15bpp e 0,50bpp. As figuras 6.11, 6.12, 6.13, 6.14 mostram as imagens recuperadas após compressão

/ decompressão com EZW, com utilização dos seguintes cenários, respectivamente: operador α_A e 0.15bpp, operador κ_A e 0.15bpp, operador α_A e 0.50bpp e operador κ_A e 0.50bpp. Pode-se perceber que a qualidade visual das imagens recuperadas com o uso do operador α_A , que não conserva o valor médio, é muito ruim.



Figura 6.11: Imagem recuperada com o uso do projeto 3 e operador α_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.

Mesmo os melhores resultados obtidos com estas simulações não são, como veremos mais adiante, semelhantes aos encontrados com o uso da wavelet linear Daubechies 9-7 (ver tabela 6.7). Além disso, pode-se perceber ainda o problema do efeito de blocos, mesmo nas imagens recuperadas com taxa de 0,50bpp, como na figura 6.14. Assim, passamos a explorar a terceira possível causa para estes problemas: o comprimento dos elementos estruturantes.



Figura 6.12: Imagem recuperada com o uso do projeto 3 e operador κ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,15bpp.

6.3 Operadores modificados e novos testes

Apesar de não ser possível utilizar elementos estruturantes com menos de 2 u.c., pois se assim o fizéssemos as operações morfológicas seriam equivalentes ao operador identidade, usamos um artifício: criamos uma imagem modificada de modo que a atuação, nesta imagem, de um EE de um dado comprimento seja semelhante à atuação de um EE de comprimento menor na imagem original. Para que esse artifício seja entendido, necessitamos compreender o comportamento dos operadores π e λ nos passos de *lifting*.

O método de projeto descrito no capítulo 4 requer que o sinal $x_0(n)$ a ser decomposto seja, antes da atuação dos passos de *lifting*, separado em suas compo-



Figura 6.13: Imagem recuperada com o uso do projeto 3 e operador α_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,50bpp.

mentes polifásicas, ou seja, sofra uma decomposição prévia por uma *lazy wavelet*. Isto implica no fato que a posição n de um elemento $x_{1,i}$ (veja notação no capítulo 4) da banda de aproximação corresponde à posição $2n$ do sinal original x_0 . De forma análoga, a posição n de um elemento $y_{1,i}$ da banda de detalhes corresponde à posição $2n + 1$ do sinal x_0 . A figura 6.15 ilustra esse fato.

Suponha que a operação π de *lifting* para predição seja um operador morfológico com um elemento estruturante $A = \{-1; 0; 1\}$, por exemplo. Neste caso, o sinal $y_{1,i}(n)$ será modificado pela operação π a partir das amostras $x_{1,i}(n - 1)$, $x_{1,i}(n)$ e $x_{1,i}(n + 1)$, as quais, no sinal original, correspondem às posições $2n - 2$, $2n$ e $2n + 2$; ou seja, em cada passo de *lifting*, a predição de uma dada amostra é feita utilizando-se vizinhos distantes (tomando-se como referência o sinal x_0), mesmo com



Figura 6.14: Imagem recuperada com o uso do projeto 3 e operador κ_A . Características: 5 níveis de decomposição, EE unidimensional de comprimento 2, taxa de 0,50bpp.

Figura 6.15: Correspondência entre as posições dos sinais: (a) x_0 , (b) $x_{1,i}$ e (c) $y_{1,i}$ e numa decomposição wavelet. Note que $x_{1,i}$ e $y_{1,i}$ são espacialmente desalinhados.

$$\begin{array}{cccccccccccc}
 (a) & x_0(0) & x_0(1) & x_0(2) & x_0(3) & x_0(4) & x_0(5) & \cdots & x_0(2n) & x_0(2n+1) & \cdots \\
 (b) & x_{1,j}(0) & & x_{1,j}(1) & & x_{1,j}(2) & & \cdots & x_{1,j}(n) & & \cdots \\
 (c) & & y_{1,j}(0) & & y_{1,j}(1) & & y_{1,j}(2) & \cdots & & y_{1,j}(n) & \cdots
 \end{array}$$

elementos estruturantes pequenos. A mesma conclusão é obtida para os passos de *lifting* para atualização.

Para evitar esse fato, antes de aplicarmos o operador π , superamostramos o

sinal $x_{1,i}$ com o operador σ_t^\downarrow definido na equação 6.10:

$$\begin{cases} \sigma_t^\downarrow\{x\}(2n) = x(n) \\ \sigma_t^\downarrow\{x\}(2n+1) = t \end{cases} \quad (6.10)$$

gerando o sinal $\pi\sigma_t^\downarrow\{x_{1,i}\}$. Com isso, a amostra situada na posição n do sinal $\pi\sigma_t^\downarrow\{x_{1,i}\}$ corresponde à posição n do sinal original x_0 . Além disso, se o EE é dado por $A = \{-1; 0; 1\}$ a amostra da posição n de $\pi\sigma_t^\downarrow\{x_{1,i}\}$ é calculada com base nas amostras das posições $n-1$, n e $n+1$ de $\sigma_t^\downarrow\{x_{1,i}\}$ que, por sua vez correspondem às posições $n-1$, n e $n+1$ de x_0 . Assim, com o uso do operador σ_t^\downarrow vizinhos mais próximos são usados para o cálculo de uma amostra num passo de *lifting* para predição, o que faz parecer que o EE foi menor que no caso anterior (sem operador de superamostragem).

Em seguida, devemos subamostrar $\pi\sigma_t^\downarrow\{x_{1,i}\}$ nas posições $2n+1$, o que corresponde às posições n do sinal $y_{1,i}$. Essa subamostragem é feita utilizando-se o operador σ^\uparrow definido na equação 6.11:

$$\sigma^\uparrow\{x\}(n) = x(2n+1) \quad (6.11)$$

Com isso, um passo de *lifting* para predição passa a ser governado pela seguinte equação:

$$y_{1,i+1} = y_{1,i} - \pi^{\text{mod}}\{x_{1,i}\}, \text{ onde } \pi^{\text{mod}} = \sigma^\uparrow\pi\sigma_t^\downarrow\{x_{1,i}\} \text{ (ver equação 4.1).}$$

A figura 6.16 ilustra a correspondência, com relação a x_0 , entre as posições ocupadas pelas sinais $x_{1,i}$, $\sigma_{t_n}^\downarrow\{x_{1,i}\}$, $\sigma^\uparrow\sigma_{t_n}^\downarrow\{x_{1,i}\}$ e $y_{1,i}$. Repare o alinhamento existente entre os sinais $\sigma^\uparrow\sigma_{t_n}^\downarrow\{x_{1,i}\}$ e $y_{1,i}$.

Analogamente à modificação que fizemos nos passos de *lifting* para predição, criamos as operações de superamostragem ρ_t^\downarrow e subamostragem ρ^\uparrow definidas,

Figura 6.16: Correspondência entre as posições dos sinais: (a) x_0 , (b) $x_{1,i}$, (c) $\sigma_{t_n}^\downarrow\{x_{1,i}\}$, (d) $\sigma^\uparrow\sigma_{t_n}^\downarrow\{x_{1,i}\}$ e (e) $y_{1,i}$.

(a)	$x_0(0)$	$x_0(1)$	$x_0(2)$	$x_0(3)$	$x_0(4)$	$x_0(5)$	\cdots	$x_0(2n)$	$x_0(2n+1)$	\cdots
(b)	$x_{1,j}(0)$		$x_{1,j}(1)$		$x_{1,j}(2)$		\cdots	$x_{1,j}(n)$		\cdots
(c)	$x_{1,j}(0)$	t_0	$x_{1,j}(1)$	t_1	$x_{1,j}(2)$	t_2	\cdots	$x_{1,j}(n)$	t_n	\cdots
(d)		t_0		t_1		t_2	\cdots		t_n	\cdots
(e)		$y_{1,j}(0)$		$y_{1,j}(1)$		$y_{1,j}(2)$	\cdots		$y_{1,j}(n)$	\cdots

respectivamente, nas equações 6.12 e 6.13:

$$\begin{cases} \rho_t^\downarrow\{y\}(2n+1) = y(n) \\ \rho_t^\downarrow\{y\}(2n) = t \end{cases} \quad (6.12)$$

$$\rho^\uparrow\{y\}(n) = y(2n) \quad (6.13)$$

Assim, substituímos o operador λ dos passos de *lifting* para atualização pelo operador composto $\rho^\uparrow\lambda\rho_t^\downarrow$. A figura 6.17 ilustra a correspondência, com relação a x_0 , entre as posições ocupadas pelas sinais $y_{1,i}$, $\rho_{t_n}^\downarrow\{y_{1,i}\}$, $\rho^\uparrow\rho_{t_n}^\downarrow\{y_{1,i}\}$ e $x_{1,i}$. Repare o alinhamento existente entre os sinais $\rho^\uparrow\rho_{t_n}^\downarrow\{y_{1,i}\}$ e $x_{1,i}$.

Cabe observar que o valor de t usado nas operações σ_t^\downarrow e ρ_t^\downarrow depende do operador morfológico que será aplicado após a superamostragem. Por exemplo, sejam as operações abertura α_A e fechamento β_A em um reticulado completo \mathcal{L} . Se \top é o supremo de \mathcal{L} e \perp é o ínfimo, devemos fazer $\alpha_A\sigma_\top^\downarrow$ (e $\alpha_A\rho_\top^\downarrow$) e $\beta_A\sigma_\perp^\downarrow$ (e $\beta_A\rho_\perp^\downarrow$).

A partir de agora, apresentamos alguns operadores modificados, cujos resultados de testes são dados em seguida. Considere as definições abaixo.

Definição 1 (Operador γ_A^{mod}) Sejam α_A e β_A governados pelas equações 5.13

Figura 6.17: Correspondência entre as posições dos sinais: (a) x_0 , (b) $y_{1,i}$, (c) $\lambda\rho_{t_n}^\downarrow\{y_{1,i}\}$, (d) $\rho^\uparrow\lambda\rho_{t_n}^\downarrow\{y_{1,i}\}$ e (e) $x_{1,i}$.

(a)	$x_0(0)$	$x_0(1)$	$x_0(2)$	$x_0(3)$	$x_0(4)$	$x_0(5)$	\cdots	$x_0(2n)$	$x_0(2n+1)$	\cdots
(b)		$y_{1,j}(0)$		$y_{1,j}(1)$		$y_{1,j}(2)$	\cdots		$y_{1,j}(n)$	\cdots
(c)	t_0	$y_{1,j}(0)$	t_1	$y_{1,j}(1)$	t_2	$y_{1,j}(2)$	\cdots	t_n	$y_{1,j}(n)$	\cdots
(d)	t_0		t_1		t_2		\cdots	t_n		\cdots
(e)	$x_{1,j}(0)$		$x_{1,j}(1)$		$x_{1,j}(2)$		\cdots	$x_{1,j}(n)$		\cdots

e 5.14.

Nos passos de *lifting* para predição, o operador $\gamma_A^{\text{mod}} : \text{Fun}(\mathbb{Z}, \mathcal{L}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{L})$ é definido como segue:

$$\gamma_A^{\text{mod}} = \frac{\sigma^\uparrow\alpha_A\sigma_\top^\downarrow + \sigma^\uparrow\beta_A\sigma_\perp^\downarrow}{2} \quad (6.14)$$

Nos passos de *lifting* para atualização, o operador $\gamma_A^{\text{mod}} : \text{Fun}(\mathbb{Z}, \mathcal{L}) \rightarrow \text{Fun}(\mathbb{Z}, \mathcal{L})$ é definido como segue:

$$\gamma_A^{\text{mod}} = \frac{\rho^\uparrow\alpha_A\rho_\top^\downarrow + \rho^\uparrow\beta_A\rho_\perp^\downarrow}{2} \quad (6.15)$$

Definição 2 (Operador toggle contrast modificado) O operador *toggle contrast* modificado $\kappa_A^{\text{mod}} : \mathcal{L} \rightarrow \mathcal{L}$ é definido por

$$\kappa_A^{\text{mod}} = \begin{cases} \sigma^\uparrow\beta_A\sigma_\perp^\downarrow, & \text{se } \sigma^\uparrow\beta_A\sigma_\perp^\downarrow - \text{id} < \text{id} - \sigma^\uparrow\alpha_A\sigma_\top^\downarrow \\ \sigma^\uparrow\alpha_A\sigma_\top^\downarrow, & \text{nos demais casos} \end{cases} \quad (6.16)$$

nos passos de *lifting* para atualização, e por

$$\kappa_A^{\text{mod}} = \begin{cases} \rho^\uparrow \beta_A \rho_\perp^\downarrow, & \text{se } \rho^\uparrow \beta_A \rho_\perp^\downarrow - \text{id} < \text{id} - \rho^\uparrow \alpha_A \rho_\top^\downarrow \\ \rho^\uparrow \alpha_A \rho_\top^\downarrow, & \text{nos demais casos} \end{cases} \quad (6.17)$$

nos passos de *lifting* para atualização.

A tabela 6.6 mostra os resultados de testes com os operadores abaixo, usando elementos estruturantes com 2 e 3 u.c., para as taxas de quantização / codificação de 0,15bpp e 0,50bpp:

1. γ_A^{mod} , definido nas equações 6.14 e 6.15;
2. κ_A^{mod} , definido nas equações 6.16 e 6.17.

Observando a tabela, concluímos que os operadores de superamostragem e subamostragem só proporcionaram bons resultados quando utilizados com a operação γ_A . Por outro lado, o operador κ_A^{mod} gerou resultados ligeiramente piores que os de κ_A .

A figura 6.18 apresenta a imagem recuperada após processo de compressão / decompressão a uma taxa de 0,15bpp, onde foi usada uma decomposição em 5 níveis baseada nos coeficientes do projeto 3 e operador γ_A^{mod} com EE de comprimento 3. A figura 6.19 foi recuperada com uma taxa de 0,50bpp com os demais parâmetros da simulação anterior mantidos.

Veja que com esse operador modificado, não há mais o problema do efeito de blocos. Esse fato, juntamente com o aumento na PSNR nos permite dizer que, para a aplicação em questão, a decomposição não-linear morfológica gerada com o projeto 3, utilizando-se o operador γ_A^{mod} com $A = \{-1; 0; 1\}$, tem qualidade comparável à da wavelet linear Daubechies 9-7, amplamente usada na etapa de transformação do EZW por gerar bons resultados. A tabela 6.7 permite comparar as decomposições morfológica e linear.

Observando a tabela 6.7, vemos que, em parte dos cenários simulados, os resultados obtidos com a decomposição morfológica foram melhores que aqueles ge-

Tabela 6.6: Resultados da aplicação do EZW à imagem da figura 6.5 com a utilização do projeto 3. São apresentados resultados para os operadores morfológicos modificados γ_A^{mod} e κ_A^{mod} , EE com 2 e 3 u.c. e taxas de quantização / codificação de 0,15bpp e 0,50bpp.

operador	níveis decomp	EE (u.c.)	taxa (bpp)	PSNR (dB)
γ_A^{mod}	5	2	0,15	30.8488
κ_A^{mod}	5	2	0,15	27.7364
γ_A^{mod}	5	3	0,15	30.8488
κ_A^{mod}	5	3	0,15	27.7364
γ_A^{mod}	5	2	0,50	36.4073
κ_A^{mod}	5	2	0,50	33.0755
γ_A^{mod}	5	3	0,50	36.4073
κ_A^{mod}	5	3	0,50	33.0755

rados pela decomposição linear. Esse é um resultado surpreendente e é uma contribuição importante deste trabalho, pois deixa evidente que a técnica de decomposição apresentada no capítulo 4 (inicialmente proposta em [4] e aperfeiçoada neste trabalho) é capaz de gerar resultados tão bons e, às vezes, até melhores - em termos de PSNR - que os conseguidos com o uso de boas decomposições lineares, como é o caso da DWT Daubechies 9-7.

Vimos que é possível obter valores de PSNR maiores quando utilizamos (no EZW) uma decomposição morfológica ao invés de uma decomposição linear; contudo o leitor poderia perguntar: “Qual das duas decomposições faz com que a imagem reconstruída apresente melhor qualidade subjetiva, ou seja, menos artefatos?”. Para responder essa pergunta, apresentamos na figura 6.20(a) a imagem recuperada a partir da decomposição Daubechies 9-7 e na figura 6.20(b) a reconstrução feita a partir da decomposição morfológica do projeto 3, onde foi usado EE de comprimento 3 com o operador α_A^{mod} . Em ambos os casos, a decomposição foi feita em 5 níveis e a taxa de quantização / codificação foi de 0,15bpp. Comparando visualmente essas duas imagens, podemos perceber que ambas possuem artefatos: na imagem da



Figura 6.18: Imagem recuperada com o uso do projeto 3, operador γ_A^{mod} , EE com 3 u.c., 5 níveis de decomposição e taxa de 0,15bpp.

figura (a), podemos perceber a sua presença no lado esquerdo do chapéu, enquanto na figura (b) há mais artefatos na região da boca. Devido a esses e outros detalhes, julgamos que ambas (neste exemplo particular) possuem qualidade subjetiva semelhante, apesar da imagem da figura (b) ter PSNR menor que a da figura (a).

Os resultados encontrados neste capítulo evidenciam que é possível construir boas decomposições não-lineares utilizando operadores morfológicos. Percebemos que a abertura auto-dual definida em um inf-semireticulado completo não gerou bons resultados, ao contrário do que esperávamos. Apesar disso, todos os resultados obtidos com α'_A foram melhores que os obtidos com α_A , uma vez que o primeiro tende a conservar o valor médio por ser auto-dual.

Vemos algumas oportunidades de melhoria que podem ser incorporadas em trabalhos futuros. Podemos, assim como foi feito neste trabalho com o método



Figura 6.19: Imagem recuperada com o uso do projeto 3, operador γ_A^{mod} , EE com 3 u.c., 5 níveis de decomposição e taxa de 0,50bpp.

de projeto de decomposições unidimensionais, formalizar o método de projeto de decomposições bidimensionais não-separáveis proposto em [4] utilizando o sistema axiomático de Heijmans e Goutsias [9,10]. Além disso, também como fizemos neste trabalho, podemos realizar melhorias neste método (bidimensional não-separável), de forma a adequá-lo melhor a uma dada aplicação. Como já foi dito, também é possível modificarmos as etapas de quantização e codificação do EZW para que possamos utilizá-las, de forma otimizada, em conjunto com uma decomposição wavelet morfológica projetada neste trabalho. Finalmente, podemos investigar outros operadores morfológicos auto-duais, ou outros que tendem a conservar o valor médio.

Tabela 6.7: Comparação entre a decomposição linear obtida com a wavelet Daubechies 9-7 e a decomposição morfológica obtida com o projeto 3, onde o operador utilizado foi o $\mathcal{L}_B\{.\}$ com EE de comprimento 3.

níveis decomp	taxa (bpp)	PSNR (dB; projeto3)	PSNR (dB; Daub. 9-7)
5	0,15	30,85	31,01
5	0,50	36,41	36,14
6	0,15	31,18	30,85
6	0,50	35,96	36,05



(a)



(b)

Figura 6.20: Imagem recuperada após decomposição morfológica e linear, ambas com 5 níveis, e taxa de quantização de 0,15bpp. (a) Decomposição linear resultante do uso da wavelet Daubechies 9-7. (b) Decomposição morfológica do projeto 3, EE de comprimento 3 e operador γ_A^{mod} .

Capítulo 7

Conclusão

Em [4], Cardoso apresentou um método para projeto de Transformadas Wavelets não-lineares, com a utilização de passos de *lifting* e operadores morfológicos [6–8]. Neste trabalho, começamos por formalizar tal método utilizando, para isto, a notação e o sistema axiomático desenvolvidos por Heijmans e Goutsias em [9, 10].

O método originalmente consiste em solucionar três equações, e o número de coeficientes (incógnitas) pode variar de acordo com o número de passos de *lifting* utilizado. Com mais de dois passos completos, vimos que a solução do método é sobredeterminada, o que nos permitiu propor uma modificação no sentido restringir o campo de busca de solução de modo a otimizá-la para uma dada aplicação. A função objetivo escolhida para a aplicação, baseada em transformadas de Fourier das funções de transferência das bandas de aproximação e detalhes, tende a minimizar a energia da banda de rejeição.

Vimos que a fatoração da DWT (*Discrete Wavelet Transform*) linear Daubechies 9-7 gera coeficientes que satisfazem as equações do método de projeto modificado, além de minimizar a função objetivo escolhida. Decidimos usar esse conjunto de coeficientes para a construção das decomposições wavelets não-lineares, pois, ao fazer comparações com a DWT linear Daubechies 9-7, saberíamos quais as vantagens e desvantagens em usarmos operadores morfológicos, ao invés de lineares, nos passos de *lifting*.

O método de projeto descrito no capítulo 4 requer que os operadores usados

nos passos de *lifting* sejam suavizadores e conservem o valor médio do sinal sobre o qual eles atuam. Com essa necessidade em mente, a propriedade da auto-dualidade de operadores morfológicos foi bastante explorada, pois acreditamos que ela, aproximadamente, garante aos operadores que a possuem a conservação do valor médio do sinal de entrada.

Foram apresentados os conceitos de reticulados completos e inf-semireticulados completos (*cisl*'s). Vimos que os primeiros possuem uma limitação à construção de operadores intrinsecamente auto-duais; porém, combinações inteligentes de operadores padrão podem fazer com que se consiga satisfazer a propriedade de auto-dualidade. Já os inf-semireticulados completos são conjuntos que permitem a definição natural de operadores intrinsecamente auto-duais; porém, eles possuem uma limitação: a possibilidade da não-existência de supremos, que reduz bastante o conjunto-domínio de sinais que permitem aplicação de tais operadores; basicamente, somente a erosão e a abertura em *cisl*'s podem ser aplicadas nos sinais que usamos neste trabalho.

Para testar as decomposições projetadas, as aplicamos na etapa de transformação do algoritmo EZW para compressão de imagens, em substituição a uma DWT linear; as decomposições foram consideradas tanto melhores quanto maior o valor de PSNR calculado com a imagem reconstruída. Para isso, foi apresentada uma descrição sucinta do princípio de funcionamento do algoritmo EZW. Além disso, foi mostrado como realizar decomposições bidimensionais a partir de wavelets unidimensionais, já que as decomposições geradas neste trabalho são unidimensionais.

Nas simulações, foram testadas algumas soluções de projeto e a influência de alguns parâmetros na qualidade das decomposições geradas, como, por exemplo, o número de níveis de decomposição, diferentes operadores morfológicos e diversos elementos estruturantes.

Com os diversos cenários de simulações, concluímos alguns fatos importantes. Primeiramente, há um número ótimo de níveis de decomposição que maximiza os valores de PSNR das imagens recuperadas após o processo de compressão/decompressão. Em segundo lugar, percebemos que o elemento estruturante também desempenha um papel importante na qualidade das decomposições; quanto

maior o tamanho do EE, pior é o resultado das decomposições. Isto acontece, pois com EE grandes, o cálculo do valor de uma amostra é baseada nos valores de vizinhos distantes, o que faz com que o erro de quantização seja grande; além disso, esse erro é propagado através das bandas quando da sua reconstrução gerando um sinal reconstruído com muitos artefatos. A compreensão deste fato foi outra contribuição desse trabalho. Finalmente, vimos que nem todas as operações utilizadas em passos de *lifting* geram bons resultados. Em algumas delas, isso acontece por não satisfazerem a característica de conservação do valor médio, como por exemplo a abertura α_A ; em outras, o desempenho foi prejudicado pelo fato dos sinais das bandas de aproximação e detalhes estarem espacialmente (ou temporalmente) desalinhados com referência ao sinal original que está sendo decomposto, fazendo com que as predições e atualizações executadas em uma banda a partir da outra (passos de *lifting*) não sejam eficientes.

Com intenção de melhorar as predições e atualizações executadas nos passos de *lifting*, apresentamos um artifício que utiliza operações de superamostragem e subamostragem de forma a alinhar os sinais que são envolvidos nestes cálculos. Esse artifício só gerou bons resultados quando associados à operação γ_A .

Por fim, concluímos que as decomposições projetadas com os método de projeto apresentado são capazes de gerar resultados tão bons quanto os tradicionalmente encontrados na literatura, tanto do ponto de vista subjetivo (visual) quanto do objetivo (PSNR). O EE desempenha um papel fundamental nessas decomposições e sua escolha deve levar em conta os comentários e interpretações dados anteriormente.

Como alternativa para trabalhos futuros, podemos, assim como foi feito neste trabalho com o método de projeto de decomposições unidimensionais, formalizar a descrição do método de projeto de decomposições bidimensionais não-separáveis proposto em [4] utilizando o sistema axiomático de Heijmans e Goutsias [9, 10]. Além disso, também como foi feito neste trabalho, podemos realizar melhorias neste método (bidimensional não-separável), de forma a adequá-lo melhor a uma dada aplicação. Como já foi dito, também é possível modificarmos as etapas de quantização e codificação do EZW para que possamos utilizá-las, de forma otimizada, em conjunto com uma decomposição wavelet morfológica projetada neste trabalho. Fi-

nalmente, podemos investigar outros operadores morfológicos auto-duais, ou outros que tendem a conservar o valor médio.

Referências Bibliográficas

- [1] HEIJMANS, H. J. A. M., (KRESCH), R. K., *Inf-Semilattice Approach to Self-Dual Morphology*, Report, Centrum voor Wiskunde en Informatica (CWI), 2001.
- [2] DINIZ, P. S. R., da Silva, E. A. B., NETTO, S. L., *Digital Signal Processing: System Analysis and Design*. 1 ed. Cambridge University Press, 2003.
- [3] VAIDYANAYHAN, P. P., *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [4] JÚNIOR, E. S. C., *Compressão de Imagens Utilizando Decomposições em Multiresolução Morfológicas*. M.Sc. dissertation, COPPE, UFRJ, Março 1999.
- [5] DAUBECHIES, I., SWELDENS, W., *Factoring Wavelet Transforms into Lifting Steps*, Report, Bell Laboratories - Lucent Technologies, 1996.
- [6] SOILLE, P., *Morphological Image Analysis: Principles and Applications*. 2 ed. Springer, 1999.
- [7] HEIJMANS, H. J. A. M., *Morphological Image Operators*. Academic Press, 1994.
- [8] FACON, J., *Morfologia Matemática: Teoria e Exemplos*. 1 ed. Editora Universitária da PUC do Paraná, 1996.
- [9] GOUTSIAS, J., HEIJMANS, H. J. A. M., *Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids.*, Report, Centrum voor Wiskunde en Informatica (CWI), 1998.

- [10] GOUTSIAS, J., HEIJMANS, H. J. A. M., *Multiresolution signal decomposition schemes. Part 2: Morphological wavelets.*, Report, Centrum voor Wiskunde en Informatica (CWI), 1999.
- [11] NEVES, M. A., *Decomposição Morfológica de Sinais em Multiresolução*, Projeto final, UFRJ, DEL, Março 2004.
- [12] MALLAT, S. G., *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [13] HEIJMANS, H. J. A. M., (KRESCH), R. K., “First Steps Towards a Self-Dual Morphology”, *ICIP*, , 2000.
- [14] SAYOOD, K., *Introduction to Data Compression*. 2 ed. Morgan Kaufmann, 2000.
- [15] SHAPIRO, J. M., “Embedded Image Coding using Zerotrees of Wavelet Coefficients”, *IEEE Transaction Signal Processing*, v. 41, December 1993.
- [16] ORTEGA, A., RAMCHANDRAN, K., “Rate-Distortion Methods for Image and Video Compression”, *IEEE Signal Processing Magazine*, , November 1998.
- [17] SHANNON, C. E., “A Mathematical Theory of Communication”, *Bell System Technical Journal*, , 1948.
- [18] COVER, T. M., THOMAS, J. A., *Elements of Information Theory*. Wiley-Interscience, 1991.