



IV Colóquio de Matemática  
da Região Sul

# TRANSFORMADA WAVELET DE HAAR: CONCEITOS, FORMULAÇÕES E APLICAÇÕES

**THIAGO L. T. DA SILVEIRA**  
**ALICE J. KOZAKEVICIUS**



**SBM**

SOCIEDADE BRASILEIRA DE MATEMÁTICA

# **Transformada wavelet de Haar:** conceitos, formulações e aplicações





IV Colóquio de Matemática  
da Região Sul

# **TRANSFORMADA WAVELET DE HAAR: CONCEITOS, FORMULAÇÕES E APLICAÇÕES**

**THIAGO L. T. DA SILVEIRA  
ALICE J. KOZAKEVICIUS**

**1ª EDIÇÃO  
2016  
RIO GRANDE**



**SBM**  
SOCIEDADE BRASILEIRA DE MATEMÁTICA

Dedicado a todos os estudantes interessados em  
desvendar os encantos e desafios das funções  
*wavelets*, suas transformadas e suas mais diversas  
aplicações. Que este material possa servir de  
inspiração e motivação para iniciar esta jornada.

# Sumário

<b>1</b>	<b>Transformada <i>wavelet</i> discreta</b>	<b>3</b>
1.1	Transformada <i>wavelet</i> de Haar 1D . . . . .	4
1.1.1	Formulação Algorítmica para a TWD de Haar . . . . .	4
1.1.2	Formulação Matricial para a TWD de Haar . . . . .	7
1.1.3	TWD de Haar 1D Ortonormal . . . . .	11
1.1.4	Análise Multiresolução . . . . .	12
1.1.5	Base de Haar . . . . .	14
1.1.6	Relação de Escala e Relação <i>Wavelet</i> . . . . .	16
1.1.7	Expansão de funções em $V_J$ . . . . .	18
1.1.8	Projeções em $V_j$ . . . . .	21
<b>2</b>	<b>Aplicações 1D</b>	<b>23</b>
2.1	Propriedade dos Momentos Nulos . . . . .	23
2.1.1	Decaimento dos coeficientes <i>wavelet</i> . . . . .	24
2.2	Operações de truncamento . . . . .	26
2.2.1	Operadores de Truncamento . . . . .	27
2.3	Experimentos Computacionais . . . . .	30
2.4	Filtragem <i>wavelet</i> . . . . .	33
2.5	Análise de Sinais EEG . . . . .	35
<b>3</b>	<b>TWD 2D de Haar</b>	<b>39</b>
3.1	Base para $L^2(\mathbb{R}^2)$ . . . . .	39
3.1.1	Algoritmo <i>standard</i> da TWD 2D . . . . .	41
3.1.2	Algoritmo não- <i>standard</i> da TWD 2D . . . . .	42
3.2	Aplicação em análise de imagens . . . . .	44
3.2.1	Filtragem de imagens . . . . .	44
3.2.2	Compressão de imagens . . . . .	45
<b>4</b>	<b>Transformada <i>wavelet</i> Packet de Haar</b>	<b>47</b>
4.1	Formulação . . . . .	47
4.1.1	Algoritmo <i>Best-Basis</i> e Função de Custo . . . . .	49
4.2	Aplicação em Análise de Sinais . . . . .	51



# Prefácio

Funções *wavelet* têm, ao longo dos últimos cinquenta anos, despertado interesse em ambos matemáticos teóricos e aplicados [23]. O que diferencia as transformadas *wavelet* de outras “ferramentas matemáticas” é sua habilidade de decompor hierarquicamente funções [31], permitindo analisá-las em diferentes escalas e diferentes regiões de seu domínio ao mesmo tempo. Embora as *wavelets* tenham suas raízes na teoria das aproximações e processamento de sinais [23], elas têm sido exploradas e utilizadas como parte fundamental de vários esquemas numéricos associados às mais diferentes aplicações. Alguns exemplos são os métodos para resolução adaptativa de equações diferenciais [6], para análise e classificação de sinais biológicos [30, 29, 1], para detecção de anomalias em redes de computadores [19] e para reconstrução de cenas tri-dimensionais a partir de imagens bidimensionais [5].

A proposta deste minicurso é apresentar, através da Álgebra Linear, os conceitos fundamentais da teoria de transformadas *wavelets* ortogonais discretas, a partir de sua representante mais simples: a transformada *wavelet* de Haar [31, 12]. Neste minicurso, o conceito fundamental de transformação linear e sua representação matricial serão o ponto de partida para a apresentação das principais propriedades da transformada *wavelet* de Haar, que, na verdade, são compartilhadas por todas as demais funções *wavelets* ortogonais integrantes da família de Daubechies [12].

Espera-se instigar os participantes deste minicurso a se aprofundarem mais sobre o assunto, percebendo a relevância de disciplinas como Álgebra Linear, Algoritmos e Métodos Numéricos para a construção de uma base sólida de conhecimento. Para ilustrar o potencial da transformada *wavelet* de Haar em diferentes aplicações, exemplos uni e bidimensionais serão exibidos.





## Capítulo 1

# Transformada *wavelet* discreta

De acordo com Stollnitz *et al.* [31], funções *wavelets* e suas transformadas são ferramentas matemáticas para análise e decomposição hierárquica de funções, ou seja, representação da função em diferentes níveis de resolução, ou escalas. Desta forma, na escala mais grosseira a função (que pode ser tanto um sinal, uma imagem ou um conjunto qualquer de dados) é representada por uma "curva" média, e em cada escala mais refinada são, então, acrescentados detalhes complementares que, somados ao padrão médio, reconstroem exatamente a função original. Assim como senos e cossenos da transformada de Fourier [4], as *wavelets* servem como base para representação de outras funções [33]. Em especial, as transformadas *wavelet* mostram-se mais eficientes em relação a de Fourier quando os sinais a serem analisados são não-periódicos [2], como é o caso de vários sinais biomédicos, como eletrocardiogramas (ECG) [18] e eletroencefalogramas (EEG) [30].

Transformadas *wavelets* uni (1D) e bidimensionais (2D) são utilizadas em diversas áreas de pesquisa, como processamento digital de sinais, diversos métodos para resolução numérica de equações diferenciais parciais, compressão de imagens e análise estatística [2]. O uso destas transformadas 1D e 2D no processamento de sinais e imagens permite, por exemplo, a visualização e extração de importantes parâmetros que podem ser direta ou indiretamente utilizados em aplicações de reconhecimento e classificação de padrões [25, 1].

Nas últimas décadas, uma das famílias de funções *wavelet* mais difundidas e utilizadas para a obtenção de transformadas ortonormais é a de Daubechies [9], cujas propriedades fundamentais são indispensáveis para muitas aplicações. Neste minicurso, muitas destas propriedades serão exploradas via exemplos através da representante mais simples da família de Daubechies, denominada *wavelet* de Haar. Na verdade, esta base de funções foi definida por Haar em 1910 [16], muito antes da teoria sobre *wavelets* ter sido estruturada por Daubechies, quando as funções de Haar foram então reconhecidas e classificadas como funções *wavelets* ortonormais.

## 1.1 Transformada *wavelet* de Haar 1D

Neste trabalho, assim como em textos introdutórios sobre *wavelets* [33, 31], a transformada *wavelet* discreta (TWD) de Haar é primeiramente apresentada via exemplo. Como consequência, são dadas formulações algorítmicas para as transformadas 1D direta e inversa. Depois, através de conceitos básicos de Álgebra Linear são obtidas as formulações matriciais correspondentes.

Além disso, através das relações entre diferentes escalas é apresentada a estrutura de análise de multiresolução associada a todas as funções *wavelets* ortonormais da família de Daubechies, tomando-se como exemplo o caso particular associado a de Haar.

### 1.1.1 Formulação Algorítmica para a TWD de Haar

Considere  $C_j$  um vetor de dimensão (tamanho)  $2^j = 2^2$ ,

$$C_j = C_2 = [2 \ 0 \ 1 \ 5].$$

O vetor  $C_2$  pode ser visto como uma coleção de valores  $C_{2,k}$ ,  $k = 0, 1, 2, 3$ , que podem ser graficados de acordo com a Figura 1.1(a), formando então um primeiro exemplo de sinal discreto 1D, ou seja, uma função real a valores reais, cujo domínio é um conjunto discreto.

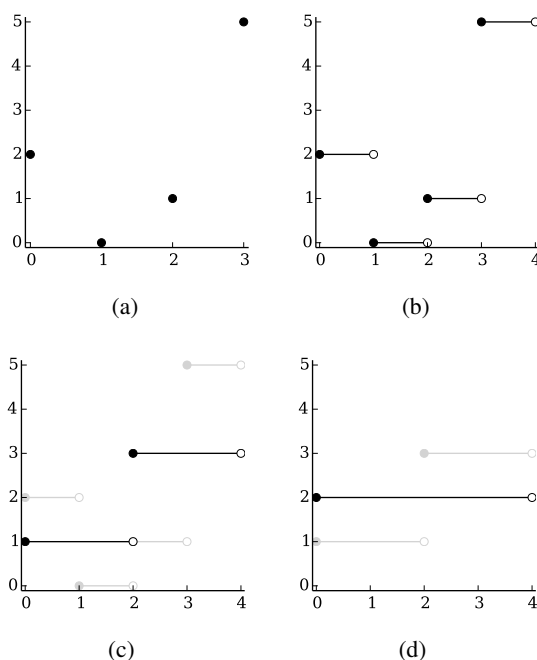


Figura 1.1: Vetor  $C_2$  visto como (a) uma sequência de pontos e (b) função constante por partes. Vetores (c)  $C_1$  e (d)  $C_0$  vistos como funções constantes por partes.

Entretanto, uma outra possibilidade é interpretar que cada um dos elementos do vetor  $C_2$  seja a imagem de uma função constante por partes (função escada), como ilustrado na Figura 1.1(b). Essa segunda representação é particularmente útil, como será visto na Seção 1.1.5, e portanto será adotada ao longo deste texto.

Agora, dentre as inúmeras possibilidades de se manipular os elementos de  $C_2$ , estes podem ser transformados (modificados) da seguinte maneira: calcula-se a média aritmética – dois a dois de forma disjunta – de todos os elementos de  $C_2$ :

$$C_{1,0} = \frac{C_{2,0} + C_{2,1}}{2}, \quad C_{1,1} = \frac{C_{2,2} + C_{2,3}}{2}. \quad (1.1)$$

Para este exemplo tem-se  $C_1 = [1 \ 3]$  como sendo o vetor com os valores resultantes desta operação, cuja representação através de funções constantes por partes é dada na Figura 1.1(c). Notoriamente  $C_1$  não mais tem a mesma dimensão de  $C_2$ . Na verdade, o tamanho de  $C_1$  é a metade do tamanho de  $C_2$ ,  $(\dim C_1) = (\dim C_2)/2$ . Naturalmente, seria possível aplicar a mesma transformação aos dados (elementos) de  $C_1$ , e novamente armazenar seus resultados em um novo vetor, que neste caso teria apenas um elemento, denotado por  $C_0 = [2]$ , representado pela Figura 1.1(d).

De forma geral, partindo de um vetor inicial  $C_J$ , com  $\dim C_J = 2^J$ , a sequência de transformações aplicadas aos vetores  $C_j$ , com  $j$  variando de  $J$  até 1, na qual apenas as médias de valores dois a dois disjuntos são calculadas, poderia ser escrita em forma de algoritmo como segue:

$$C_{j-1,i} = (C_{j,2i} + C_{j,2i+1})/2, \quad i = 0, \dots, 2^{j-1} - 1 \quad (1.2)$$

Na Expressão 1.2, o índice  $i$  indica a posição dos elementos do novo vetor de informações médias ou coeficientes de escala ( $C_{j-1}$ ), em cada nível de resolução  $j$ . Comparando as Figuras 1.1(b), 1.1(c) e 1.1(d), percebe-se que há perda de informação quando um vetor  $C_j$  é substituído simplesmente por sua informação média  $C_{j-1}$ . No entanto, esta ideia é amplamente utilizada em aplicações que necessitam de redução de dimensão. Um exemplo comum é o envio de imagens ou *streaming* de vídeo através da Internet. Determinadas aplicações podem não requerer imagens (ou vídeos) em alta resolução, permitindo aplicar algoritmos similares ao 1.2, sem degradação notável da informação essencial e possibilitando a transferência desses dados de forma muito mais rápida [22].

Por outro lado, de acordo com o que foi visto até o momento para esta transformação definida a partir do cálculo das médias, uma vez que a substituição de um vetor de dimensão maior  $C_j$  por um outro de dimensão menor tenha sido feita, é impossível recuperar os valores de  $C_j$  a partir dos dados reduzidos. Para que se possa realizar uma reconstrução exata dos dados originais (transformação inversa) é preciso que entre cada dois níveis consecutivos quaisquer,  $j$  e  $j - 1$ , seja também considerado um vetor  $D_{j-1}$  de informações complementares (denominado de vetor de detalhes ou coeficientes wavelets) entre  $C_j$  e  $C_{j-1}$ . Assim, os elementos do vetor  $D_{j-1}$  são obtidos da seguinte maneira:

$$D_{j-1,i} = C_{j,2i} - C_{j-1,i}, \quad i = 0, 1, 2, \dots, 2^{j-1} - 1. \quad (1.3)$$

Ao processo de encontrar as médias e diferenças em relação aos valores de um vetor  $C_j$ , gerando-se  $C_{j-1}$  e  $D_{j-1}$  respectivamente, dá-se o nome de transformada (decomposição) *wavelet* de Haar em um nível. Se esse procedimento for reaplicado, agora para os demais níveis  $j - 1, \dots, 1$ , construindo-se, além das médias, os vetores de detalhes  $D_{j-2}, \dots, D_0$ , obtém-se, então, uma decomposição completa entre o nível mais fino  $j = J$  e o mais grosseiro  $j = 0$ . A esta transformação envolvendo todos os níveis possíveis de decomposição de um vetor original  $C_J$  denomina-se transformada *wavelet* discreta de Haar unidimensional (TWD de Haar 1D), cujo algoritmo completo para  $J$  níveis de decomposição é dado a seguir.

---

**Algoritmo 1** TWD de Haar 1D

---

- 1: **Para**  $j \leftarrow J$  **até** 1 **faça** ▷ Nível de resolução  $j$
  - 2:     **Para**  $i \leftarrow 0$  **até**  $2^{j-1} - 1$  **faça** ▷ Posição dos elementos de  $C_{j-1}$  e  $D_{j-1}$
  - 3:          $C_{j-1,i} \leftarrow (C_{j,2i} + C_{j,2i+1})/2$
  - 4:          $D_{j-1,i} \leftarrow C_{j,2i} - C_{j-1,i}$
  - 5:     **Fim Para**
  - 6: **Fim Para**
- 

Retomando o exemplo inicial, têm-se  $D_{j-1} = D_1 = [1 \ -2]$  como o vetor de detalhes calculado em função de  $C_2$  e  $C_1$ . Considerando agora  $C_1$  como sendo o próximo (e último) nível a ser decomposto, obtém-se  $C_0 = [2]$  e  $D_0 = [-1]$ , que são os vetores contendo a média e o detalhe no último nível da TWD de Haar 1D.

A partir dessa transformação completa em vários níveis, consegue-se associar o vetor  $C_2 = [2 \ 0 \ 1 \ 5]$  a suas duas componentes  $C_1 = [1 \ 3]$  e  $D_1 = [1 \ -2]$ . Note que ambos  $C_1$  e  $D_1$  possuem dimensão metade da dimensão de  $C_2$ , o que possibilitaria o armazenamento destas informações novamente em um único vetor com tamanho igual ao de  $C_2$ , a saber  $[1 \ 3 \ 1 \ -2]$ , cujas coordenadas permitem a reconstrução de  $C_2$ . Como  $C_1$  por sua vez também foi decomposto em componentes  $C_0 = [2]$  e  $D_0 = [-1]$ , o vetor  $C_1$  também poderia ser associado a um vetor equivalente, neste caso  $[2 \ -1]$ . Desta forma, substituindo-se essa nova representação de  $C_1$  na representação equivalente associada a  $C_2$ , este poderia então ser representado por  $[2 \ -1 \ 1 \ -2]$ : que são a informação média do último nível de decomposição e todas as demais informações de detalhes, ou seja,  $C_2$  é transformado (mapeado), preservando sua dimensão, no vetor  $[C_0 \ D_0 \ D_1]$  composto por cada um dos blocos da TWD de Haar 1D,  $C_2 \rightarrow [C_0 \ D_0 \ D_1]$ .

No caso geral, de um vetor inicial no nível  $J$  (de dimensão  $2^J$ ) teríamos  $C_J \rightarrow [C_0 \ D_0 \ D_1 \ \dots \ D_{J-1}]$  que é chamada representação multiresolução de um sinal inicial definido por  $C_J$ . Neste primeiro momento, esta representação multiresolução pode ser interpretada apenas como uma forma econômica de se armazenar todos os dados obtidos ao longo de todos os  $J$  níveis da TWD de Haar 1D no mesmo vetor de entrada  $C_J$ , evitando a definição de vetores auxiliares para armazenamento dos dados intermediários ao longo de todos os níveis de decomposição.

Note que, o processo descrito anteriormente é reversível, uma vez que os dados originais  $C_j$  de cada nível  $j$  podem ser reconstruídos a partir das médias e dos deta-

lhes do nível imediatamente mais grosseiro  $C_{j-1}$  e  $D_{j-1}$ . Assim, considerando-se  $C_{j-1}$  e  $D_{j-1}$  como dados de entrada, pode-se definir a DWT de Haar 1D Inversa para reconstruir dados dos níveis 1 até  $J$  através do seguinte algoritmo:

---

**Algoritmo 2** TWD de Haar 1D Inversa
 

---

```

1: Para  $j \leftarrow 0$  até  $J - 1$  faça                                 $\triangleright$  Nível de resolução  $j$ 
2:   Para  $i \leftarrow 0$  até  $2^{j-1} - 1$  faça       $\triangleright$  Posição dos elementos de  $C_{j-1}$  e  $D_{j-1}$ 
3:      $C_{j,2i} \leftarrow C_{j-1,i} + D_{j-1,i}$ 
4:      $C_{j,2i+1} \leftarrow C_{j-1,i} - D_{j-1,i}$ 
5:   Fim Para
6: Fim Para
  
```

---

Para o exemplo dado inicialmente, assumindo-se  $C_0 = [2]$  e  $D_0 = [-1]$  como dados de entrada para o Algoritmo 2, reconstrói-se exatamente  $C_1 = [1 \ 3]$ , sendo  $C_{1,0} = C_{0,0} + D_{1,0} = 2 - 1 = 1$  e  $C_{1,1} = C_{0,0} - D_{1,0} = 2 + 1 = 3$ . Analogamente, a reconstrução de  $C_2 = [2 \ 0 \ 1 \ 5]$  através do Algoritmo 2 se dá através de  $C_1 = [1 \ 3]$  e  $D_1 = [1 \ -2]$  como sendo os novos dados de entrada. Dessa forma, tem-se  $[C_0 \ D_0 \ D_1] \rightarrow C_2$ . E no caso geral,  $[C_0 \ D_0 \ D_1 \ \dots \ D_{J-1}] \rightarrow C_J$ . Portanto a representação multiresolução é, na verdade, uma forma equivalente de representação do sinal  $C_J$ ,  $C_J \sim [C_0 \ D_0 \ D_1 \ \dots \ D_{J-1}]$ .

Os Algoritmos 1 e 2 são também denominados algoritmos de Cascata [22] para a versão decimada da TWD da Haar 1D direta e inversa, respectivamente. A operação de decimação é a responsável pela redução de dimensão num fator de 2 a cada nível de decomposição. Formulações que preservam o tamanho dos vetores ao longo da decomposição são denominadas não decimadas. Uma possibilidade é a formulação através do algoritmo *à trous* [26, 20]. No entanto, para as formulações não decimadas, diferentes tipos de transformações inversas são possíveis. Para informações mais detalhadas sobre transformadas decimadas, não decimadas e suas inversas são sugeridos os textos [9, 22, 26].

### 1.1.2 Formulação Matricial para a TWD de Haar

O objetivo principal agora é retomar o exemplo da Seção 1.1.1, apresentando-o sob o ponto de vista de transformações lineares. Todos os conceitos considerados de Álgebra Linear serão abordados conforme textos introdutórios, vistos em cursos de graduação, como por exemplo na referência [3]. Desta forma as formulações matriciais para as TWD de Haar 1D direta e inversa são obtidas naturalmente.

#### Transformação S para as médias

Consideram-se, portanto, os espaços vetoriais  $V = \mathbb{R}^4$  e  $W = \mathbb{R}^2$ . Define-se a transformação linear  $S : V \rightarrow W$  como sendo a aplicação que associa a cada

vetor  $C_2 \in V$  um outro vetor  $C_1 = S(C_2) \in W$  através da seguinte lei:

$$S(C_2) = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} C_{2,0} \\ C_{2,1} \\ C_{2,2} \\ C_{2,3} \end{bmatrix} = \begin{bmatrix} \frac{C_{2,0}+C_{2,1}}{2} \\ \frac{C_{2,2}+C_{2,3}}{2} \end{bmatrix} = \begin{bmatrix} C_{1,0} \\ C_{1,1} \end{bmatrix} \quad (1.4)$$

Comparando-se as coordenadas do vetor  $C_1$  (imagem de  $C_2$  pela transformação linear  $S$  dada pela Equação 1.4) com os valores referentes às médias calculadas para a transformada de Haar dados pela Equação 1.1, da subseção anterior, verifica-se que ambas expressões são iguais. Com isso, a matriz  $[S] = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$ , associada à transformação linear  $S$ , é na verdade a forma matricial que representa a decomposição em relação às médias em um nível, a partir de um vetor inicial com 4 elementos. Caso o vetor inicial  $C_J$  possuisse  $2^J$  elementos, ou seja,  $C_J \in \mathbb{R}^n$ , com  $n = 2^J$ , então a matriz  $[S_J]$  teria dimensão  $2^{J-1} \times 2^J$  e lei de formação similar, com pesos  $1/2$  para as duas posições da linha que produzirão a média e zero para as demais.

Como a dimensão do domínio  $V$  é maior do que a dimensão do contra-domínio  $W$ ,  $\dim V = 4 > \dim W = 2$  (no caso geral,  $\dim V = 2^J > \dim W = 2^{J-1}$ ), a transformação  $S$  não é injetora e portanto fica caracterizada a redução de dimensionalidade na transição de  $V$  para  $W$ . Essa redução representa a perda de informação na passagem de dados associados a  $C_2$  para valores médios definidos em  $C_1$ , (ou, de  $C_j$  para  $C_{j-1}$ , com  $j = 1, 2, \dots, J$ ), como já observado na subseção anterior.

## Transformação T para os detalhes

Além de  $S$ , pode-se definir uma segunda transformação linear  $T : V \rightarrow W$ , que a cada vetor  $C_2 \in V$  associa o vetor  $D_1 = T(C_2) \in W$  através da lei correspondente ao que foi definido anteriormente para o cálculo dos detalhes, Equação 1.3, sendo os valores de  $C_{j-1}$  substituídos pela Equação 1.2. Aqui neste exemplo para  $j = 2$ , tem-se:

$$T(C_2) = \begin{bmatrix} 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} C_{2,0} \\ C_{2,1} \\ C_{2,2} \\ C_{2,3} \end{bmatrix} = \begin{bmatrix} \frac{C_{2,0}-C_{2,1}}{2} \\ \frac{C_{2,2}-C_{2,3}}{2} \end{bmatrix} = \begin{bmatrix} D_{1,0} \\ D_{1,1} \end{bmatrix} \quad (1.5)$$

A transformação  $T$  também não é injetora e sua matriz associada é  $[T] = \begin{bmatrix} 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \end{bmatrix}$ .

### Transformação de Haar - Forma Matricial para 1 nível

Define-se agora a transformação linear  $Q_2 : V \rightarrow V$ , construída a partir de  $S$  e  $T$ , cuja formulação matricial é dada por:

$$Q_2(C_2) = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} C_{2,0} \\ C_{2,1} \\ C_{2,2} \\ C_{2,3} \end{bmatrix} = \begin{bmatrix} \frac{C_{2,0}+C_{2,1}}{2} \\ \frac{C_{2,2}+C_{2,3}}{2} \\ \frac{C_{2,0}-C_{2,1}}{2} \\ \frac{C_{2,2}-C_{2,3}}{2} \end{bmatrix} = \begin{bmatrix} C_{1,0} \\ C_{1,1} \\ D_{1,0} \\ D_{1,1} \end{bmatrix} \quad (1.6)$$

Observa-se que agora  $Q_2$  é uma matriz quadrada (domínio e contra-domínio da transformação têm mesma dimensão). Além disso, o núcleo de  $Q_2$  é apenas o vetor nulo,  $\ker\{Q_2\} = \vec{0}$ . Sendo assim,  $Q_2$  é uma transformação injetora e sobrejetora, o que caracteriza  $Q_2$  como sendo uma transformação inversível.

Assim,  $[Q_2]$ , a matriz associada à transformação  $Q_2$ , é uma das diferentes formulações matriciais que podem representar a TWD de Haar 1D para um nível apenas de decomposição. Neste exemplo,  $Q_2$  representa um algoritmo ligeiramente diferente do Algoritmo 1. Agora todas as médias são calculadas primeiro, para após serem calculadas todas as diferenças.

No caso geral,  $V = \mathbb{R}^n$ ,  $n = 2^J$ , a matriz  $Q_J$  é então formada pelos dois blocos anteriormente construídos:  $S_J$  para as médias e  $T_J$  para os detalhes, cujos subíndices acrescentados à notação apenas indicam a dimensão dos blocos, gerando desta forma a matriz da transformação:

$$Q_J = \begin{bmatrix} S_J \\ T_J \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & -1/2 \end{bmatrix} \quad (1.7)$$

Para obter-se uma representação matricial fiel ao Algoritmo 1 para um nível de decomposição, a transformação a ser considerada deve ser:

$$H(C_2) = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} C_{2,0} \\ C_{2,1} \\ C_{2,2} \\ C_{2,3} \end{bmatrix} = \begin{bmatrix} \frac{C_{2,0}+C_{2,1}}{2} \\ \frac{C_{2,0}-C_{2,1}}{2} \\ \frac{C_{2,2}+C_{2,3}}{2} \\ \frac{C_{2,2}-C_{2,3}}{2} \end{bmatrix} = \begin{bmatrix} C_{1,0} \\ D_{1,0} \\ C_{1,1} \\ D_{1,1} \end{bmatrix} \quad (1.8)$$

Na prática, o Algoritmo 1 é mais empregado do que sua formulação matricial  $H$ .



### Transformação de Haar - Forma Matricial para $j$ níveis

No entanto, para que se possa construir uma formulação matricial para a TWD de Haar 1D para vários níveis de decomposição, a representação matricial através da transformação linear  $Q_2$  (ou no caso geral  $Q_J$ ) oferece mais flexibilidade. Isto porque a imagem  $Q_2(C_2)$  (o vetor de saída) fica separada em dois blocos:  $C_1$  para as informações médias e  $D_1$ , sendo  $\dim C_1 = \dim D_1 = \dim C_2/2$ .

Dessa forma, para que se possa ter um próximo nível de decomposição, basta que a matriz da transformação dessa vez atue apenas na parte referente a  $C_1$ , deixando  $D_1$  inalterado. Como a transformação no próximo nível continua a calcular médias e detalhes, a estrutura matricial para o novo bloco a ser construído é a mesma.

Assim, a nova transformação  $\hat{Q}_1 : V \rightarrow V$  possui um bloco formado pela identidade  $I$  encarregado de preservar  $D_1$  e um bloco  $Q_1$  para atuar sobre a outra metade dos dados,  $C_1$ . A formulação matricial resultante é dada por:

$$\hat{Q}_1(Q_2(C_2)) = \left[ \begin{array}{cc|cc} 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[ \begin{array}{c} C_{1,0} \\ C_{1,1} \\ \hline D_{1,0} \\ D_{1,1} \end{array} \right] = \left[ \begin{array}{c} \frac{C_{1,0}+C_{1,1}}{2} \\ \frac{C_{1,0}-C_{1,1}}{2} \\ \hline D_{1,0} \\ D_{1,1} \end{array} \right] = \left[ \begin{array}{c} \frac{C_{0,0}}{D_{0,0}} \\ \hline D_{1,0} \\ D_{1,1} \end{array} \right] \quad (1.9)$$

Como no Algoritmo 1, quando são aplicados  $J$  níveis de decomposição da TWD de Haar 1D, para cada nível  $j$  variando de  $J$  a 1, tem-se a transformação linear  $\hat{Q}_j : V \rightarrow V$ , cuja matriz associada é dada por:

$$[\hat{Q}_k] = \left[ \begin{array}{c|c} Q_j & 0 \\ \hline 0 & I_k(j, J) \end{array} \right] = \left[ \begin{array}{c|cccc} Q_j & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad (1.10)$$

A matriz  $[Q_j]$  de dimensão  $2^j$  é formada por blocos  $S_j$  e  $T_j$ , já a matriz identidade  $I_k(j, J)$  tem dimensão  $k(j, J) = 2^J - 2^j$ , responsável por preservar os detalhes de cada nível inalterados.

Desta forma, pode-se então representar matricialmente toda a TWD de Haar para  $J$  níveis como sendo a matriz  $\mathbf{W}$  associada à composta de todas as transformações para cada um dos níveis.

$$\mathbf{W} = \hat{Q}_1 \dots \hat{Q}_{J-2} \hat{Q}_{J-1} Q_J.$$

Caso o último nível de decomposição da transformada não seja o nível zero, que é o mais grosseiro possível com apenas um valor para média e um para detalhe, mas seja algum nível intermediário  $J_0$ , então denotamos  $L = J - J_0$  e:

$$\mathbf{W}^L = \hat{Q}_{J_0} \dots \hat{Q}_{J-2} \hat{Q}_{J-1} Q_J.$$

### Transformação de Haar Inversa - Forma Matricial para 1 e $j$ níveis

Nesta seção apresenta-se a formulação matricial para a transformada inversa referente ao exemplo associado à matriz  $Q_2$ . Os demais casos são todos análogos e portanto apresentados de forma sucinta.

Define-se  $Q_2^{-1} : V \rightarrow V$ , a transformação linear inversa de  $Q_2$ , cuja formulação matricial é dada por:

$$Q_2^{-1}(C_1) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} C_{1,0} \\ C_{1,1} \\ D_{1,0} \\ D_{1,1} \end{bmatrix} = \begin{bmatrix} C_{2,0} \\ C_{2,1} \\ C_{2,2} \\ C_{2,3} \end{bmatrix} = [C_2] \quad (1.11)$$

Desta forma, seguindo a mesma heurística de obtenção para a forma matricial envolvendo  $J$  níveis da TWD de Haar, pode-se portanto obter a inversa correspondente:

$$\mathbf{W}^{-1} = Q_J^{-1} \hat{Q}_{J-1}^{-1} \dots \hat{Q}_2^{-1} \hat{Q}_1^{-1}.$$

Sendo que, para  $k = 1, 2, \dots, J - 1$  as matrizes

$$\left[\hat{Q}_k^{-1}\right] = \left[\begin{array}{c|c} Q_j^{-1} & 0 \\ \hline 0 & I_k(j, J) \end{array}\right] \quad (1.12)$$

são construídas de forma análoga ao que foi descrito anteriormente.

### 1.1.3 TWD de Haar 1D Ortonormal

Uma propriedade interessante que se pode observar nas matrizes  $Q_j$ ,  $j = 1, 2, \dots, J$  é que suas colunas são formadas por vetores ortogonais, ou seja, o produto interno entre quaisquer duas de suas colunas é sempre zero. Tomando como exemplo a matriz  $Q_2$ , definida pela Equação 1.6, temos como vetores coluna  $\vec{u}_1 = (1/2, 0, 1/2, 0)$ ,  $\vec{u}_2 = (1/2, 0, -1/2, 0)$ ,  $\vec{u}_3 = (0, 1/2, 0, 1/2)$  e  $\vec{u}_4 = (0, 1/2, 0, -1/2)$ , e  $u_i \cdot u_k = 0$  para quaisquer  $i$  e  $k$ . No entanto, todas as colunas têm tamanho  $\|u_i\| = 1/\sqrt{2}$ , ou seja, os vetores colunas não são normais.

Para que as  $k = 1, 2, \dots, 4$  colunas de  $Q_2$  sejam vetores ortonormais, basta considerar os versores de cada um dos vetores coluna,  $\vec{u}_k/||\vec{u}_k||$ . Assim, a matriz resultante seria uma matriz ortogonal, o que é uma propriedade muito importante, pois a inversa de uma matriz ortogonal é simplesmente sua transposta.

Dessa forma tem-se a versão ortonormal da TWD de Haar para  $j = 1, 2, \dots, J$ , cuja matriz ortogonal é dada por

$$\mathbf{Q}_j = \left[ \begin{array}{cccccc} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ \hline 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{array} \right] \quad (1.13)$$

Assim para obter-se cada uma das transformações, consideram-se as matrizes ortonormais em cada nível, para  $j$  variando do nível mais fino  $J$  até o mais grosseiro possível, dado por 1. Como no caso geral não ortonormal, as matrizes  $I_{k(j,J)}$  são identidades de dimensão  $2^J - 2^j$ . Agora, o bloco  $\hat{\mathbf{Q}}_k$  é uma matriz ortonormal como a definida pela Equação 1.13.

$$[\hat{\mathbf{Q}}_k] = \left[ \begin{array}{c|c} \mathbf{Q}_j & 0 \\ \hline 0 & I_{k(j,J)} \end{array} \right] \quad (1.14)$$

Finalmente, a sequência de transformações da TWD de Haar em  $J$  níveis é:

$$\mathbb{W} = \hat{\mathbf{Q}}_1 \dots \hat{\mathbf{Q}}_{J-2} \hat{\mathbf{Q}}_{J-1} \mathbf{Q}_J.$$

E portanto a sua transformada inversa é, na verdade, a transposta de  $\mathbb{W}$ :

$$\mathbb{W}^{-1} = \mathbf{Q}_J^T \hat{\mathbf{Q}}_{J-1}^T \dots \hat{\mathbf{Q}}_2^T \hat{\mathbf{Q}}_1^T.$$

Com esta apresentação inicial, encerra-se a parte referente aos algoritmos e às formulações matriciais que definem a TWD de Haar 1D. Na próxima seção serão vistas propriedades mais específicas das funções de Haar e alguns dos conceitos fundamentais que caracterizam as funções *wavelets* ortonormais de Daubechies.

### 1.1.4 Análise Multiresolução

Na seção anterior, a partir de um vetor inicial de tamanho  $2^J$ , foram apresentados os algoritmos Cascata para as transformadas de Haar direta e inversa, que basicamente são responsáveis pelo cálculo de médias e diferenças entre dados. Além disso, foram construídas as matrizes associadas à TWD de Haar com vários níveis de decomposição (níveis de resolução).

Há ainda aspectos funcionais envolvendo as funções *wavelets* e suas transformadas, como por exemplo continuidade, suavidade, convergência e outras propriedades mais sofisticadas dependendo da família de funções *wavelet* escolhida. Ou seja, o espaço vetorial em questão deixa de ser considerado apenas como um espaço de dimensão finita,  $V = \mathbb{R}^n$ , e passa a ser um espaço de funções, que em muitos casos pode ser  $V = L^2(\mathbb{R})$ , conhecido como o espaço das funções de quadrado integrável ou espaço de Lebesgue. Para aplicações de domínio finito, outra

possibilidade é  $L^2([0, 1])$ , espaço de todas as funções integráveis em  $[0, 1]$ , cujo quadrado também é integrável em  $[0, 1]$  [33].

Abordar estas questões funcionais com rigor está longe do escopo deste texto inicial e introdutório, uma vez que a disciplina de Análise Funcional seria um dos pré-requisitos necessários para isto. No entanto a definição de análise multiresolução, introduzida por Stephane Mallat e Yves Meyer (1988/1989) será apresentada, uma vez que ela pode ser interpretada como um elo de ligação entre o que se entende como representação discreta e o "mundo" contínuo correspondente. Desta forma, uma outra maneira de se apresentar e construir uma TWD seria começar através do design da estrutura funcional que se deseja para o espaço de funções associado, ou seja, a análise multiresolução correspondente.

**Definição:**

Uma análise multiresolução de um espaço vetorial  $V = L^2(\mathbb{R})$  (ou  $L^2([0, 1])$ ) é uma sequência de subespaços fechados  $V_j \in V$ ,  $j \in \mathbb{Z}$ , com as seguintes propriedades:

1.  $\{\mathbf{0}\} \cdots \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_n \subset V_{(n+1)} \subset \cdots \subset V;$
2.  $\bigcap_{j \in \mathbb{Z}} V_j = \{\mathbf{0}\} \quad \text{e} \quad \overline{\bigcup_{j \in \mathbb{Z}} V_j} = V;$
3.  $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1};$
4.  $f(x) \in V_0 \Leftrightarrow f(x+k) \in V_0, \forall k \in \mathbb{Z};$
5.  $\exists \phi(x) \in V_0$  com integral não nula, denominada função escala, tal que  $\{\phi(x-k), k \in \mathbb{Z}\}$  é uma base ortonormal de  $V_0$ .

Para se poder avaliar a força (ou a extensão) da definição acima, são feitas algumas observações sobre cada um dos cinco itens apresentados com o objetivo de esclarecer as relações entre os subespaços encadeados que fatoram o espaço vetorial  $V = L^2(\mathbb{R})$  (ou  $L^2([0, 1])$ ) em diferentes níveis de resolução [23].

- As propriedades (1) e (2) garantem que a sequência de subespaços encaixados  $(V_j)_{j \in \mathbb{Z}}$  não é vazia, pois a função nula está contida em cada subespaço da sequência. Apesar de haver redundância entre dois quaisquer níveis consecutivos, pois  $V_j \subset V_{j+1}$ , a sequência  $(V_j)_{j \in \mathbb{Z}}$  completa tem como única intersecção a função nula, indicando que a redundância existente, na verdade, não é tão expressiva. Além disso, o fecho de sua união recobre o espaço todo,  $V$ . Isso quer dizer que projeções de uma função  $f(x) \in V$  em  $V_j$  são aproximações de  $f$ , que convergem para  $f$  ao  $j \rightarrow \infty$ ; Outra forma de interpretar (2) é afirmar que união de todos os subespaços é densa em  $V$ .
- A propriedade (3) indica que a relação de inclusão entre os subespaços encadeados  $V_j$  é de fato uma relação de auto-similaridade toda vez que uma função  $f(x)$  for reescalada por um fator de 2.

- A propriedade (4) está relacionada à auto-similaridade dos subespaços  $V_j$  com relação à variável  $x$ , que pode ser interpretada como o tempo nas aplicações envolvendo sinais. Com isso, os espaços  $V_j$  são invariantes com relação às translações (*shifts*);
- A propriedade (5) indica que as translações inteiras da função escala formam uma base ortonormal para o espaço  $V_0$ . Em muitos casos, exige-se ainda que a função escala seja uma função contínua por partes e não identicamente nula apenas em um intervalo fechado e limitado da reta (tenha suporte compacto), como é o caso das *wavelets* de Haar e de Daubechies. Na verdade, a propriedade (5) pode ser generalizada de tal modo que a base formada seja uma base de Riesz para  $V = L^2(\mathbb{R})$  (ou  $L^2([0, 1])$ ).

Assim através da estrutura de análise multiresolução, encontra-se a conexão entre a abordagem discreta e a contínua, pois informações representadas de forma discreta através de elementos de  $V_j$  tendem a informações contínuas, conforme a escala (ou nível de resolução)  $j$  aumenta.

Tendo isso em mente, de agora em diante, apresenta-se a análise multiresolução associada à função de Haar e, portanto, à TWD de Haar 1D, acrescentando-se informações complementares ao que foi exposto até o momento.

### 1.1.5 Base de Haar

Assume-se o espaço vetorial  $V = L^2(\mathbb{R})$ . Considera-se então a função constante por partes  $\phi(x) \in V$ , também conhecida como função de Haar, definida por:

$$\phi(x) := \begin{cases} 1, & \text{se } 0 \leq x < 1 \\ 0, & \text{caso contrário} \end{cases} \quad (1.15)$$

$\phi(x)$  é escolhida como a função escala para gerar a análise multiresolução de  $V$ , uma vez que sua integral é não nula e ainda sua norma é igual a 1:

$$\int_{-\infty}^{\infty} \phi(x) dx = 1 \text{ e } \|\phi(x)\|_2 = \left( \int_{-\infty}^{\infty} |\phi(x)|^2 dx \right)^{1/2} = 1. \quad (1.16)$$

Assim,  $\{\phi(x - k), k \in \mathbb{Z}\}$  é uma base para o subespaço  $V_0 \subset V$ , formada por funções contantes por partes em intervalos  $[k, k + 1)$  de tamanho 1. No caso particular de  $V = L^2([0, 1])$ ,  $V_0$  será o espaço gerado apenas por  $\phi(x)$ .

Através das relações de auto-similaridade em relação à translação e escala (Propriedades (3) e (4)), obtém-se ainda que  $\{\phi(2x - k), k \in \mathbb{Z}\}$  é uma base para o subespaço  $V_1$  e, consequentemente, que  $\{\phi(2^j x - k), k \in \mathbb{Z}\}$  é uma base para o subespaço  $V_j \in V, \forall j \in \mathbb{Z}$ . Na Figura 1.2, são ilustradas  $\phi(x) \in V_0$  e duas representantes em  $V_1$ , obtidas pela contração de  $\phi(x)$  por um fator de escala igual a 2 e por translação. Como para cada nível  $j$

$$\|\phi(2^j x - k)\|_2 = \left( \int_{-\infty}^{\infty} |\phi(2^j x - k)|^2 dx \right)^{1/2} = \sqrt{(1/2^j)}, \quad (1.17)$$

é necessário normalizar cada uma das funções para obter-se uma base ortonormal. Assim,  $\{2^{j/2}\phi(2x - k), k \in \mathbb{Z}\}$  é a base ortonormal procurada de cada um dos subespaços  $V_j$ , construída a partir da função escala de Haar. Nas aplicações apresentadas,  $V_j$ ,  $j = 0, 1, 2, \dots, J$  são os subespaços nos quais as informações médias dos sinais a serem analisados serão representados.

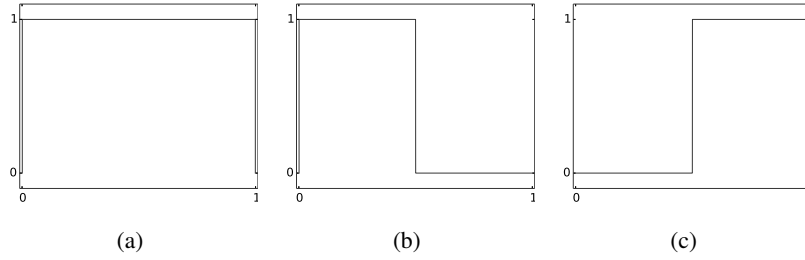


Figura 1.2: Funções  $\phi(x)$  em (a);  $\phi(2x)$  em (b) e  $\phi(2x - 1)$  em (c).

Como  $V_j \subset V_{j+1}$ , ao mesmo tempo que  $V_j \neq V_{j+1}$ , define-se  $W_j \subset V_{j+1}$  como sendo o complemento ortogonal de  $V_j$  em  $V_{j+1}$ , ou seja,  $V_{j+1} = V_j \oplus W_j$ . Considerando agora os subespaços  $V_J$  e  $V_{J_0}$  com  $J > J_0$ , e os complementos ortogonais entre subespaços de escalas consecutivas, tem-se

$$V_J = (\dots((V_{J_0} \oplus W_{J_0}) \oplus W_{J_0+1}) \dots \oplus W_{J-1}). \quad (1.18)$$

Assim, qualquer função pertencente a  $V_J$  pode ser expressa como combinação linear de funções em  $V_{J_0}$  e  $W_j$ ;  $j = J_0; J_0 + 1; \dots; J$ , podendo desta forma ser analisada separadamente em diferentes escalas. A Análise Multiresolução recebeu seu nome por permitir esta separação de escalas.

Continuando a decomposição envolvendo os complementos ortogonais em cada nível, quando  $J_0 \rightarrow -\infty$  e  $J \rightarrow \infty$ , obtém-se

$$\left( \bigoplus_{j=-\infty}^{\infty} W_j \right) = L^2(\mathbb{R}). \quad (1.19)$$

Como consequência, todos os subespaços  $W_j$  são dois a dois ortogonais. Além disso, existe  $\psi(x) \in W_0$  tal que  $\{\psi(x - k), k \in \mathbb{Z}\}$  é base de  $W_0$  e suas dilatações e translações formam base dos demais espaços  $W_j$ ,  $\{\psi(2^j x - k), j, k \in \mathbb{Z}\}$ , como demonstrado por Daubechies [9]. A função  $\psi(x) \in W_j$ , denominada de função *wavelet* de Haar, é definida por:

$$\psi(x) := \begin{cases} 1, & \text{se } 0 \leq x < 1/2 \\ -1, & \text{se } 1/2 \leq x < 1 \\ 0, & \text{caso contrário.} \end{cases} \quad (1.20)$$

Na Figura 1.3 são apresentadas a função  $\psi(x) \in W_0$  e duas representantes em  $W_1$ ,  $\psi(2x)$  e  $\psi(2x - 1)$ . A função  $\psi(x)$  possui integral nula e sua norma  $\|\psi(x)\|_2 =$

$\sqrt{1/2^j}$ . Analogamente ao que foi feito para as funções escala, pode-se obter uma base ortonormal para cada um dos subespaços  $W_j, j \in \mathbb{Z}$ , através de translações e dilatações de  $\psi(x)$ , dada por  $\{2^{j/2}\psi(2^j x - k), j \in \mathbb{Z}\}$ .

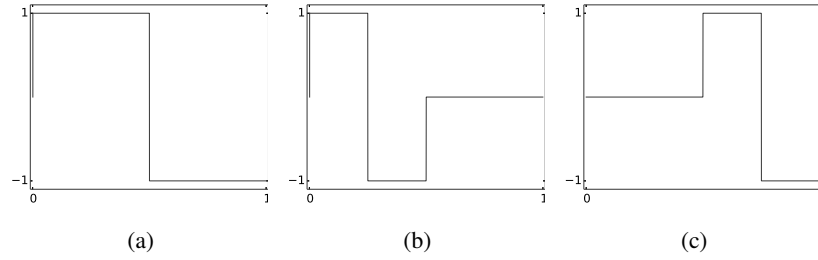


Figura 1.3: Função  $\psi(x)$  em (a);  $\psi(2x)$  em (b) e  $\psi(2x - 1)$  em (c).

A notação compacta para representar as funções escala e *wavelet*, com  $j$  indicando a escala e  $i$  as translações referentes à escala considerada, é dada por:

$$\phi_{j,i}(x) := 2^{j/2} \phi(2^j x - i) \quad (1.21)$$

e

$$\psi_{j,i}(x) := 2^{j/2} \psi(2^j x - i). \quad (1.22)$$

Pela construção apresentada,  $\phi(x) = \phi_{0,0}(x) \in V_0$  e  $\psi(x) = \psi_{0,0}(x) \in W_0$  são ortonormais. Na verdade, além de ser preservada ao longo das escalas, a ortogonalidade entre essas funções ainda ocorre em relação ao parâmetro de translação, como apresentado a seguir:

$$\langle \phi_{j,i}(x), \phi_{j,k}(x) \rangle = \int_{-\infty}^{\infty} \phi_{j,i}(x) \phi_{j,k}(x) dx = \delta_{i,k}, \quad (1.23)$$

$$\langle \psi_{j,i}(x), \psi_{l,k}(x) \rangle = \int_{-\infty}^{\infty} \psi_{j,i}(x) \psi_{l,k}(x) dx = \delta_{j,l} \delta_{i,k}, \quad (1.24)$$

$$\langle \phi_{j,i}(x), \psi_{l,k}(x) \rangle = \int_{-\infty}^{\infty} \phi_{j,i}(x) \psi_{l,k}(x) dx = 0, \text{ para } l \geq j, \quad (1.25)$$

sendo  $j, i, l, k \in \mathbb{Z}$  e  $\delta_{i,k}$  a função Delta de Kronecker que vale 1, quando  $i = k$  e vale zero, caso contrário.

### 1.1.6 Relação de Escala e Relação Wavelet

O objetivo desta Seção é começar a responder a pergunta: "Mas, afinal, o que a parte vetorial vista no exemplo inicial tem a ver com a base de Haar recém apresentada?"

Para responder esta pergunta, inicialmente apresenta-se uma propriedade fundamental das *wavelets* de Daubechies e em particular da *wavelet* de Haar. Esta

propriedade, chamada Relação de Escala, ou Equação de Refinamento é uma consequência imediata da estrutura de análise multiresolução, propriedades (1) e (3):

Como  $V_0 \subset V_1$  e  $W_0 \subset V_1$ , qualquer função de  $V_0$  e de  $W_0$  podem ser escritas com respeito à base de  $V_1$ , a saber, a própria função escala  $\phi$  e a *wavelet*  $\psi$  podem ser escritas como combinação linear em relação à base ortonormal de  $V_1$ :

$$\phi(x) = \sum_{k=-\infty}^{+\infty} h_k \phi_{1,k}(x) = \sqrt{2} \sum_{k=-\infty}^{+\infty} h_k \phi(2x - k), \quad (1.26)$$

$$\psi(x) = \sum_{k=-\infty}^{+\infty} g_k \phi_{1,k}(x) = \sqrt{2} \sum_{k=-\infty}^{+\infty} g_k \phi(2x - k), \quad (1.27)$$

sendo os valores  $h_k$  e  $g_k$ ,  $k \in \mathbb{Z}$  as constantes destas combinações lineares, também denominados filtros associados às funções escala e *wavelet*.

Como  $V_1$  possui base ortonormal,

$$h_k = proj_{\phi_{1,k}(x)} \phi(x) = \langle \phi(x), \phi_{1,0}(x) \rangle = \int_{-\infty}^{\infty} \phi(x) \phi_{1,k}(x) dx, \quad (1.28)$$

$$g_k = proj_{\phi_{1,k}(x)} \psi(x) = \langle \psi(x), \phi_{1,0}(x) \rangle = \int_{-\infty}^{\infty} \psi(x) \phi_{1,k}(x) dx. \quad (1.29)$$

No caso geral da família de Daubechies, a Relação 1.26 terá apenas um número finito de filtros não nulos, isso porque as funções escala e *wavelet* são identicamente nulas, com exceção apenas em um intervalo limitado e fechado da reta. Assim, diz-se que as *wavelets* da família de Daubechies são funções de suporte compacto. Daubechies em seu trabalho seminal [9] ainda demonstrou que há uma relação 1:1 entre os filtros e as funções escala e *wavelet* de suporte compacto, dependendo de sua suavidade (diferenciabilidade) e do número de momentos nulos que a base possui, ou seja, o grau máximo dos polinômios que podem ser exatamente representados apenas por funções escala, tendo nulos seus coeficientes na expansão em relação às *wavelets*. Na verdade, o número de coeficientes não nulos  $D$  depende do número de momentos nulos  $N$ ,  $D = 2N$  e a Relação de escala 1.26 fica então simplificada:

$$\phi(x) = \sum_{k=0}^{D-1} h_k \phi_{1,k}(x) = \sqrt{2} \sum_{k=0}^{D-1} h_k \phi(2x - k), \quad (1.30)$$

$$\psi(x) = \sum_{k=0}^{D-1} g_k \phi_{1,k}(x) = \sqrt{2} \sum_{k=0}^{D-1} g_k \phi(2x - k), \quad (1.31)$$

No caso particular da função de Haar,  $D=2$  e tem-se

$$\phi(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) + \frac{1}{\sqrt{2}}\phi_{1,1}(x) = \phi(2x) + \phi(2x-1) \quad (1.32)$$

$$\psi(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) - \frac{1}{\sqrt{2}}\phi_{1,1}(x) = \phi(2x) - \phi(2x-1), \quad (1.33)$$



sendo os filtros  $h = (h_0, h_1) = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$  e  $g = (g_0, g_1) = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$  exatamente iguais aos coeficientes da matriz  $Q$ , dada pela Equação 1.13, que representa a TWD de Haar normalizada. Além disso, há ainda uma relação entre os filtros  $h$  e  $g$ , a saber:  $g_k = (-1)^k h_{D-1-k}$  para  $k = 0, 1, \dots, D-1$  [9], que faz com que o número de dados iniciais para o cálculo da transformada *wavelet* seja ainda mais reduzido. Isto porque, a partir da função escala escolhida, as demais informações sobre as funções *wavelets* associadas podem ser obtidas como consequência. A Tabela 1.1 destaca os filtros de escala e *wavelet* dos casos não normalizado e normalizado.

Tabela 1.1: Coeficientes de filtro não normalizados e normalizados para DWT de Haar

Filtros não normalizados			Filtros normalizados	
$k$	$h_k$	$g_k$	$h_k$	$g_k$
0	1/2	1/2	$1/\sqrt{2}$	$1/\sqrt{2}$
1	1/2	-1/2	$1/\sqrt{2}$	$-1/\sqrt{2}$

Apesar de as funções  $\phi(x)$  e  $\psi(x)$  de Haar serem facilmente definidas e possuírem formulação analítica (Equações 1.15 e 1.20 ou 1.21 e 1.22), geralmente não há fórmulas explícitas para outras *wavelets*. Portanto, a maioria dos algoritmos para decomposição *wavelet* é formulado em termos de bancos de filtros de coeficientes [23], sendo uma consequência direta das relações de escala e *wavelet*.

### 1.1.7 Expansão de funções em $V_j$

Os subespaços vetoriais  $V_j \subset L^2(\mathbb{R})$ ,  $j \in \mathbb{Z}$ , são espaços de funções e uma função  $f(x) \in V_j$  pode ser escrita como combinação linear de diferentes bases de  $V_j$ . Em particular, pode-se considerar a base ortonormal de Haar formada pelas funções escala  $\phi_{j,k}(x)$  no nível  $j$ . A expansão de  $f(x)$  nesta base é dada por:

$$f(x) = \sum_{k=-\infty}^{\infty} C_{j,k} \phi_{j,k}(x), \quad (1.34)$$

sendo os coeficientes da combinação linear definidos pela projeção de  $f(x)$  na direção de cada um dos vetores da base:

$$C_{j,k} = \frac{\langle f(x), \phi_{j,k}(x) \rangle}{\langle \phi_{j,k}(x), \phi_{j,k}(x) \rangle} = \langle f(x), \phi_{j,k}(x) \rangle = \int_{-\infty}^{\infty} f(x) \phi_{j,k}(x) dx. \quad (1.35)$$

Os coeficientes  $C_{j,k}$  são as coordenadas de  $f(x)$  na base de  $V_j$  e a TWD de Haar é um procedimento para calculá-las a partir de dados discretos.

Caso  $V_j \subset L^2(I)$ , com  $I = [0, 1)$  ou algum outro intervalo limitado da reta, então os limites de variação do somatório na Expressão 1.34 e os limites de integração na Expressão 1.35 são todos finitos. Os limites de integração coincidem com os limites do intervalo considerado e o número de pontos do somatório depende, além do tamanho do intervalo, também da escala (nível de resolução), responsável pelo espaçamento entre os pontos obtidos na discretização da função. Neste texto, assume-se que para cada  $j \in \mathbb{Z}$ , o espaçamento seja  $1/2^j$  e portanto em  $I = [0, 1)$  tem-se  $N_j = 2^j$  pontos, iniciando-se a contagem sempre em zero. Para as aplicações, quando for necessário considerar um intervalo maior como domínio, então os ajustes necessários serão apontados.

Dessa forma, a expansão de  $f(x) \in V_j \subset L^2(I)$  é dada a seguir:

$$f(x) = \sum_{k=0}^{N_j-1} C_{j,k} \phi_{j,k}(x), \text{ com } N_j = 2^j \text{ e} \quad (1.36)$$

$$C_{j,k} = \langle f(x), \phi_{j,k}(x) \rangle = \int_0^1 f(x) \phi_{j,k}(x) dx. \quad (1.37)$$

Como a quantidade de coeficientes  $C_{j,k}$  é finita, eles podem ser armazenados em um vetor  $C_j$ . E... *vois lá !* Têm-se os vetores que apareceram inicialmente na Seção 1.1 para apresentar o exemplo!

Ou seja, a Expressão 1.36 dá o enfoque funcional que faltava para se entender as representações do vetor  $C_2 = [2 \ 0 \ 1 \ 5]$  apresentadas nas Figuras 1.1(b), 1.1(c) e 1.1(d). O que a formulação funcional deixa claro agora é a relação biunívoca (1:1) entre  $C_2 = [2 \ 0 \ 1 \ 5]$  e a combinação linear das funções da base de Haar,  $C_2 \leftrightarrow f(x)$ , como ilustra a Figura 1.4.

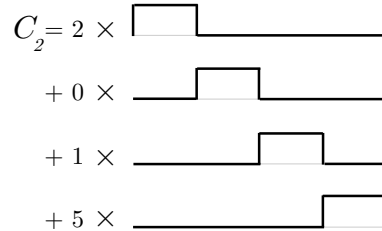


Figura 1.4:  $C_2 \leftrightarrow f(x) = 2\phi_{2,0}(x) + 0\phi_{2,1}(x) + 1\phi_{2,2}(x) + 5\phi_{2,3}(x)$ .

Lembrando que  $V_j = V_{j-1} \oplus W_{j-1}$ , uma outra possibilidade de se expandir uma função  $f(x) \in V_j \subset L^2(I)$  é considerar agora a base de  $V_j$  como sendo a união das bases de  $V_{j-1}$  e  $W_{j-1}$  (o que é possível, pois  $V_j$  é soma direta destes dois subespaços).  $N_{j-1} = N_j/2 = 2^{j-1}$  é a dimensão desses dois subespaços,

$$f(x) = \sum_{k=0}^{N_{j-1}-1} C_{j-1,k} \phi_{j-1,k}(x) + \sum_{k=0}^{N_{j-1}-1} D_{j-1,k} \psi_{j-1,k}(x), \text{ com} \quad (1.38)$$

$$C_{j-1,k} = \langle f(x), \phi_{j-1,k}(x) \rangle \text{ e } D_{j-1,k} = \langle f(x), \psi_{j-1,k}(x) \rangle. \quad (1.39)$$

A Expressão 1.38 é a versão funcional do Algoritmo 1 para a versão decimada da TWD de Haar em um nível de decomposição.

Considerando agora que o nível mais fino de resolução para representação de  $f$  seja  $J$ ,  $f(x) \in V_J$ , e assumindo que sejam feitos todos os  $J$  níveis de transformação possíveis, tem-se a expansão da  $f(x)$  em multinível dada por:

$$f(x) = \sum_{k=0}^{N_0-1} C_{0,k} \phi_{0,k}(x) + \sum_{j=0}^{J-1} \sum_{k=0}^{N_j-1} D_{j,k} \psi_{j,k}(x), \quad N_j = 2^j. \quad (1.40)$$

A expansão de  $f(x) \in V_J$  dada pela Equação 1.40 tem como coeficientes os elementos dos vetores  $C_0, D_0, D_1, D_2, \dots, D_{J-1}$ . Desta forma, agora fica evidente a relação 1:1 entre  $f(x)$  e sua representação multiresolução  $C_{MR} = (C_0 D_0 D_1 D_2 \dots D_{J-1})$ , obtida como resultado final do Algoritmo 1 para a TWD de Haar com todos os vetores obtidos nos níveis de decomposição ordenados em blocos. Caso a fatoração pela transformada não seja completa, isto é, o nível mais grosseiro escolhido seja  $J_0$ , com  $J_0 \neq 0$ , tem-se uma representação multiresolução em  $J - J_0$  níveis:  $f(x) \leftrightarrow C_{MR} = (C_{J_0} D_{J_0} D_1 D_2 \dots D_{J-1})$ , com  $C_{J_0}$  contendo  $2^{J_0}$  elementos. Ainda denota-se por  $L = J - J_0$  a profundidade da expansão *wavelet* e naturalmente, da representação multiresolução associada.

A Figura 1.5(a) ilustra a representação em um nível de resolução e a Figura 1.5(b) representa a combinação linear envolvendo todos os níveis de decomposição, até atingir  $V_0$  que representa o nível mais grosseiro contendo as funções constantes em todo intervalo  $I$ .

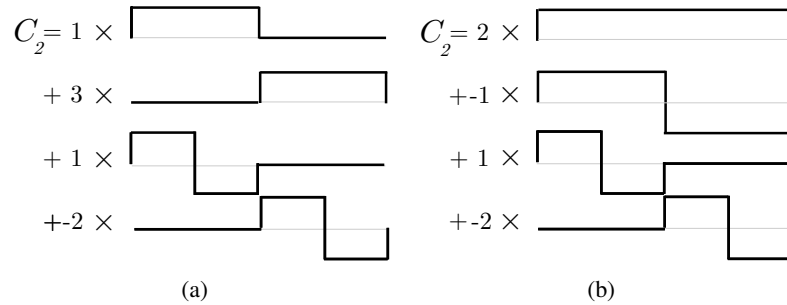


Figura 1.5: (a)  $f(x) \in V_1 \oplus W_1$  e  $C_2 \leftrightarrow 1\phi_{1,0}(x) + 3\phi_{1,1}(x) + 1\psi_{1,0}(x) + (-2)\psi_{1,1}(x)$ ; (b)  $f(x) \in V_0 \oplus W_0 \oplus W_1$  e  $C_2 \leftrightarrow 2\phi_{0,0}(x) + (-1)\psi_{0,0}(x) + 1\psi_{1,0}(x) + (-2)\psi_{1,1}(x)$ .

Na prática é muito comum e útil poder aplicar a TWD de Haar para decompor um sinal em uma quantidade desejada de níveis, não necessariamente chegando até o nível zero. Isso porque muitas vezes é essencial que o sinal mais grosseiro continue mantendo propriedades significativas do sinal original discretizado em um nível mais refinado.

### Norma de $f(x)$

Uma consequência importante de se ter uma base ortonormal em  $V_J$  é a possibilidade de se calcular a norma (tamanho) de um vetor qualquer de  $V_J$  usando-se diretamente suas coordenadas nesta base. Assim, o cálculo da norma de  $f(x)$  pode ser feito considerando, por exemplo, sua expansão apenas na base de funções escalas no nível  $J$  ou sua expansão envolvendo multiníveis:

$$\|f\|_2^2 = \langle f(x), f(x) \rangle = \sum_{k=0}^{N_J-1} |C_{J,k}|^2 = \sum_{k=0}^{N_{J_0}-1} |C_{J_0,k}|^2 + \sum_{j=0}^{J-1} \sum_{k=0}^{N_j-1} |D_{j,k}|^2. \quad (1.41)$$

Através dessa flexibilidade para o cálculo da norma de um vetor de  $V_J$  será possível avaliar o erro cometido nas aproximações de funções que não estão em  $V_J$ , mas são aproximadas por funções deste subespaço. Nas aplicações isso será muito útil para se poder avaliar a qualidade dos resultados computacionais obtidos, especialmente por ser necessário lidar com dados e expansões truncados em todas as etapas das análises.

### 1.1.8 Projeções em $V_j$

Quando for dada uma função  $f(x) \in L^2(\mathbb{R})$  que não seja originalmente constante por partes, essa função não poderá ser representada exatamente por apenas um único subespaço  $V_j$  para um certo nível  $j$  escolhido, mas será possível obter sua projeção no espaço  $V_j$ , denotada por  $(P_{V_j}f)(x)$ , assim como será possível obter sua projeção em  $W_j$ ,  $(P_{W_j}f)(x)$ :

$$(P_{V_j}f)(x) = \sum_{k=0}^{N_j-1} C_{j,k} \phi_{j,k}(x), \text{ e } (P_{W_j}f)(x) = \sum_{k=0}^{N_j-1} D_{j,k} \psi_{j,k}(x), \quad (1.42)$$

cujos coeficientes  $C_{j,k}$  e  $D_{j,k}$  são calculados como anteriormente, através do produto interno entre  $f(x)$  e a respectiva função da base no nível  $j$  e na posição  $k$ .

Naturalmente, através da estrutura de análise multiresolução, continua válido que  $(P_{V_j}f)(x) = (P_{V_{j-1}}f)(x) + (P_{W_{j-1}}f)(x)$ .

O objetivo agora é então amarrar a última ponta que ainda está solta, que diz respeito a como calcular estes coeficientes  $C_{j,k}$  e  $D_{j,k}$  de forma eficiente, considerando os ingredientes funcionais apresentados nesta seção. Dessa forma, o que será obtido são novamente os algoritmos apresentados a partir do exemplo, caracterizando a relação biunívoca entre  $C_j$  e  $(C_{j-1} \ D_{j-1})$ , também conhecida como transformada rápida *wavelet* (*fast wavelet transform*).

A chave para a obtenção desta relação entre duas escalas consecutivas quaisquer,  $j$  e  $j-1$ , está na Relação de Escala 1.30 adaptada agora para estes níveis. Assim começando pela definição de uma função escala normalizada no nível  $j-1$ , tem-se:

$$\phi_{j-1,l}(x) = 2^{(j-1)/2} \phi(2^{j-1}x - l) = 2^{(j-1)/2} \phi(y), \text{ com } y = 2^{j-1}x - l,$$

$$\text{Por 1.30 para } \phi(y), \text{ tem-se } \phi_{j-1,l}(x) = 2^{(j-1)/2} \left[ 2^{1/2} \sum_{k=0}^{D-1} h_k \phi(2y - k) \right]$$

$$\text{Assim, } \phi_{j-1,l}(x) = 2^{j/2} \sum_{k=0}^{D-1} h_k \phi(2^j x - (2l + k)) = \sum_{k=0}^{D-1} h_k \phi_{j,2l+k}(x).$$

Lembrando apenas que  $D=2$ , no caso da TWD de Haar. De forma análoga, tem-se

$$\psi_{j-1,l}(x) = \sum_{k=0}^{D-1} g_k \phi_{j,2l+k}(x).$$

Agora, pela definição de  $C_{j-1,l}$

$$\begin{aligned} C_{j-1,l} &= \int_0^1 f(x) \phi_{j-1,l}(x) dx = \int_0^1 f(x) \left[ \sum_{k=0}^{D-1} h_k \phi_{j,2l+k}(x) \right] dx \\ C_{j-1,l} &= \sum_{k=0}^{D-1} h_k \int_0^1 f(x) \phi_{j,2l+k}(x) dx = \sum_{k=0}^{D-1} h_k C_{j,2l+k}(x). \end{aligned}$$

Analogamente, tem-se o mesmo tipo de relação para os coeficientes *wavelets*:

$$D_{j-1,l} = \sum_{k=0}^{D-1} g_k C_{j,2l+k}(x).$$

E no caso particular da TWD de Haar essas relações para os coeficientes de escala e *wavelets* se traduzem exatamente na formulação apresentada anteriormente em função dos filtros  $h_k$  e  $g_k$  (transformada rápida *wavelet* via algoritmo de Cascata) :

$$\begin{aligned} C_{j-1,l} &= \sum_{k=0}^1 h_k C_{j,2l+k}(x) = h_0 C_{j,2l}(x) + h_1 C_{j,2l+1}(x), \quad l = 0, \dots, 2^{j-1} - 1, \\ D_{j-1,l} &= \sum_{k=0}^1 g_k C_{j,2l+k}(x) = g_0 C_{j,2l}(x) + g_1 C_{j,2l+1}(x), \quad l = 0, \dots, 2^{j-1} - 1. \end{aligned}$$

Essa transformada é especialmente eficiente no cálculo dos coeficientes  $C_{j-1}$  e  $D_{j-1}$  especialmente por serem necessários apenas os coeficientes  $C_j$  e os filtros  $h$  da função escala, uma vez que  $g$  pode ser obtido a partir de  $h$ . Assim, a questão é como começar o processo, como obter os dados de  $C_j$ . Intuitivamente, faz sentido começar o processo, assumindo  $C_{j,k} = f(x_{j,k}) = f(2^j x - k)$ , desde que  $j$  seja suficientemente grande, ou seja, que o espaçamento  $1/2^j$  seja suficientemente pequeno. Para ver uma demonstração geral da justificativa desta escolha, ver [23].

## Capítulo 2

# Aplicações 1D

O objetivo principal deste capítulo é apresentar algumas aplicações para a TWD de Haar 1D, cuja formulação foi vista no capítulo anterior através dos algoritmos de Cascata, seus análogos em forma matricial e ainda através do enfoque funcional promovido pela estrutura de análise de multiresolução. Nas seções que seguem, alguns conceitos importantes são introduzidos ou reforçados, pois são ferramentas essenciais para a análise de dados (sejam eles funções, sinais, imagens...). Como este texto se propõe a utilizar basicamente tópicos de Álgebra Linear vistos em disciplinas de graduação, muitos dos conceitos abstratos são omitidos e apenas a parte intuitiva é explorada. No entanto, existem vários textos especializados, como por exemplo as referências [9], [22] e [10], que exploram de forma sistemática e rigorosa, tanto a parte teórica, quanto uma gama maior de aplicações.

### 2.1 Propriedade dos Momentos Nulos

No capítulo anterior, Seção 1.1.6, foi mencionada brevemente a importante propriedade dos  $N$  momentos nulos que as funções escala da família de Daubechies possuem, que é a habilidade de representar polinômios exatamente até grau  $(N - 1)$ . Mais precisamente, cada elemento da base de polinômios até grau  $N - 1$  é expandido como segue:

$$x^p = \sum_{k=-\infty}^{+\infty} M_k^p \phi(x - k), \quad x \in \mathbb{R}, \quad p = 0, 1, \dots, N - 1, \text{ com} \quad (2.1)$$

$$M_k^p = \langle x^p, \phi(x - k) \rangle, \quad k \in \mathbb{Z}, \quad p = 0, 1, \dots, N - 1. \quad (2.2)$$

$M_k^p$  denota o  $p$ -ésimo momento da função escala  $\phi(x - k)$  e pode ser também calculado de maneira rápida para qualquer uma das funções de Daubechies, como apresentado em [23]. No caso particular da função escala de Haar,  $M_k^p$  pode ser calculado diretamente pela sua definição.

A Equação 2.1 pode ser traduzida em uma condição envolvendo as funções *wavelets*. Considerando o produto interno de  $x^p$  com  $\psi(x)$  e a ortogonalidade entre as funções da base *wavelet*, tem-se para  $x \in \mathbb{R}$ ,  $p = 0, 1, \dots, N - 1$  que:

$$\int_{k=-\infty}^{+\infty} x^p \psi(x) dx = \sum_{k=-\infty}^{+\infty} M_k^p \int_{k=-\infty}^{+\infty} \psi(x) dx \phi(x-k) = 0. \quad (2.3)$$

Esta é exatamente a propriedade dos  $N$  momentos nulos (em relação às funções *wavelets*), dado que em relação às funções escala, os valores  $M_K^p$  não são nulos.

Intuitivamente esta propriedade significa que pode-se fazer uma economia considerável na representação de uma função  $f$  na base *wavelet*. Toda vez que a  $f$  for uma função polinomial ou polinomial por partes com grau até  $N - 1$ , serão necessários apenas metade dos coeficientes na sua expansão na base *wavelet* com  $N$  momentos. No caso particular da base de Haar, que possui apenas  $N = 1$  momento, toda vez que a função for constante ou constante por partes, essa economia na representação é obtida em todo o domínio, ou localmente, no caso de funções constantes por partes. Mesmo no caso de uma função ser constante apenas em uma região do seu domínio, os coeficientes *wavelets* da expansão nesta região serão nulos, havendo uma economia de representação apenas no local correspondente.

Na verdade, esse fato abre então uma nova linha de pensamento. Supondo agora que a função  $f$  não seja polinomial, mas seja, em algumas regiões de seu domínio, bem representada por polinômios ou próxima de um, isso implicaria que, apesar de não serem nulos, os coeficientes *wavelets* seriam pequenos. Bem, para justificar tudo isso são necessárias ferramentas mais técnicas. No entanto esse fato, que diz respeito ao comportamento quantitativo dos coeficientes *wavelets*, será explorado através de exemplos na seção de experimentos numéricos e apenas formulado sem demonstração logo a seguir.

### 2.1.1 Decaimento dos coeficientes *wavelet*

**Teorema** *Seja  $N$  o número de momentos nulos da função wavelet  $\psi_{j,k}$  e seja  $f \in C^N(\mathbb{R})$ , espaço das funções reais contínuas com derivadas contínuas até grau  $N$ . Então os coeficientes wavelets dados em 1.38 decaem em módulo ao longo dos níveis de decomposição da TWD como segue:*

$$|D_{j,k}| \leq \alpha_N 2^{-j(N+1/2)} \max_{\xi \in I_{j,k}} |f^{(N)}(\xi)|, \quad (2.4)$$

sendo  $\alpha_N$  uma constante independente de  $j$ ,  $k$  e de  $f$ ,  $D = 2N$  e  $I_{j,k} = \text{supp}\{\psi_{j,k}\} = [k/2^j, (k + D - 1)/2^j]$ .

No caso da base de Haar, o suporte das funções *wavelet*  $I_{j,k} = \text{supp}\{\psi_{j,k}\}$  são os intervalos  $I_{j,k} = [k/2^j, (k + 1)/2^j]$  de tamanho  $1/2^j$ . O módulo dos coeficientes *wavelet* decai dependendo do nível  $j$  e o máximo da derivada da  $f$  em cada intervalo  $I_{j,k}$ . Assim, quanto mais fino o nível de resolução ( $j \gg 1$ ), menor o módulo do coeficiente *wavelet*. A Figura 2.1 ilustra este comportamento através da decomposição em 7 níveis de uma função  $f$  definida por partes como sendo constante, linear, linear novamente apenas com sinal oposto e quadrática.

## 2.1. PROPRIEDADE DOS MOMENTOS NULOS

25

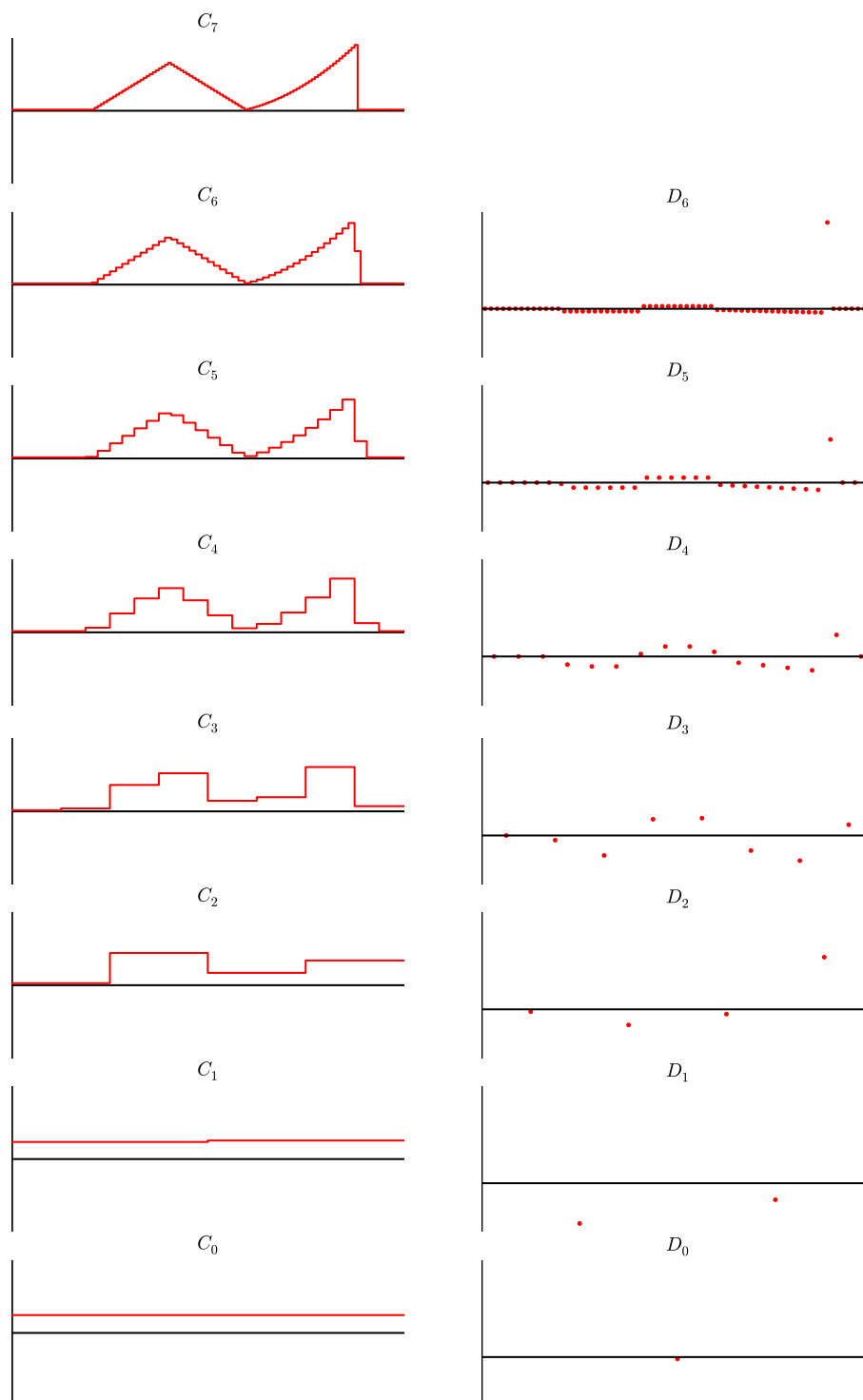


Figura 2.1: TWD de Haar de  $f$ . Variação do módulo dos detalhes por nível



## 2.2 Operações de truncamento

A primeira operação de truncamento foi mencionada no capítulo anterior e diz respeito à representação da função  $f \in L^2(\mathbb{R})$  (ou  $L^2([0, 1])$ ) em algum espaço  $V_J$  através da projeção ortogonal de  $f$  em  $V_J$ ,  $(P_{V_J}f)(x)$ . Neste processo o que de fato ocorre é o truncamento da expansão da função  $f$  para que sua representação seja feita em relação ao nível  $J$ . Com isso, define-se o erro de aproximação pontual nesta operação, para  $x$  arbitrariamente fixado, e supondo  $f \in \mathcal{C}^N$  por:

$$e_J(x) = f(x) - (P_{V_J}f)(x), \quad x \in \mathbb{R}. \quad (2.5)$$

Assumindo a expansão *wavelet* para  $f$ , do nível mais grosseiro  $J_0$  até o mais fino  $J$ , tem-se:

$$f(x) = \sum_{k=-\infty}^{+\infty} C_{J_0,k} \phi_{J_0,k}(x) + \sum_{j=J_0}^{+\infty} \sum_{k=-\infty}^{+\infty} D_{j,k} \psi_{j,k}(x) \quad (2.6)$$

De forma análoga, tem-se  $(P_{V_J}f)(x)$ , a projeção de  $f$  em  $V_J$

$$(P_{V_J}f)(x) = \sum_{k=-\infty}^{+\infty} C_{J_0,k} \phi_{J_0,k}(x) + \sum_{j=J_0}^{J-1} \sum_{k=-\infty}^{+\infty} D_{j,k} \psi_{j,k}(x) \quad (2.7)$$

Assim, o erro para cada  $x \in \mathbb{R}$  nesta operação de projeção é dado em termos das *wavelets* nas escalas  $j \geq J$ :

$$e_J(x) = f(x) - (P_{V_J}f)(x) = \sum_{j=J}^{+\infty} \sum_{k=-\infty}^{+\infty} D_{j,k} \psi_{j,k}(x). \quad (2.8)$$

Na tentativa de se achar uma quota superior para este erro, define-se para cada  $I_{j,k} = \text{supp}(\psi_{j,k}) = [k/2^j, (k+D-1)/2^j]$

$$C_\psi = \max_{x \in I_{j,k}} |\psi(2^j x - k)| = \max_{y \in [0, D-1]} |\psi(y)|. \quad (2.9)$$

Através do Teorema 2.1.1, estima-se que cada termo do somatório possa ser majorado por:

$$|D_{j,k} \psi_{j,k}(x)| \leq \alpha_N 2^{-j(N+1/2)} \max_{\xi \in I_{j,k}} |f^{(N)}(\xi)| 2^{j/2} C_\psi = \alpha_N 2^{-jN} \max_{\xi \in I_{j,k}} |f^{(N)}(\xi)| C_\psi.$$

Como os suportes das funções *wavelets* são os intervalos  $I_{j,k}$ , para cada valor de  $x$  escolhido, existem apenas  $D-1$  intervalos contendo este valor, e portanto o somatório em  $k$  possui apenas  $D-1$  termos não nulos. Assim, definimos  $\mu_j^N(x)$  como sendo

## 2.2. OPERAÇÕES DE TRUNCAMENTO

27

$$\mu_j^N(x) = \max_{x \in I_j(x)} |f^{(N)}(\xi)|, \quad \text{com } I_j(x) = \bigcup_{\{l: x \in I_{j,l}\}} I_{j,l}. \quad (2.10)$$

Com isso, o somatório em  $k$  pode ser majorado por:

$$\sum_{k=-\infty}^{+\infty} |D_{j,k} \psi_{j,k}(x)| \leq \alpha_N C_\psi 2^{-jN} (D-1) \mu_j^N(x). \quad (2.11)$$

Finalmente o somatório em  $j$  também pode ser majorado, pois o máximo da derivada  $n$ -ésima da  $f$  é procurado em intervalos cada vez menores, conforme a escala  $J$  cresce, gerando uma não crescente para esses valores de máximo,  $\mu_J^N(x) \geq \mu_{J+1}^N(x) \geq \mu_{J+2}^N(x) \geq \dots$ . Com isso,

$$\begin{aligned} |e_J(x)| &\leq \sum_{j=J}^{+\infty} \sum_{k=-\infty}^{+\infty} |D_{j,k} \psi_{j,k}(x)| \leq \alpha_N C_\psi (D-1) \mu_J^N(x) \sum_{j=J}^{+\infty} 2^{-jN} \\ &\quad \alpha_N C_\psi (D-1) \mu_J^N(x) \frac{2^{-JN}}{1 - 2^{-N}} = C 2^{-JN}. \end{aligned}$$

Como conclusão desse processo todo de majorações para cada  $x$  escolhido, tem-se como principal resultado que o erro de truncamento gerado pela projeção de  $f$  em  $V_J$  decairá de forma exponencial, dependendo da escala  $J$  e do número de momentos nulos  $N$  da função *wavelet*:

$$|e_J(x)| = \mathcal{O}(2^{-JN}). \quad (2.12)$$

No caso da função de Haar,  $N = 1$ , o decaimento do erro de truncamento depende apenas da escala.

Além desse tipo de truncamento, uma outra operação pode ser definida a partir dos coeficientes *wavelets* da expansão da  $f$ , como será visto a seguir.

### 2.2.1 Operadores de Truncamento

Os operadores de truncamento que serão apresentados nesta seção foram introduzidos e sistematizados por David Donoho e Iain Johnstone [10] na busca por representações esparsas de sinais através de métodos envolvendo transformadas *wavelets*. Aqui o grande *insight* explorado é o poder de transformar sinais extremamente longos e cheios de variações (densos em quantidade de informação) em vetores esparsos com relativamente poucos valores significativos e não nulos, mas que ainda representam corretamente o sinal original em muitos aspectos. Essas representações esparsas trazem vantagens imediatas para aplicações envolvendo regularização, compressão de dados e redução de ruído [21], como será brevemente explorado nas seções de experimentos e aplicações.

### Hard e Soft Thresholding

Assim, dada a função inicial  $f$ , que representa o sinal a ser filtrado ou comprimido, obtém-se uma aproximação  $\hat{f}$  a partir da expansão *wavelet* de  $f$  e de um operador de truncamento  $\text{thr}_\lambda(\cdot)$  que reduz ou anula os coeficientes  $D_{j,k}$  menores do que um certo escalar  $\lambda$ , escolhido segundo algum critério. As duas propostas de truncamento mais difundidas [10], denominadas de *hard* e *soft thresholding*, são dadas, respectivamente, por:

$$\text{thr}_\lambda^H(D_{j,k}) = \begin{cases} 0, & \text{se } |D_{j,k}| \leq \lambda \\ D_{j,k}, & \text{se } |D_{j,k}| > \lambda \end{cases},$$

$$\text{thr}_\lambda^S(D_{j,k}) = \begin{cases} 0, & \text{se } |D_{j,k}| \leq \lambda \\ \text{sign}(D_{j,k})(|D_{j,k}| - \lambda), & \text{se } |D_{j,k}| > \lambda \end{cases},$$

sendo então a expansão resultante dada por:

$$\hat{f}(x) = \sum_{k=0}^{N_0-1} C_{J_0,k} \phi_{J_0,k}(x) + \sum_{j=J_0}^{J-1} \sum_{k=0}^{N_j-1} \text{thr}_\lambda(D_{j,k}) \psi_{j,k}(x). \quad (2.13)$$

A escolha entre *hard* ou *soft* e do limiar de corte  $\lambda$  é feita muitas vezes caso a caso, dependendo da aplicação. Contudo, os métodos de truncamento *hard* e *soft* não são os únicos possíveis. Vários métodos de truncamento de coeficientes *wavelets* podem ser encontrados na literatura, como os discutidos no trabalho de revisão [21].

De qualquer forma, uma questão relevante a ser analisada é o erro  $e_J$  cometido nestas operações de *thresholding* para que se possa classificar a aproximação  $\hat{f}$  como satisfatória ou não.

$$e_J = f(x) - \hat{f}(x) = \sum_{i=0}^{q-1} d_{j,i} \psi_{j,i}(x), \quad (2.14)$$

sendo  $\{d_{j,i}\}$  todos os coeficientes *wavelets* que foram descartados pela operação de truncamento, e  $q = \#\{d_{j,i}\}$  o número de elementos desse conjunto. As posições  $i$  neste somatório são na verdade associadas às verdadeiras posições que os coeficientes  $D_{j,k}$  originais assumiam na representação *wavelet* de  $f$ . Com isso, através da ortogonalidade da base *wavelet*, tem-se

$$\|e_J\|_2^2 = \|f(x) - \hat{f}(x)\|_2^2 = \langle f(x) - \hat{f}(x), f(x) - \hat{f}(x) \rangle = \sum_{i=0}^{m-1} |d_{j,i}|^2. \quad (2.15)$$

### M-best approximation

Uma outra possibilidade de se produzir representações esparsas (ou, pelo menos, com uma quantidade menor de coeficientes) a partir de sinais iniciais é através da estratégia de escolha dos  $m$ -primeiros maiores coeficientes *wavelet* da série *wavelet* original, descartando todos os demais. Esta estratégia é denominada

## 2.2. OPERAÇÕES DE TRUNCAMENTO

29

*m*-best approximation [31]. Desta forma, a aproximação obtida  $\hat{f}$  possui expansão contendo os coeficientes escala do nível mais grosseiro,  $J_0$ , e a contribuição dos  $m$  coeficientes *wavelet* com maior módulo, independentemente da escala,  $j = J_0, J_0 + 1, \dots, J - 1$  e da posição  $k$  referente à escala:

$$\hat{f}(x) = \sum_{k=0}^{N_0-1} C_{J_0,k} \phi_{J_0,k}(x) + \sum_{j=0}^{m-1} D_{\sigma(j,k)} \psi_{\sigma(j,k)}(x), \quad (2.16)$$

sendo  $\sigma(j, k)$  a permutação que faz a busca dos coeficientes  $D_{j,k}$  de maior módulo, de tal forma que  $|D_0| \geq |D_1| \geq |D_2| \dots \geq |D_{m-1}|$ .

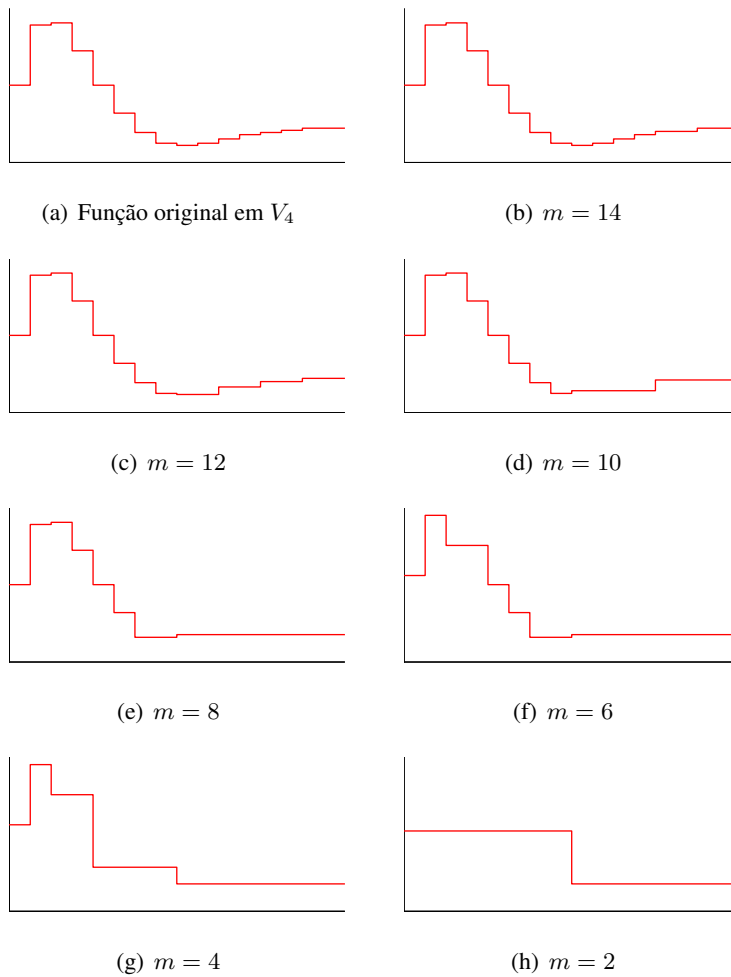


Figura 2.2: Aproximações para a função dependendo do número de coeficientes mantidos pela estratégia *m*-best approximation. Função original contida em  $V_4$ .

A Figura 2.2 mostra uma função em  $V_4$ , definida por uma expansão na base de Haar contendo inicialmente 16 amostras. Na sequência, são apresentadas as

representações desta função obtidas pelo *m-best approximation* considerando  $m = 14, 12, 10, \dots, 2$ .

Novamente, para se estimar o erro cometido na estratégia de *m-best approximation*, em relação à  $f$  originalmente dada em  $V_J$ , calcula-se

$$\|e_m\| = \|f(x) - \hat{f}(x)\|_2^2 = \langle f(x) - \hat{f}(x), f(x) - \hat{f}(x) \rangle = \sum_{j=m}^{N_J-1} |D_{\sigma(j,k)}|^2, \quad (2.17)$$

sendo considerados todos os demais  $N_J - m$  elementos cujos coeficientes  $D_{j,k}$  são, em módulo, menores do que  $|D_{m-1}|$ , definido pela escolha feita em 2.16. Assim, o erro é estimado pelo tamanho dos coeficientes descartados após a estratégia de truncamento.

A seguir, alguns experimentos computacionais serão propostos para se reforçar alguns pontos interessantes e importantes abordados até o momento.

## 2.3 Experimentos Computacionais

Esta seção está organizada em testes que ilustram aspectos já abordados nas seções anteriores, como a relevância da normalização da transformada, e a aplicação da transformada inversa para a construção das funções *wavelets* e escala.

### Teste 1

O Teste 1 avalia a implementação da TWD de Haar partindo de um vetor de comprimento  $J = 4$ , análogo ao exemplo inicial. A Tabela 2.1 ilustra os diferentes resultados obtidos pela aplicação da TWD de Haar não-normalizada e da normalizada sobre o vetor  $\mathbf{v}_1 = [8 \ 4 \ 5 \ 4]^T$ . O valor  $2^j$ ,  $0 \leq j \leq J$ , define o tamanho dos vetores analisados, sendo  $2^J$  o tamanho do vetor de entrada. Valores em ponto flutuante estão arredondados na quarta casa decimal.

Tabela 2.1: TWD de Haar 1D não-normalizada e normalizada sobre vetor  $\mathbf{v}_1$ .

TWD não-normalizada			TWD normalizada	
$j$	$C_j$	$D_j$	$C_j$	$D_j$
2	[8 4 5 4]	-	[8 4 5 4]	-
1	[6 4.5]	[2 0.5]	[8.4853 6.3640]	[2.8284 0.7071]
0	[5.25]	[0.75]	[10.5]	[1.5]

A Figura 2.3 apresenta uma representação simplificada e apenas qualitativa da decomposição normalizada *wavelet* do vetor  $\mathbf{v}_1 = [8 \ 4 \ 5 \ 4]^T$ .

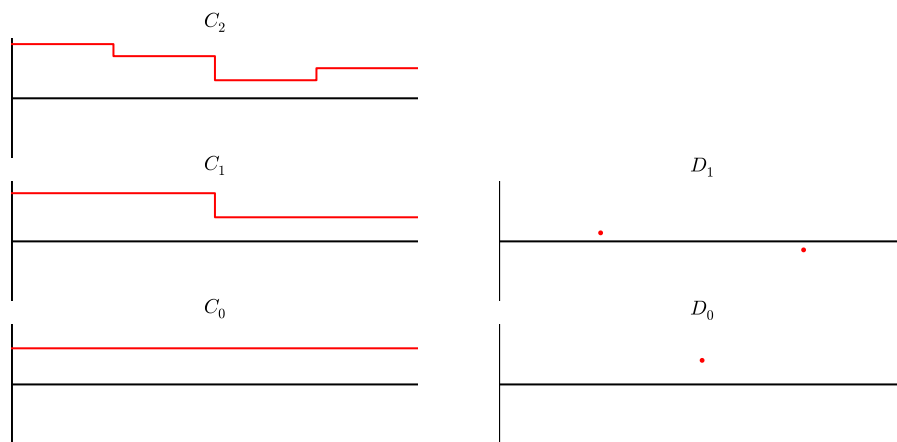


Figura 2.3: Gráfico da decomposição *wavelet*, através da TWD normalizada, para o vetor exemplo  $\mathbf{v}_1$ .

## Teste 2

Para o Teste 2 consideram-se duas operações (métricas): (a) o cálculo do produto interno de um vetor por ele mesmo  $\langle v, v \rangle = \sum_{i=1}^n (v_i)^2$ , também denominado em aplicações de análise de sinais como sendo a energia ao quadrado ( $E^2(v)$ ) do sinal e (b) a média  $\mu = 1/n \sum_{i=1}^n v_i$  dos elementos de um vetor  $v$  de tamanho  $n$ .

Nas Tabelas 2.2 e 2.3 são apresentados os valores dessas métricas aplicadas para os coeficientes de escala ( $C_j$ ) e de detalhe ( $D_j$ ) de todos níveis de decomposição do vetor  $\mathbf{v}_1$  do Teste 1. E ainda, sobre a representação multiresolução de  $\mathbf{v}_1$ ,  $C_{MR_j} = (C_j \ D_j \ D_{j+1} \ \cdots \ D_{J-1})$ , conforme Equação 1.40 para cada nível  $j$ .

Tabela 2.2: Energia ao quadrado dos coeficientes de detalhe e escala quando aplicadas as TWDs não normalizada e normalizada sobre o vetor  $\mathbf{v}_1$ .

$j$	TWD não normalizada			TWD normalizada		
	$E^2(C_j)$	$E^2(D_j)$	$E^2(C_{MR_j})$	$E^2(C_j)$	$E^2(D_j)$	$E^2(C_{MR_j})$
2	121,0	-	121,0	121,0	-	121,0
1	56,25	4,25	60,50	112,5008	8,4998	121,0006
0	27,5625	0,5625	32,375	110,25	2,25	120,9998

Percebe-se, através da análise das Tabelas 2.2 e 2.3, que a TWD normalizada preserva a energia total (a menos de erros computacionais de arredondamento) independentemente do nível de decomposição, enquanto que a TWD não-normalizada preserva sempre a mesma média para cada conjunto de coeficientes de

Tabela 2.3: Média dos coeficientes de detalhe e escala quando aplicadas as TWDs não normalizada e normalizada sobre o vetor  $\mathbf{v}_1$ .

$j$	TWD não-normalizada			TWD normalizada		
	$\mu(C_j)$	$\mu(D_j)$	$\mu(C_{MR_j})$	$\mu(C_j)$	$\mu(D_j)$	$\mu(C_{MR_j})$
2	5,25	-	5,25	5,25	-	5,25
1	5,25	1,25	3,25	7,4246	1,7677	4,5962
0	5,25	0,75	3,0	10,5	1,5	6,0

escala, também independentemente do nível de decomposição. Como consequência, os algoritmos de *thresholding* para aproximar funções poderiam ser parametrizados de acordo com os filtros utilizados (normalizados ou não) na TWD.

### Teste 3

Este teste é responsável pela avaliação da TWD Inversa. O vetor exemplo  $\mathbf{v}_1$  do Teste 1, mais uma vez foi tomado para realizar os testes. Os resultados da TWD Inversa não normalizada foram exatos, enquanto o da normalizada sofreu truncamento (problemas de arredondamento). O vetor reconstituído,  $\mathbf{v}'_1$ , através da aplicação da TWD Inversa normalizada é

$$\mathbf{v}'_1 = [8.0000000000002693 \ 3.999999999973053 \ 5.000000000000032 \ 3.9999999999968].$$

### Teste 4

O Teste 4 ilustra uma propriedade extremamente relevante da família de Daubechies, que é a relação biunívoca entre a função e seu filtro correspondente. A função de Haar é a única da família de Daubechies que possui expressão analítica, todas as demais são obtidas apenas através de seus filtros. O procedimento para obter os gráficos de qualquer função desta família é através da avaliação da TWD Inversa a partir de um grosseiro nível  $J_0$ . Como dados de entrada para a TWD Inversa são considerados um par de vetores, um completamente nulo e outro com valor igual à 1 apenas em uma posição, sendo todas as demais nulas. Assim, para serem reconstruídas as funções escala, assume-se  $C_{J_0}$  como sendo o vetor com um único valor não nulo e o vetor de coeficientes *wavelet*  $D_{J_0}$  como sendo nulo. No caso da construção dos valores das funções *wavelets*, o vetor  $C_{J_0}$  é nulo e o vetor  $D_{J_0}$  possui uma posição não nula. A posição do valor não nulo no vetor, faz com que a função seja gerada com deslocamento correspondente.

Desta forma são obtidas as funções escala e *wavelet* de Haar:  $\phi(x - k)$  e  $\psi(x - k)$ , com  $k = 0$  e  $1$ . A Figura 2.4 mostra a função escala com deslocamento  $k = 0$ , assumindo-se  $C_2 = [1, 0]$  e  $D_2 = [0, 0]$  e para  $k = 1$  os dados iniciais são  $C_2 = [0, 1]$  e  $D_2 = [0, 0]$ . Para a função *wavelet* com deslocamento  $k = 0$  tem-se  $C_2 = [0, 0]$  e  $D_2 = [1, 0]$  e finalmente para  $k = 1$ ,  $C_2 = [0, 0]$  e  $D_2 = [0, 1]$ .

## 2.4. FILTRAGEM WAVELET

33

Estas funções são construídas após 12 níveis de reconstrução através de aplicações sucessivas da TWD Inversa não normalizada.

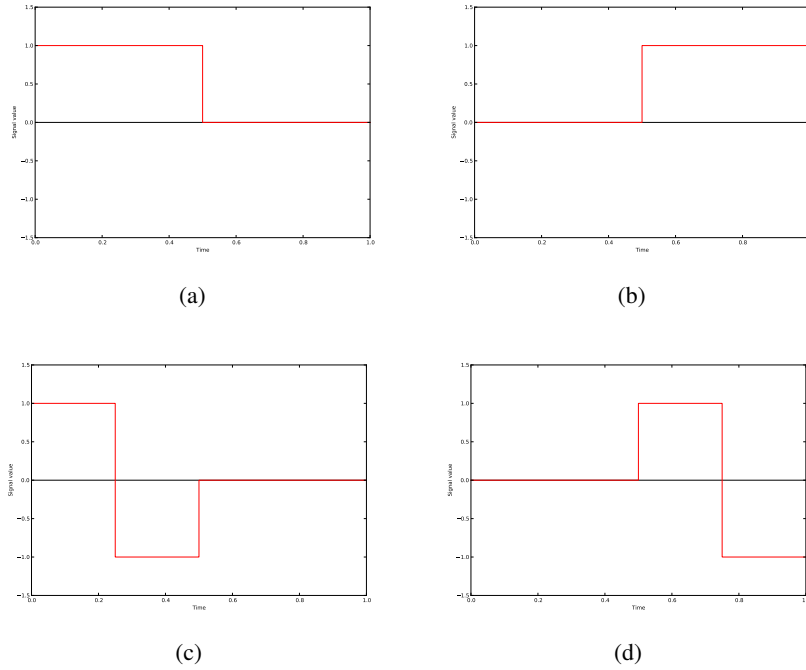


Figura 2.4: Função escala com deslocamento (a)  $k = 0$  e (b)  $k = 1$ . Função *wavelet* com deslocamento (c)  $k = 0$  e (d)  $k = 1$ .

## 2.4 Filtragem wavelet

Quando informações discretas são obtidas através de medições ou manipuladas por aplicativos computacionais e então armazenadas digitalmente, em alguma etapa deste processo, estas podem ser corrompidas por diversas formas de interferência, denominadas de ruídos. As técnicas de limiarização dos coeficientes de uma série *wavelet* têm como objetivo a redução, ou mesmo eliminação, do ruído presente em um sinal [21].

Estas técnicas estão baseadas na manipulação dos coeficientes *wavelet*, que podem ter seus valores diminuídos ou anulados, como foi visto na Seção 2.2, nos vários níveis de fatoração gerados pela TWD Direta [2]. Após esse processo, a TWD Inversa é aplicada na expansão *wavelet* com coeficientes truncados [21]. O sinal reconstruído será uma aproximação do sinal sem ruído, sendo, portanto, o resultado da filtragem.

Considera-se o vetor  $N_J$ -dimensional  $\mathbf{f} = (f_0, f_1, \dots, f_{N_J-1})$ , com  $N_J = 2^J$  e  $J \in \mathbb{Z}^+$ , sendo cada componente uma amostra do sinal sem ruído  $f$  em um



instante  $t_i$ , ou seja  $f_i = f(t_i)$ . Considera-se ainda o vetor de ruído gaussiano  $\epsilon = (\epsilon_0, \epsilon_1, \dots, \epsilon_{N_J-1})$ , cujas componentes são variáveis aleatórias independentes provenientes de uma distribuição normal (Gaussiana) com média 0 e desvio padrão  $\sigma^2$ ,  $\epsilon_i \sim N(0, \sigma^2)$ , para  $i = 0, 1, \dots, N_J - 1$ . Desta forma, um sinal amostrado  $\mathbf{y}$  pode ser considerado como uma soma de uma componente limpa  $\mathbf{f}$  e um ruído  $\epsilon$ ,  $\mathbf{y} = \mathbf{f} + \epsilon$ , ou seja, coordenada a coordenada assume-se:

$$(y_0, y_1, y_2, \dots, y_{N_J-1}) = (f_0 + \epsilon_0, f_1 + \epsilon_1, f_2 + \epsilon_2, \dots, f_{N_J-1} + \epsilon_{N_J-1}).$$

Na prática, a única informação conhecida é o vetor  $\mathbf{y}$ , e o objetivo então é, a partir de  $\mathbf{y}$ , obter uma estimativa para a componente  $\mathbf{f}$ .

Assim, pela linearidade da TWD :

$$\text{TWD}(\mathbf{y}) = \text{TWD}(\mathbf{f} + \epsilon) = \text{TWD}(\mathbf{f}) + \text{TWD}(\epsilon).$$

Por ser ortogonal, a TWD ainda preserva a energia do sinal analisado. Portanto, no caso deste sinal ser um ruído gaussiano, suas propriedades de média zero e variância constante  $\sigma^2$  são preservadas após a transformação. Em termos de coeficientes *wavelets*, esta constatação pode ser expressa por:  $d_{j,k} = w_{j,k} + \sigma z_{j,k}$ , sendo  $d_{j,k}$  os coeficientes *wavelet* de  $\mathbf{y}$ ,  $w_{j,k}$  os coeficientes *wavelets* de  $\mathbf{f}$  e  $z_{j,k} \sim N(0, 1)$  variáveis aleatórias provenientes de distribuição normal com média 0 e desvio padrão 1. Ou seja, os coeficientes *wavelets* de uma amostra com ruído podem ser escritos como os coeficientes *wavelets* sem ruído adicionados a ruído branco, dado por  $z_{j,k} \sim N(0, 1)$  [21].

### Algoritmo para Filtragem

Dada a função  $\text{thr}_\lambda(\cdot)$ , o procedimento para filtragem do sinal  $\mathbf{y}$  e obtenção da aproximação para o sinal filtrado  $\hat{\mathbf{f}}$ , por meio do truncamento dos coeficientes *wavelets*, segue as seguintes etapas:

1. Aplicação da TWD direta no sinal original em J níveis:  
 $\text{TWD}_J(\mathbf{y}) = \mathbf{d} = (C_{J_0} \ D_{J_0} \ D_{J_0-1} \ \dots \ D_{J-1});$
2. Truncamento dos coeficientes *wavelets* da decomposição  $\mathbf{d}$  :  
 $\text{thr}_\lambda(\mathbf{d}) = \tilde{\mathbf{d}} = (C_{J_0} \ \tilde{D}_{J_0} \ \tilde{D}_{J_0-1} \ \dots \ \tilde{D}_{J-1});$
3. Aplicação da TWD Inversa na série com coeficientes *wavelets* truncados:  
 $\text{TWDI}_J(\tilde{\mathbf{d}}) = \hat{\mathbf{f}}.$

No final do procedimento de filtragem é obtida uma estimativa  $\hat{\mathbf{f}}$  do sinal sem ruído  $\mathbf{f}$ . A Figura 2.5 mostra uma representação desse procedimento. É considerado como entrada o sinal ruidoso  $\mathbf{y}$ , composto por uma curva sem ruído  $f$  e um ruído aleatório,  $\epsilon$ , com média 0 e desvio padrão 1,  $\epsilon \sim N(0, 1)$ , apresentado na Figura 2.5(a). O resultado final, obtido com 2 níveis de decomposição *wavelet* de Haar e *hard thresholding* com  $\lambda = 5$ , está dado na Figura 2.5(b).

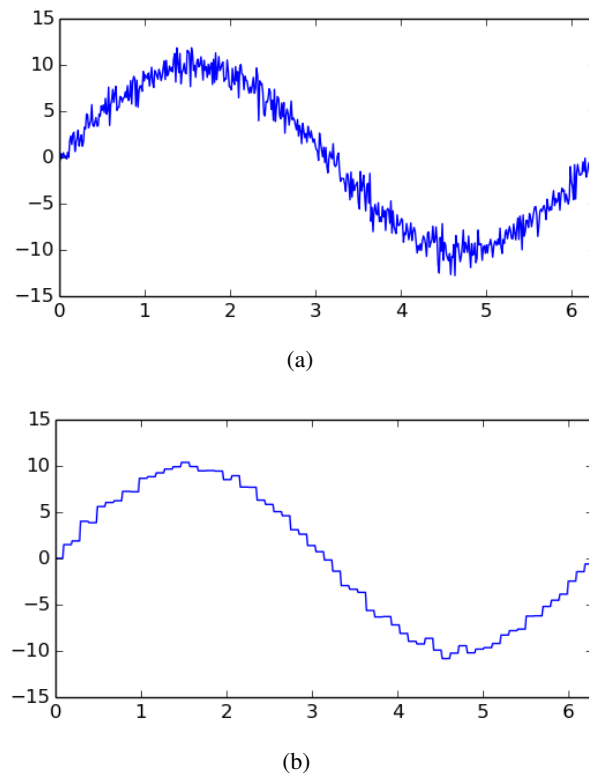


Figura 2.5: Sinal ruidoso  $y = f + \epsilon$ , com  $f = 10 \sin(x)$ ,  $x \in [0, 2\pi]$ ,  $\epsilon \sim N(0, 1)$  em (a) e sua versão filtrada a partir da decomposição *wavelet* em dois níveis, utilizando *hard thresholding* com  $\lambda = 5$  em (b).

## 2.5 Análise de Sinais EEG

Esta seção tem como objetivo principal ilustrar os tópicos apresentados e discutidos até o momento através da apresentação de uma aplicação prática e associada a um problema real relevante: a análise de sinais cerebrais. A discussão feita aqui é apenas uma pequena parte da Dissertação de Mestrado [27], cujo objetivo é a classificação de estágios de sono a partir de sinais de eletroencefalograma (EEG) provenientes de uma única derivação de eletrodos, comumente denominados canais de aquisição. Estes canais são as especificações de onde os eletrodos foram posicionados no escalpo no momento da captação do sinal.

O reconhecimento e análise dos diferentes padrões cerebrais associados aos estágios de sono é extremamente importante no auxílio à detecção e ao tratamento de doenças e distúrbios do sono. Estes, além de afetar o desempenho das mais diferentes tarefas, são indicados como causas primeiras de vários acidentes, como por exemplo os de trânsito.

Cada vez que um problema prático é abordado, termos técnicos específicos precisam ser definidos e apresentados, assim como conceitos específicos referentes

à física associada na descrição e modelagem do problema. Assim, é inevitável que alguns termos sejam utilizados aqui, no entanto estes serão em quantidade reduzida, apenas para que o leitor possa se familiarizar com a terminologia mínima indispensável.

Os sinais de EEG considerados em [27, 29] foram extraídos de uma base de dados pública, *PhysioBank* [13], disponibilizada pelo MIT (*Massachusetts Institut of Technology*). Esses sinais são armazenados em formato específico e foram convertidos para que pudessem ser tratados como vetores de dados. Estes vetores podem ser enormes, pois em muitos casos um sinal de EEG é captado durante toda uma noite ou por muitas horas, dependendo do evento (ou anomalia) a ser monitorado. Outro fator importante que contribui para o aumento do volume de dados é a taxa de amostragem (frequência de amostragem) com que o sinal é captado, fazendo com que para cada segundo sejam captadas 100 amostras (100Hz).

Uma convenção entre os especialistas desta área é a partição desses sinais em pequenas partes, chamadas de épocas, para que possam ser analisadas e classificadas. Para simplificação do problema, consideram-se aqui épocas com  $2^J$  amostras. O caso padrão com épocas de 3000 amostras é estudado em [27, 29]. Uma propriedade importante da TWD, e que é fundamental neste problema de classificação de estágios de sono, é o fato de cada escala (ou nível de resolução) da TWD estar associada a faixas de frequências. Essa associação é de interesse para o estudo em voga, uma vez que há ritmos cerebrais com frequências bem definidas. Uma vez que, pelo teorema de Nyquist [15], só se pode representar frequências até a metade da frequência de amostragem, pode-se desmembrar, via decomposição *wavelet*, faixas de frequências entre 0 e 50Hz, para dados amostrados a 100Hz. Na Figura 2.6 são apresentadas as faixas de frequências que correspondem a cada um dos 5 níveis da TWD, assim como os ritmos cerebrais que são perceptíveis nestas faixas.

Nota-se, através da Figura 2.6, que cinco níveis de decomposição são necessários para que se possa representar as informações de uma época nas faixas de frequência próximas às dos ritmos baixo-gama, beta, alfa, teta e delta. Ao determinar quais informações estão associadas a quais faixas de frequência, é possível com o auxílio métodos estatísticos (e ferramentas computacionais) classificar padrões de sono. A Figura 2.7 ilustra o processo de decomposição de uma época seguindo o mesmo esquema da Figura 2.6. A partir da decomposição gerada pela TWD aplicada a cada uma das épocas podem ser extraídas novas informações associadas ao sinal cerebral analisado. Em [27] um esquema completo de classificação de diferentes estágios de sono é apresentado e validado computacionalmente, utilizando abordagem semelhante a descrita aqui, porém com a transformada *wavelet* de Daubechies, não com um momento nulo, mas sim com dois.

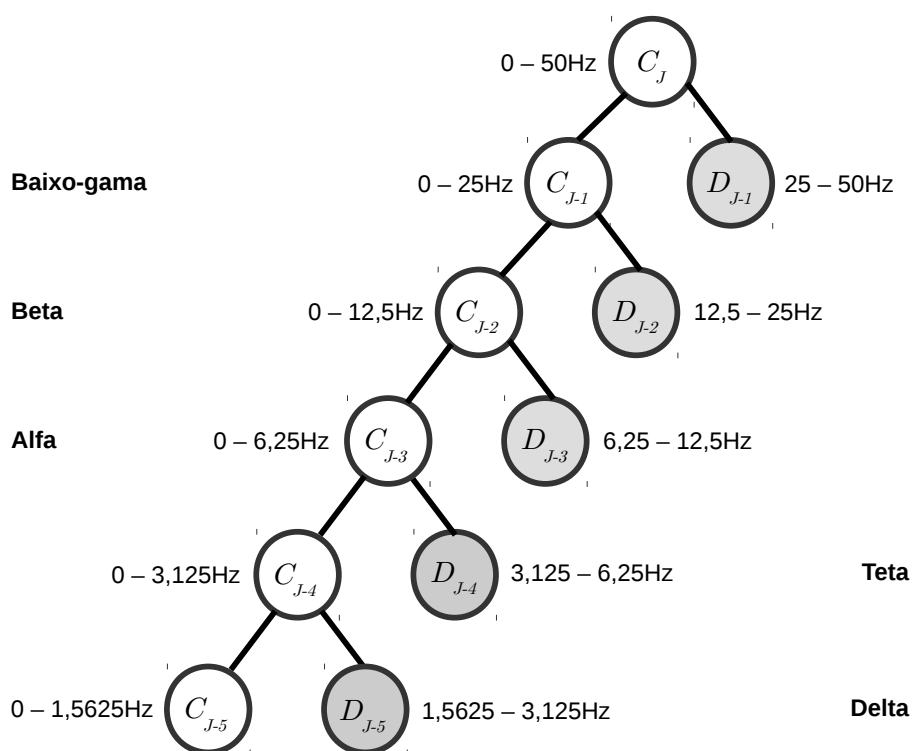


Figura 2.6: Decomposição *wavelet* em cinco níveis, faixas de frequência em cada nível e ritmos cerebrais associados. Este esquema de decomposição *wavelet* considera o sinal de entrada  $C_J$  com frequência de amostragem de 100Hz.

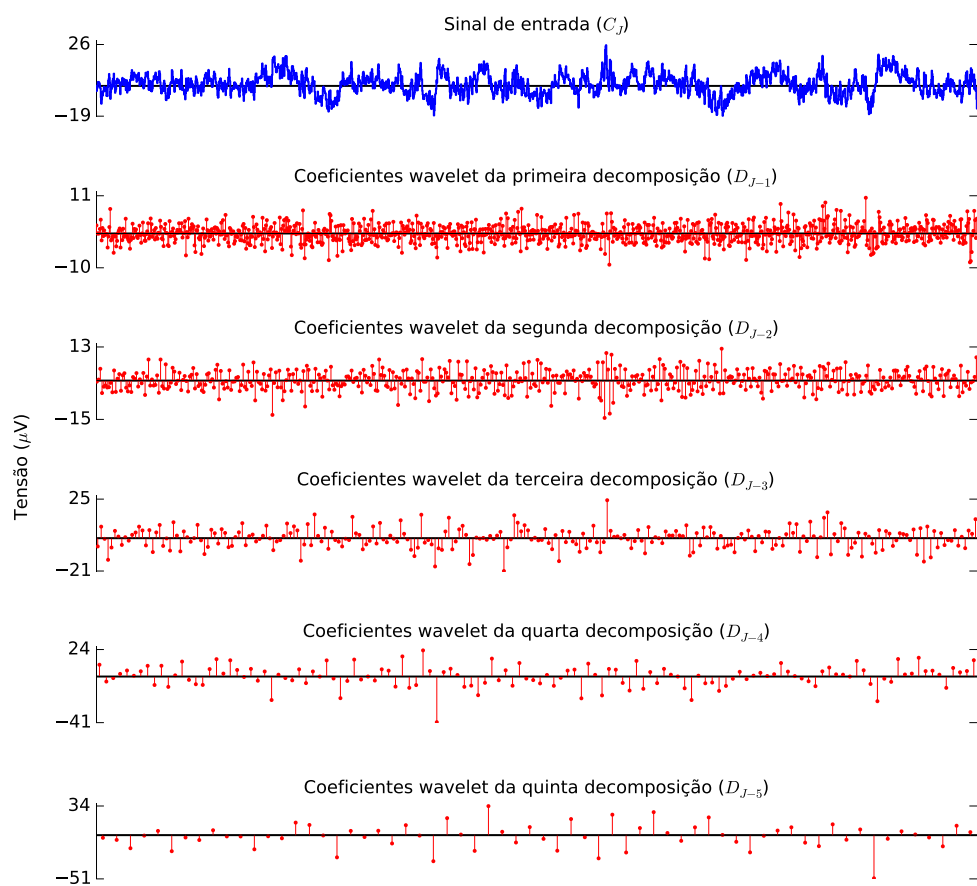


Figura 2.7: Decomposição *wavelet* em cinco níveis de uma época de EEG com 2048 ( $J = 11$ ) amostras.

## Capítulo 3

# TWD 2D de Haar

O objetivo principal deste capítulo é apresentar a TWD 2D de forma mais simplificada possível e explorar diretamente seu grande potencial para análise e compressão de dados bidimensionais (que podem ser imagens [1], séries temporais [32], soluções de equações diferenciais [11], amostras de sensores de segurança de redes de computadores [19], entre outros). Através de seus algoritmos decimados, considerados como consequência imediata da formulação em Cascata para os algoritmos rápidos da TWD 1D, serão explorados alguns aspectos gerais para todas as funções *wavelets* de Daubechies. Novamente, muitas das propriedades e os exemplos serão apresentados apenas para a TWD de Haar 2D.

Este capítulo será apresentado de acordo com [31], cujo enfoque é mais em relação aos algoritmos do que em relação à estrutura de análise multiresolução associada à base de funções de  $V = L^2(\mathbb{R}^2)$ . Esta é construída e abordada em detalhes em [24], utilizando tópicos de Teoria da Medida como ferramenta para demonstrações e desenvolvimento da teoria, sendo portanto fora do escopo deste texto introdutório.

### 3.1 Base para $L^2(\mathbb{R}^2)$

As funções escala  $\phi(x, y)$  e *wavelets*  $\psi^i(x, y)$  no caso 2D são funções a duas variáveis reais, cujas translações e dilatações são, de forma análoga ao caso ortogonal 1D, denotadas por:

$$\phi_{j,k,l}(x, y) = 2^{j/2} \phi(2^j x - k, 2^j y - l), \quad (3.1)$$

$$\psi_{j,k,l}^i(x, y) = 2^{j/2} \psi(2^j x - k, 2^j y - l), \quad i \in \{H, V, D\}. \quad (3.2)$$

Diferentemente da base de  $L^2(\mathbb{R})$ , agora tem-se três funções *wavelets* distintas e uma única função escala para compor a base de  $V = L^2(\mathbb{R}^2)$ . Como toda base de um espaço vetorial, a ordem com que os elementos da base são considerados implica na obtenção de uma nova base para o espaço. Este fato poderá ser experimentado diretamente nas formulações algorítmicas para a TWD 2D de Haar.

Assume-se como hipótese fundamental, como proposto em [9, 7], que as funções escala e *wavelets* são separáveis e suas definições são dadas por:

$$\phi(x, y) = \phi(x)\phi(y) \quad (3.3)$$

$$\psi^H(x, y) = \psi(x)\phi(y) \quad (3.4)$$

$$\psi^V(x, y) = \phi(x)\psi(y) \quad (3.5)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (3.6)$$

As ilustrações dessas quatro funções são dadas na Figura 3.1.

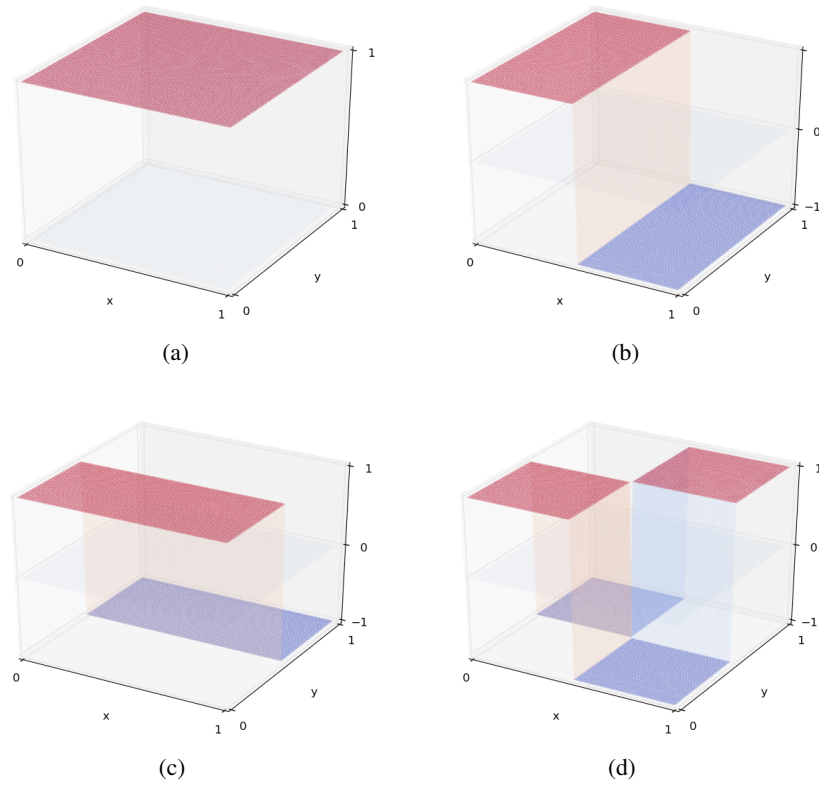


Figura 3.1: Bases de Haar 2D:  $\phi(x, y)$  em (a);  $\psi^H(x, y)$  em (b);  $\psi^V(x, y)$  em (c) e  $\psi^D(x, y)$  em (d).

Novamente, a função escala  $\phi(x, y)$  tem como principal tarefa representar as informações médias da função bidimensional (como uma imagem). As funções *wavelets* horizontal  $\psi^H(x, y)$ , vertical  $\psi^V(x, y)$  e diagonal  $\psi^D(x, y)$  recebem esta nomenclatura com o objetivo de evidenciar a maneira com que os detalhes (as diferenças de informação em relação à média) são calculados, considerando as possíveis direções de variação nos dados bidimensionais.

Nesta seção omite-se a expansão de uma função qualquer  $f \in V$  com respeito à base *wavelet* bidimensional. Passa-se portanto à apresentação das formulações algorítmicas mais consideradas para a TWD 2D de Haar baseada em algoritmos de Cascata, portanto, envolvendo formulações decimadas para esta transformação.

Considera-se como dado de entrada, independentemente do algoritmo selecionado, uma imagem monocromática  $\mathcal{I}$ , armazenada em uma matriz  $M_{m \times n}$ , com  $m$  e  $n$  valores dados em potência de 2. Cada elemento  $m_{i,j}$  da matriz  $M$  é um valor inteiro entre 0 e 255 que indica o nível de cinza correspondente ao pixel pertencente à imagem de entrada na posição  $(i, j)$ . Como consequência da escolha de funções separáveis para formar a base de representação de dados bidimensionais, tem-se a possibilidade de tratar a TWD 2D como sendo uma sucessão de transformadas unidimensionais aplicadas aos vetores linha ou vetores coluna da matriz de entrada  $M_{m \times n}$ . A seguir, apresentam-se os casos mais tradicionais para o contexto da TWD 2D de Haar.

### 3.1.1 Algoritmo *standard* da TWD 2D

A primeira possibilidade de algoritmo para a TWD 2D de Haar, denominado decomposição *standard* [31], é aplicar a TWD 1D da Haar a todas as linhas da matriz  $M_{m \times n}$ . Com isso, no final deste processo, os dados da matriz devem estar transformados e armazenados nas mesmas linhas, apenas com a organização de que a primeira metade dos elementos da linha seja para os dados médios (coeficientes de escala) e a segunda metade para os dados de detalhe (coeficientes *wavelets*), sendo denotada a nova matriz por  $M^{(1)}$ . O próximo nível da transformada é então aplicada a TWD 1D de Haar, novamente linha por linha, mas agora apenas na primeira metade dos elementos de cada linha de  $M^{(1)}$ , ou seja, apenas os coeficientes escala no nível anterior serão novamente decompostos, gerando então a nova matriz  $M^{(2)}$ , cujos dados agora estão organizados em três blocos distintos  $[C^2 D^2 D^1]$ , sendo cada um deles com o mesmo número de linhas  $m$ .  $C^2$  e  $D^2$  com  $n/4$  colunas e  $D^1$  com  $n/2$  colunas. Este processo segue então por  $j$  níveis, até que sejam obtidas apenas uma coluna em cada bloco  $C^j$  e  $D^j$ .

Com isso, metade do processo de decomposição em  $j$  níveis estará concluído. Metade, pois o processo foi aplicado em relação às linhas de  $M$ . Para se obter a transformada completa, agora as colunas da matriz  $M^{(j)}$ , obtida como o final do processo de decomposição por linhas, serão decompostas em  $j$  níveis pela aplicação da TWD 1D de Haar. A maneira como os dados bidimensionais são decompostos via algoritmo *standard* é exemplificada na Figura 3.2 e algorítmicamente apresentada em Algoritmo 3.



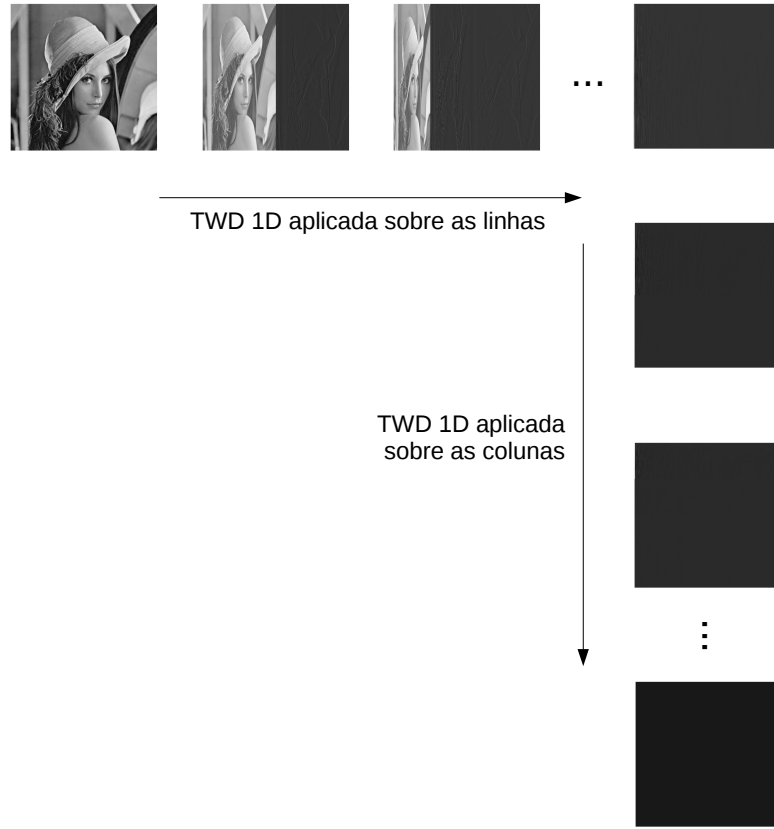


Figura 3.2: Exemplo de aplicação do algoritmo *standard* da TWD 2D.

---

**Algoritmo 3** TWD de Haar 2D *Standard*

---

- 1: **Para**  $l \leftarrow 0$  **até**  $m - 1$  **faça** ▷ Linhas  $l$
  - 2:      $M_{l;0,1,\dots,n-1} \leftarrow \text{TWD}_{\log(n)}(M_{l;0,1,\dots,n-1})$  ▷ TWD em  $j = \log(n)$  níveis
  - 3: **Fim Para**
  - 4: **Para**  $c \leftarrow 0$  **até**  $n - 1$  **faça** ▷ Colunas  $c$
  - 5:      $M_{0,1,\dots,m-1;c} \leftarrow \text{TWD}_{\log(m)}(M_{0,1,\dots,m-1;c})$  ▷ TWD em  $j = \log(m)$  níveis
  - 6: **Fim Para**
- 

### 3.1.2 Algoritmo não-*standard* da TWD 2D

A segunda alternativa para a execução da TWD 2D de Haar, novamente considerando sucessivas aplicações da TWD 1D, é denominada *decomposição não-standard* [31]. Desta vez, as decomposições por linhas e por colunas serão intercaladas a cada nível, gerando uma diferente organização dos dados intermediários. A forma com que esses dados são gerados é ilustrada na Figura 3.3 e apresentada no Algoritmo 4.

### 3.1. BASE PARA $L^2(\mathbb{R}^2)$

43

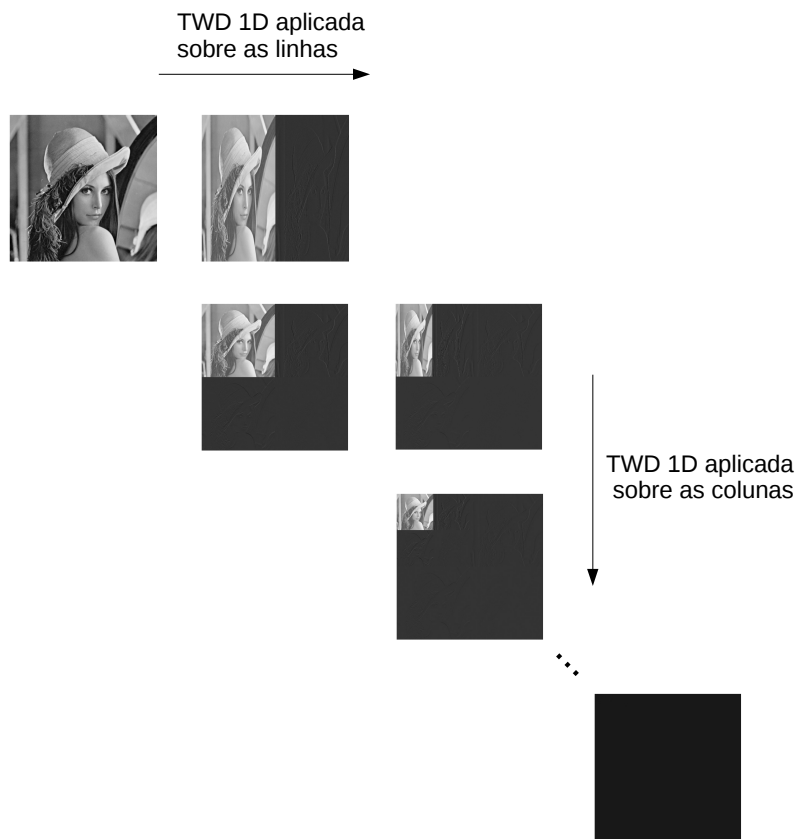


Figura 3.3: Exemplo de aplicação do algoritmo não-*standard* da TWD 2D.

#### Algoritmo 4 TWD de Haar 2D Não-*standard*

---

```

1:  $h \leftarrow m$  ▷ Por simplicidade  $m = n$ 
2: Enquanto  $h > 1$  faça
3:   Para  $l \leftarrow 0$  até  $h - 1$  faça ▷ Linhas  $l$ 
4:      $M_{l;0,1,\dots,h-1} \leftarrow \text{TWD}_1(M_{l;0,1,\dots,h-1})$  ▷ TWD em  $j = 1$  nível
5:   Fim Para
6:   Para  $c \leftarrow 0$  até  $h - 1$  faça ▷ Colunas  $c$ 
7:      $M_{0,1,\dots,h-1;c} \leftarrow \text{TWD}_1(M_{0,1,\dots,h-1;c})$  ▷ TWD em  $j = 1$  nível
8:   Fim Para
9:    $h \leftarrow h/2$ 
10: Fim Enquanto

```

---

Os algoritmos apresentados para a TWD 1D e para a TWD 2D, ambos para a *wavelet* de Haar, possuem uma simplificação em relação aos seus análogos, quando a base de funções é formada por *wavelets* com mais de  $N = 1$  momento nulo e portanto com filtros de tamanho  $D = 2N$  maior do que dois. Essa simplificação

diz respeito a como os dados são tratados nas vizinhanças dos pontos da fronteira. Para as TWD's 1D e 2D de Haar nenhum valor precisa ser extrapolado para que se possa obter a transformação nos pontos da fronteira. No caso das demais funções de Daubechies, são necessários  $D - 2$  valores extrapolados à direita para a TWD direta e outros  $D - 2$  valores extrapolados à esquerda para se calcular a TWD inversa. Outro ponto importante a ser destacado é que a TWD 2D de Haar inversa, considerando tanto o algoritmo *standard* quanto o não-*standard*, pode ser obtida seguindo os passos de cada algoritmo na ordem reversa.

## 3.2 Aplicação em análise de imagens

A seguir são apresentados dois exemplos de problemas referentes à análise de imagens: filtragem e compressão. Essa área, na verdade, engloba outros métodos numéricos e estatísticos para as mais variadas aplicações, como reconhecimento e classificação de padrões, determinação de textura, identificação de faces, etc.

### 3.2.1 Filtragem de imagens

A aplicação em foco agora é a filtragem (ou remoção de ruído) de imagens. Assim como explorado na Seção 2.4 com sinais 1D, o processo de filtragem pode ser realizado através de três etapas, : (a) decomposição *wavelet* da imagem em  $j$  níveis, Seções 3.1.1 e 3.1.2; (b) operação de *thresholding* aplicada aos coeficientes *wavelet* da decomposição em cada nível, Seção 2.4; (c) inversão da expansão truncada da imagem, seguindo a heurística proposta no final da seção anterior.

O imagem escolhida para ilustrar o procedimento de filtragem é um clássico e vem sendo amplamente utilizada como problema teste para as mais diferentes técnicas de análise de imagens através de *wavelets*. A Figura 3.4(a) mostra a imagem Lena original de  $512 \times 512$  (vista como uma função  $f$ ). A Figura 3.4(b) ilustra um ruído Gaussiano 2D ( $\epsilon$ ). Como na Seção 2.4, soma-se  $f$  e  $\epsilon$  e obtém-se  $y$ , uma representação da imagem Lena com ruído, que é apresentada na Figura 3.4(c). A Figura 3.4(d) apresenta o resultado de uma das várias possibilidades de filtragem da imagem da Figura 3.4(c). Aqui optou-se por utilizar a estratégia de *hard threshold* com limiar de corte  $\lambda = 20$  após a decomposição em quatro níveis pelo Algoritmo 4. A partir dos dados truncados aplica-se a TWD 2D de Haar inversa para se obter uma aproximação para a imagem Lena sem ruído.

Naturalmente, uma questão relevante, e que é um tópico atual de pesquisa, é o critério de escolha do parâmetro de corte  $\lambda$ . Aqui este valor foi escolhido por representar menos do que 10% do valor máximo que um coeficiente *wavelet* poderia assumir. Para avaliação do resultado da filtragem (e da escolha dos parâmetros) utiliza-se o *peak signal-to-noise ratio* (PSNR) [17], uma estimativa para a taxa de ruído entre dois sinais, o original  $I_{m \times n}$  e o outro processado  $I'_{m \times n}$ :

$$PSNR = 10 \log_{10} \left( \frac{256^2(m \times n)}{\sum_{l=0}^m \sum_{c=0}^n (I_{l,c} - I'_{l,c})^2} \right).$$

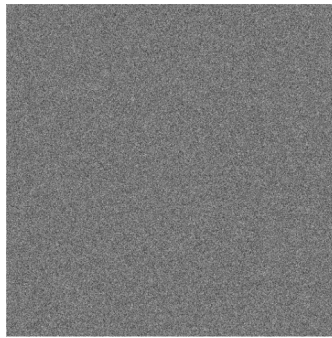
### 3.2. APLICAÇÃO EM ANÁLISE DE IMAGENS

45

Quanto maiores os valores de PSNR, mais próximos são os sinais (imagens), isto é, a imagem reconstruída com relação a sua versão original. Note, portanto, na Figura 3.4, que através da filtragem *wavelet* consegue-se reduzir a quantidade de ruído que, neste experimento, foi propositalmente introduzido.



(a) Imagem original

(b) Ruído Gaussiano  $\sim N(0, 10)$ 

(c) Versão ruidosa (PSNR=28,127dB)



(d) Versão filtrada (PSNR=30,262dB)

Figura 3.4: Imagem original em (a), ruído Gaussiano aditivo (somado ao valor 128 para melhor visualização) em (b), versão ruidosa em (c) e versão filtrada em (d).

#### 3.2.2 Compressão de imagens

Outra aplicação potencial é a de compressão de imagens. Aqui, o objetivo é reduzir a quantidade de dados necessários para armazenamento ou transmissão de sinais (imagens) ao mesmo tempo que se preserva sua qualidade ao máximo (ou em um nível aceitável dependendo da aplicação). O procedimento de compressão é similar à filtragem, sendo uma operação de *thresholding*, cujo objetivo é retirar coeficientes de detalhe que contribuem pouco (são próximos de zero) para a representação da imagem e, portanto, poderiam ser descartados sem causar danos "graves" ao resultado. Essa retirada de coeficientes de detalhe permite então a compressão de dados. Aqui é utilizada *compression ratio* (CR) [14] que mede a relação

de compressão dos dados, isto é a porcentagem de coeficientes retirados sobre o total de coeficientes de escala e detalhe ( $m \times n$ ). A Figura 3.5 apresenta algumas versões comprimidas da imagem Lena, considerando *hard thresholding* associado à decomposição *wavelet* não-*standard*.



(a) Imagem original



(b) CR = 93,72% e PSNR=25,250dB



(c) CR = 97,05% e PSNR=29,280dB



(d) CR = 76,54% e PSNR=29,961dB

Figura 3.5: Imagem original em (a) e suas versões comprimidas utilizando (b) decomposição em cinco níveis e  $\lambda = 200$ , (c) decomposição completa e  $\lambda = 50$  e (d) decomposição em três níveis e  $\lambda = 150$ .

Novamente, para que se possa atingir o melhor resultado de compressão, não há uma configuração padrão dos parâmetros, ou seja, valores previamente fixados e universais para  $\lambda$  e  $j$ . Pelo contrário, estudos sobre as propriedades do ruído e da imagem devem ser realizados para se definir da melhor forma possível tais parâmetros, dependendo de cada aplicação. Com isso, abre-se toda uma nova linha de pesquisa e desenvolvimento de métodos adaptativos, nos quais a determinação destes parâmetros depende dos dados de entrada em si, podendo ser ainda ajustados automaticamente ao longo das etapas dos procedimentos numéricos envolvidos.

## Capítulo 4

# Transformada *wavelet* Packet de Haar

As *wavelet* Packets, propostas por Coifman, Meyer and Wickerhauser [8], podem ser interpretadas como uma generalização das funções ortonormais de suporte compacto da família de Daubechies. De forma simplificada, uma *wavelet* Packet  $\psi \in L^2(\mathbb{R})$  pode ser descrita como sendo uma forma de onda modulada de quadrado integrável com média zero, bem localizada tanto em posição quanto em frequência, possuindo agora três parâmetros livres (variáveis independentes): frequência, escala e posição. Assim como as *wavelets* de Daubechies, as Packets com suas translações, dilatações e modulações (variações na frequência) também formam uma base ortonormal para  $L^2(\mathbb{R})$ . Na verdade, quando tomadas ao limite, elas formam uma infinidade de bases ortonormais para  $L^2(\mathbb{R})$ . Quando aproximadas por vetores do  $\mathbb{R}^n$ , então existe um conjunto de  $n \log(n)$  desses vetores que as representam. Assim, uma função  $f \in L^2(\mathbb{R})$  pode ser expressa nestas bases de *wavelet* Packets, e os coeficientes na expansão são as projeções da  $f$  na direções dos elementos da base. Quando  $f \in V_J$ , a base *wavelet* Packet terá então  $n \log(n)$ , com  $n = 2^J$ , elementos. Neste caso, o algoritmo rápido para a transformada *wavelet* Packet será a maneira eficiente de se calcular os coeficientes de  $f$  nesta base. Neste capítulo, o enfoque a ser dado novamente é em relação ao algoritmo associado a esta transformação. E os aspectos funcionais e teóricos pertinentes podem ser encontrados em textos mais específicos, como [8].

### 4.1 Formulação

Na Seção 2.5, a relação entre escalas da TWD 1D e faixas de frequência foi explorada na aplicação referente a reconhecimento de padrões de sono e ilustrada através da Figura 2.6. Na verdade, existe uma infinidade de outras aplicações que necessitam de localização tanto em relação ao tempo, quanto em relação à frequência. Esta é uma tarefa difícil de ser atingida, especialmente porque até metade do século passado a melhor ferramenta que se conhecia eram as transformadas de Fou-

rier janeladas para se obter o melhor compromisso entre tempo e frequência. Com o surgimento das *wavelets* foram criadas várias alternativas para se tentar mais definição em ambos os parâmetros ao mesmo tempo. As *wavelet* Packets têm este objetivo.

A Transformada *wavelet* Packet (TWP), em termos de algoritmo, pode ser vista como uma extensão natural da TWD 1D, uma vez que, a princípio, não haveria nenhuma restrição em se decompor em vários níveis as componentes relativas aos coeficientes *wavelet*. Assim, a faixa de frequência associada ao primeiro bloco de detalhes,  $D_{J-1}$ , da TWD 1D também poderia ser fracionada, permitindo esta melhor localização escala/frequência desejada. E é exatamente isso que a TWP faz. Ela não faz distinção entre os blocos  $C^j$  e  $D^j$  da TWD 1D, ambos são decompostos da mesma forma. Isso faz com que a cada nível de decomposição, faixas cada vez menores de frequência estejam associadas aos blocos obtidos no nível correspondente. No final, quando cada bloco possuir apenas um elemento cada, a associação entre frequência e escala será a melhor possível.

Quanto à implementação da TWP, cujo algoritmo será omitido, uma das dificuldades está no armazenamento das informações obtidas a cada nível da transformada, pois o número de blocos cresce exponencialmente. A Figura 4.2 ilustra a estrutura de árvore gerada neste processo para o caso de quatro níveis de decomposição. Para se evitar o alocação extra de memória, a cada nível (iteração do algoritmo), os coeficientes calculados devem substituir os coeficientes antigos, utilizando assim o mesmo espaço de memória. Além disso, uma heurística eficiente de busca de dados dentro da árvore é essencial. Algoritmos eficientes usam estruturas de dados específicas para alcançar estes objetivos. O código em Python desenvolvido para a análise de sinais EEG, apresentada no final deste capítulo, aplica noções complexas sobre pilha, tópico da disciplina Estrutura de Dados.

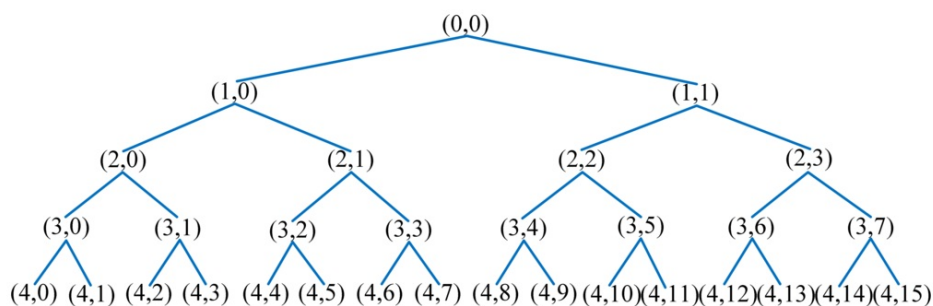


Figura 4.1: Árvore de decomposição para a TWP em quatro níveis.

O interessante na TWP é que agora tem-se a liberdade de se percorrer a árvore de decomposição, Figura 4.2, não só da maneira usual, visitando-se todos os blocos (ou folhas da árvore), mas realizando "passeios" conforme as faixas de frequência que se quer ou necessita enfatizar ou seguindo algum outro critério de seleção de



blocos. Assim nasce a TWP podada, e algoritmos de busca da melhor poda possível para a árvore analisada, conhecidos com o *best-basis algorithm*.

#### 4.1.1 Algoritmo *Best-Basis* e Função de Custo

Apenas para ilustrar a flexibilidade da TWP com relação às diferentes escolhas possíveis de blocos para se representar informações (sinais, imagens, dados em geral), apresenta-se de forma intuitiva, o exemplo proposto no site de *Bear Products International*, [http://www.bearcave.com/bear\\_contents.shtml](http://www.bearcave.com/bear_contents.shtml), especializada em projetos de software de larga escala. Após obtida a árvore com os blocos definidos pela TWP, o algoritmo *best-basis* tem como principal objetivo, selecionar blocos da árvore que produzam a representação mais desejável referente a uma função custo em particular. Esta função custo deve ser modelada de acordo com a aplicação. Por exemplo, se a aplicação for compressão dos dados, a função custo poderia ser o número de *bits* necessários para representar os resultados dos blocos. Em geral, uma função custo  $\mathcal{K}$  produz valores reais e é definida de forma aditiva em relação à concatenação de vetores. A concatenação de dois vetores  $\mathbf{a}$  e  $\mathbf{b}$ , ambos de comprimento finito, é definida pela criação de um novo vetor, denotado por  $[\mathbf{a} \ \mathbf{b}]$ , formado pelos elementos de  $\mathbf{a}$  seguidos pelos elementos de  $\mathbf{b}$ . Assim,  $\mathcal{K}([\mathbf{a} \ \mathbf{b}]) = \mathcal{K}(\mathbf{a}) + \mathcal{K}(\mathbf{b})$ . Além disso,  $\mathcal{K}(\mathbf{0}) = 0$ , sendo  $\mathbf{0}$  o vetor nulo.

Um exemplo interessante é a função de custo de truncamento  $\mathcal{K}_T$ , que conta o número de elementos de cada bloco da TWP, cujos valores são maiores do que um certo valor de *threshold*  $\lambda$ . Escolhendo-se  $\lambda = 1$  e a TWP de Haar de um vetor inicial, dada na Figura 4.2, apresentam-se os valores da função de custo de truncamento  $\mathcal{K}_T$  para cada bloco da árvore correspondente na Figura 4.3.

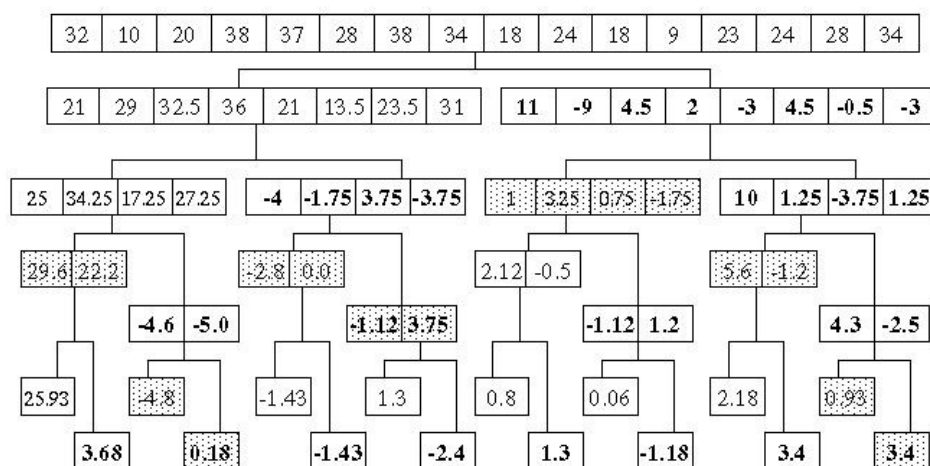


Figura 4.2: Exemplo de árvore, gerada pela TWP de Haar não-normalizada com 4 níveis de decomposição



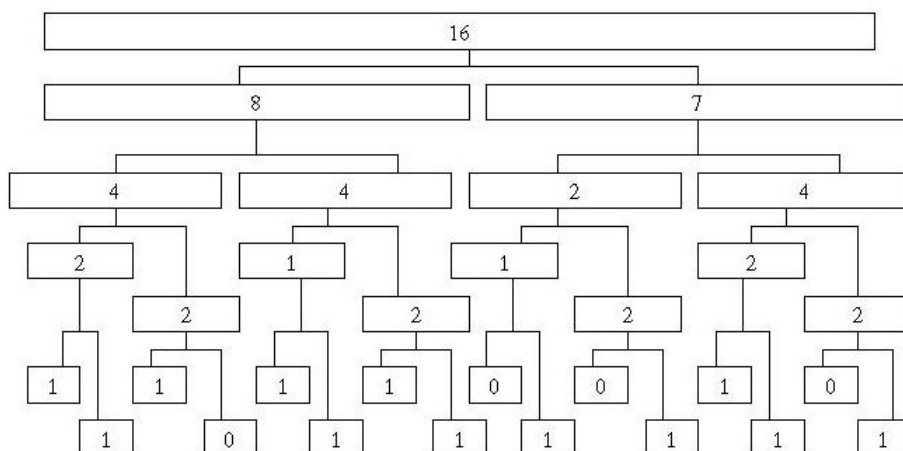


Figura 4.3: Árvore gerada pela TWP de Haar mostrando os resultados da função de custo de truncamento,  $\mathcal{K}_T$ , por bloco.

Uma vez calculados os custos de cada bloco da TWP (de Haar neste exemplo), o algoritmo *best-basis* irá procurar quais os melhores blocos dentro da árvore gerada pela TWP a serem considerados para representar os dados originais da melhor forma possível, em relação à função custo em questão. Isso implica, que após esta busca, uma representação com uma quantidade menor de blocos será obtida. Esta busca é executada via estratégia *bottom-up*, ou seja, de baixo para cima, de forma recursiva, através das seguintes regras:

- uma folha é bloco da TWP que não possui filhos, ou seja, não possui mais blocos associados em níveis ainda menores. Quando visitada, uma folha retorna sempre seu valor da função de custo. (Neste exemplo, a  $\mathcal{K}_T$ );
- conforme a busca recursiva avança na direção do bloco principal associado ao vetor inicial, a raiz da árvore, o valor da função custo de um bloco,  $v_1$  é comparado com o custo de seus filhos  $v_2$ , calculado pela propriedade aditiva em relação à concatenação. Assim:
  - Se  $(v_1 \leq v_2)$ , então o bloco é marcado como sendo parte do conjunto de blocos que irão formar a melhor base de representação;
  - Se  $(v_1 > v_2)$ , então o valor da função de custo associado ao bloco é substituído por  $v_2$

Os blocos, ou nós, sombreados na Figura 4.4 realçam aqueles que foram selecionados pela heurística de busca do algoritmo *best-basis*, dependendo dos custos da função  $\mathcal{K}_T$  para os dados apresentados na Figura 4.2. O conjunto de saída do algoritmo *best-basis*, também denominado *best-basis set* (BBS), para este exemplo é



funções de custo, uma vez que tanto as frequências de interesse quanto as faixas de frequência dos blocos são conhecidas *a priori*.

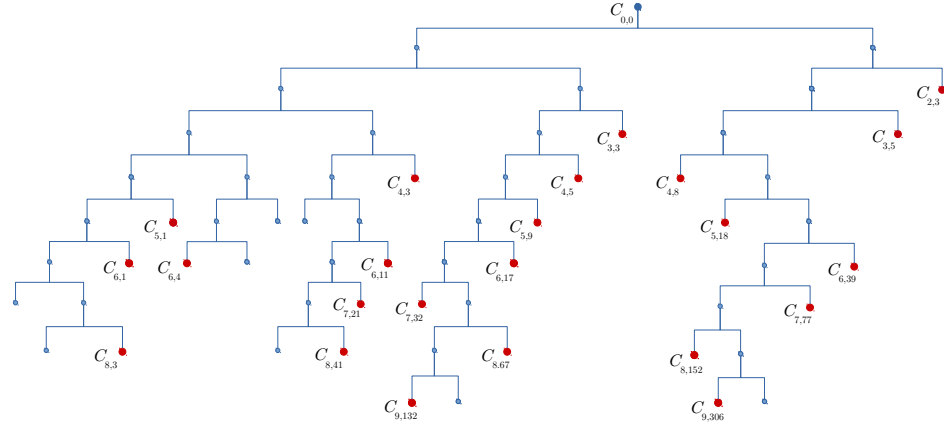


Figura 4.5: Coeficientes da TWP de Haar podada associados a ritmos cerebrais [28]: Baixo- $\gamma$  associado aos blocos ( $C_{2,3}$ ,  $C_{3,5}$ ,  $C_{6,39}$  e  $C_{7,77}$ ).  $\beta$  associado a ( $C_{3,3}$ ,  $C_{4,5}$ ,  $C_{4,8}$ ,  $C_{5,9}$ ,  $C_{5,18}$ ,  $C_{6,17}$ ,  $C_{8,67}$ ,  $C_{8,152}$  e  $C_{9,306}$ ).  $\alpha$  associado a ( $C_{4,3}$ ,  $C_{4,11}$ ,  $C_{7,21}$ ,  $C_{7,32}$ ,  $C_{8,41}$  e  $C_{9,132}$ ). E o ritmo  $\delta$  associado a ( $C_{5,1}$ ,  $C_{6,1}$ ,  $C_{6,4}$  e  $C_{8,3}$ ). O sinal de entrada é um EEG, representado por  $C_{0,0}$ , com 512 pontos, amostrado a 100Hz.

A análise visual dos coeficientes *wavelets* associados aos ritmos  $\delta$  e baixo- $\gamma$  permite que sejam identificados dois comportamentos completamente diferentes, como é evidenciado na Figura 4.6. Uma observação importante é que esta diferenciação entre estes ritmos não é observada através do sinal original. Assim, a partir dessa análise e de conhecimentos específicos sobre os ritmos cerebrais  $\alpha$ ,  $\beta$ , baixo- $\gamma$  e  $\delta$  são propostos dois índices calculados a partir desses coeficientes dos blocos selecionados da TWP de Haar, dados por:

$$\text{Índice (i)} = \frac{\gamma}{\delta} \quad (4.1)$$

e

$$\text{Índice (ii)} = \frac{(\gamma + \beta)}{(\delta + \alpha)} \quad (4.2)$$

Estes índices fazem parte do procedimento de reconhecimento de padrões de sonolência, proposto em [28]. Através de testes estatísticos é possível evidenciar a significância das quantidades definidas em 4.1 e 4.2 na distinção de eventos associados à sonolência.

Aqui, a contribuição da TWP de Haar é de permitir a extração de informações muito relevantes de um sinal inicial, inicialmente imperceptíveis, de tal modo que estas ainda podem ser reduzidas a ponto de sintetizarem características associadas

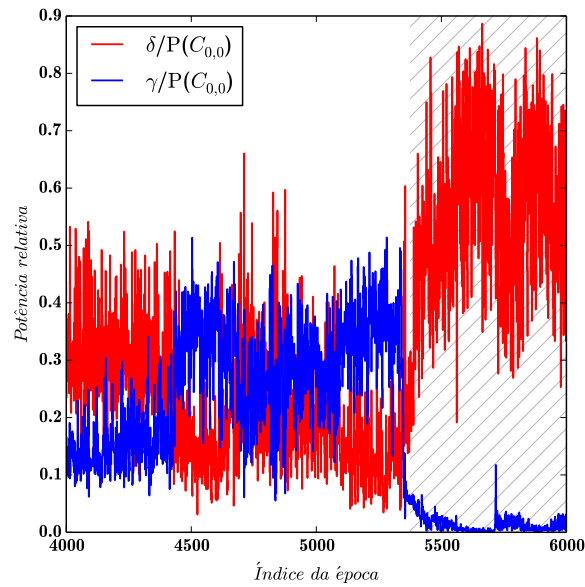


Figura 4.6: Potências relativas dos coeficientes *wavelet* packet referentes aos ritmos baixo-gama e delta [28]. Experimento realizado com o EEG do canal Pz-Oz do paciente SC4181 do banco de dados *Sleep-EDF [Expanded]*. Parte hachurada indica quando o sujeito foi classificado como sonolento pelas anotações do banco de dados *Physionet*.

a eventos específicos. Este exemplo ilustra ainda o potencial das TWD e TWP em aplicações que necessitam de redução de dimensionalidade, uma vez que os dados brutos, por serem de dimensão muito grande, não permitem a distinção imediata de padrões.

Muitos aspectos teóricos sobre as *wavelets* foram omitidos neste texto, mas através dos exemplos apresentados, dos algoritmos e das referências complementares fornecidas, acredita-se ter oferecido um passeio geral pelos tópicos mais explorados na literatura especializada nas últimas décadas.



## Agradecimentos

O primeiro autor agradece à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). O segundo autor agradece à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) nº 1873-25.51/13-0 e ambos agradecem ao Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal de Santa Maria (UFSM). Além disso, um agradecimento especial ao comitê organizador do IV Colóquio de Matemática da Região Sul-SBM pela oportunidade de divulgar este trabalho.



## Referências Bibliográficas

- [1] M. ASADZAEDH, E. HASHEMI, AND A. KOZAKEVICIUS. Efficiency of combined daubechies and statistical parameters applied in mammography. *Applied and Computational Mathematics: An International Journal*, 12.
- [2] F. M. BAYER AND A. J. KOZAKEVICIUS. SPC-threshold: Uma proposta de limiarização para filtragem adaptativa de sinais. *Trends in Applied and Computational Mathematics*, 12(2):112–132, 2010.
- [3] JOSÉ LUIZ BOLDRINI, SUELI I. RODRIGUES COSTA, VERA LÚCIA FIGUEIREDO, AND HENRY G. WETZLER. *Álgebra Linear*. Harbra, 3 edition, 1986.
- [4] W. L. BRIGGS AND V. E. HENSON. *The DFT: An Owners' Manual for the Discrete Fourier Transform*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1995.
- [5] A. BRUNTON, T. BOLKART, AND S. WUHRER. Multilinear wavelets: A statistical shape space for human faces. *Lecture Notes in Computer Science*, page 297–312, 2014.
- [6] R. BURGER AND A. KOZAKEVICIUS. Adaptative multiresolution weno schemes for multi-species kinematic flow models. *Journal of Computational Physics*, 224:1190–1222, 2007.
- [7] LIU CHUN-LIN. A tutorial of the wavelet transform, 2010.
- [8] RONALD R. COIFMAN, YVES MEYER, AND VICTOR WICKERHAUSER. Wavelet analysis and signal processing. In *IN WAVELETS AND THEIR APPLICATIONS*, pages 153–178, 1992.
- [9] INGRID DAUBECHIES. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
- [10] DAVID L. DONOHO AND IAIN M. JOHNSTONE. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, pages 1200–1224, 1995.



- [11] MARIE FARGE AND KAI SCHNEIDER. Wavelet transforms and their applications to mhd and plasma turbulence: a review. *Journal of Plasma Physics*, 81, 12 2015.
- [12] M. W. FRAIZER. *An introduction to wavelets through linear algebra*. Springer, 1999.
- [13] A. L. GOLDBERGER, L. A. N. AMARAL, L. GLASS, J. M. HAUSDORFF, P. CH. IVANOV, R. G. MARK, J. E. MIETUS, G. B. MOODY, C.-K. PENG, AND H. E. STANLEY. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.
- [14] RAFAEL C. GONZALEZ AND RICHARD E. WOODS. *Digital Image Processing*. Prentice Hall, 3 edition, 2007.
- [15] U. GRENANDER. The Nyquist frequency is that frequency whose period is two sampling intervals. In *Probability and statistics: the Harald Cramér volume*. Almqvist & Wiksell, 1959.
- [16] ALFRÉD HAAR. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 1910.
- [17] QUAN HUYNH-THU AND MOHAMMED GHANBARI. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.
- [18] HAMID KHORRAMI AND MAJID MOAVENIAN. A comparative study of dwt, {CWT} and {DCT} transformations in {ECG} arrhythmias classification. *Expert Systems with Applications*, 37(8):5751 – 5757, 2010.
- [19] A. KOZAKEVICIUS, C. CAPPO, B. A. MOZZAQUATRO, R. C. NUNES, AND C. E. SCHAEERER. URL query string anomaly sensor designed with the bi-dimensional Haar wavelet transform. 14(6):561–581, 2015.
- [20] ALICE KOZAKEVICIUS AND ALEX A. SCHMIDT. Wavelet transform with special boundary treatment for 1D data. *Computational and Applied Mathematics*, 32(3):447–457, 2013.
- [21] ALICE KOZAKEVICIUS E FÁBIO BAYER. Filtragem de sinais via limitização de coeficientes wavelet. *Ciência e Natura*, 36:37–51, 2014.
- [22] STÉPHANE MALLAT. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3 edition, 2008.
- [23] OLE MØLLER NIELSEN. *Wavelets in Scientific Computing*. PhD thesis, Technical University of Denmark, 1998.

- [24] JUAN R. ROMERO. *Generalized Multiresolution Analysis: Construction and Measure Theoretic Characterization*. PhD thesis, University of Maryland, 2005.
- [25] MARINA RONZHINA, OTO JANOUSEK, JANA KOLAROVA, MARIE NOVAKOVA, PETR HONZIK, AND IVO PROVAZNIK. Sleep scoring using artificial neural networks. *Sleep Medicine Reviews*, 16:251–263, 2012.
- [26] M. SHENSA. The discrete wavelet transform: wedding the a trous and Mallat algorithms. *Signal Processing, IEEE Transactions on*, 40(10):2464–2482, Oct 1992.
- [27] THIAGO SILVEIRA. Classificação de estágios de sono através da aplicação de transformada wavelet discreta sobre um único canal de eletroencefalograma. Dissertação de Mestrado, Universidade Federal de Santa Maria, 2016.
- [28] THIAGO SILVEIRA, ALICE DE JESUS KOZAKEVICIUS, AND CESAR RAMOS RODRIGUES. Automated drowsiness detection through wavelet packet analysis of a single EEG channel. *Expert Systems With Applications*, (55):559–565, 2016.
- [29] THIAGO SILVEIRA, ALICE DE JESUS KOZAKEVICIUS, AND CESAR RAMOS RODRIGUES. Single-channel EEG sleep stage classification based on a streamlined set of statistical features in wavelet domain. *Medical & Biological Engineering & Computing*, (Online first):1–10, 2016.
- [30] THIAGO SILVEIRA, ALICE JESUS KOZAKEVICIUS, AND CESAR RAMOS RODRIGUES. Awake/sleep scoring through wavelet analysis associated to decision tree algorithms. In *Anais do XXX Simpósio Sul de Microeletrônica*, 2015.
- [31] ERIC J. STOLLNITZ, TONY D. DEROSE, AND DAVID H. SALESIN. Wavelets for computer graphics: A primer - part 1. *IEEE Computer Graphics and Applications*, 15:76–84, 1995.
- [32] S. TAPANI AND A. KOZAKEVICIUS. Wavelet change-point analysis for non-stationary time series. *Journal of Wavelet Theory and Applications*, 5(1):29–44, 2012.
- [33] BRANI VIDA KOVIC AND PETER MUELLER. Wavelets for kids: A tutorial introduction. Technical report, Duke University, 1991.

## COLEÇÃO DO PROFESSOR DE MATEMÁTICA

- *Logaritmos* - E. L. Lima
- *Análise Combinatória e Probabilidade com as soluções dos exercícios* - A. C. Morgado, J. B. Pitombeira, P. C. P. Carvalho e P. Fernandez
- *Medida e Forma em Geometria (Comprimento, Área, Volume e Semelhança)* - E. L. Lima
- *Meu Professor de Matemática e outras Histórias* - E. L. Lima
- *Coordenadas no Plano as soluções dos exercícios* - E. L. Lima com a colaboração de P. C. P. Carvalho
- *Trigonometria, Números Complexos* - M. P. do Carmo, A. C. Morgado e E. Wagner, Notas Históricas de J. B. Pitombeira
- *Coordenadas no Espaço* - E. L. Lima
- *Progressões e Matemática Financeira* - A. C. Morgado, E. Wagner e S. C. Zani
- *Construções Geométricas* - E. Wagner com a colaboração de J. P. Q. Carneiro
- *Introdução à Geometria Espacial* - P. C. P. Carvalho
- *Geometria Euclidiana Plana* - J. L. M. Barbosa
- *Isometrias* - E. L. Lima
- *A Matemática do Ensino Médio Vol. 1* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *A Matemática do Ensino Médio Vol. 2* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *A Matemática do Ensino Médio Vol. 3* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *Matemática e Ensino* - E. L. Lima
- *Temas e Problemas* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *Episódios da História Antiga da Matemática* - A. Aaboe
- *Exame de Textos: Análise de livros de Matemática* - E. L. Lima
- *A Matemática do Ensino Médio Vol. 4 - Exercícios e Soluções* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *Construções Geométricas: Exercícios e Soluções* - S. Lima Netto
- *Um Convite à Matemática* - D.C de Morais Filho
- *Tópicos de Matemática Elementar* - Volume 1 - Números Reais - A. Caminha
- *Tópicos de Matemática Elementar* - Volume 2 - Geometria Euclidiana Plana - A. Caminha
- *Tópicos de Matemática Elementar* - Volume 3 - Introdução à Análise - A. Caminha
- *Tópicos de Matemática Elementar* - Volume 4 - Combinatória - A. Caminha
- *Tópicos de Matemática Elementar* - Volume 5 - Teoria dos Números - A. Caminha
- *Tópicos de Matemática Elementar* - Volume 6 - Polinômios - A. Caminha
- *Treze Viagens pelo Mundo da Matemática* - C. Correia de Sa e J. Rocha (editores)
- *Como Resolver Problemas Matemáticos* - T. Tao
- *Geometria em Sala de Aula* - A. C. P. Hellmeister (Comitê Editorial da RPM)
- *Números Primos, amigos que causam problemas* - P. Ribenboim
- *Manual de Redação Matemática* - D.C de Morais Filho

### COLEÇÃO PROFMAT

- *Introdução à Álgebra Linear* - A. Hefez e C.S. Fernandez
- *Tópicos de Teoria dos Números* - C. G. Moreira , F. E Brochero e N. C. Saldanha
- *Polinômios e Equações Algébricas* - A. Hefez e M.L. Villela
- *Tópicos de Historia de Matemática* - T. Roque e J. Bosco Pitombeira
- *Recursos Computacionais no Ensino de Matemática* - V. Giraldo, P. Caetano e F. Mattos
- *Temas e Problemas Elementares* - E. L. Lima, P. C. P. Carvalho, E. Wagner e A. C. Morgado
- *Números e Funções Reais* - E. L. Lima
- *Aritmética* - A. Hefez
- *Geometria* - A. Caminha
- *Avaliação Educacional* - M. Rabelo
- *Geometria Analítica* - J. Delgado, K. Frensel e L. Crissaff
- *Matemática Discreta* - A. Morgado e P. C. P. Carvalho
- *Matemática e Atualidade - Volume 1* - C. Rousseau e Y. Saint-Aubin
- *Fundamentos de Cálculo* - A. C. Muniz Neto
- *Matemática e Atualidade - Volume 2* - C. Rousseau e Y. Saint-Aubin
- *Exercícios Resolvidos de Álgebra Linear* - A. Hefez e C. de Souza Fernandez

### COLEÇÃO INICIAÇÃO CIENTÍFICA

- *Números Irracionais e Transcendentes* - D. G. de Figueiredo
- *Números Racionais e Irracionais* - I. Niven
- *Tópicos Especiais em Álgebra* - J. F. S. Andrade

### COLEÇÃO TEXTOS UNIVERSITÁRIOS

- *Introdução à Computação Algébrica com o Maple* - L. N. de Andrade
- *Elementos de Aritmética* - A. Hefez
- *Métodos Matemáticos para a Engenharia* - E. C. de Oliveira e M. Tygel
- *Geometria Diferencial de Curvas e Superfícies* - M. P. do Carmo
- *Matemática Discreta* - L. Lovász, J. Pelikán e K. Vesztergombi
- *Álgebra Linear: Um segundo Curso* - H. P. Bueno
- *Introdução às Funções de uma Variável Complexa* - C. S. Fernandez e N. C. Bernardes Jr.
- *Elementos de Topologia Geral* - E. L. Lima
- *A Construção dos Números* - J. Ferreira
- *Introdução à Geometria Projetiva* - A. Barros e P. Andrade
- *Análise Vetorial Clássica* - F. Acker
- *Funções, Limites e Continuidade* - P. Ribenboim
- *Fundamentos de Análise Funcional* - G. Botelho, D. Pellegrino e E. Teixeira
- *Teoria dos Números Transcendentes* - D. Marques
- *Introdução à Geometria Hiperbólica - O modelo de Poincaré* - P. Andrade
- *Álgebra Linear: Teoria e Aplicações* - T. P. de Araújo
- *Introdução à Análise Matemática na Reta* - C. I. Doering
- *Topologia e Análise no Espaço  $R^n$*  - R. Freire de Lima

- *Equações Ordinárias e Aplicações* - B. Scárdua

#### **COLEÇÃO MATEMÁTICA APLICADA**

- *Introdução à Inferência Estatística* - H. Bolfarine e M. Sandoval
- *Discretização de Equações Diferenciais Parciais* - J. Cuminato e M. Meneguette
- *Fenômenos de Transferência – com Aplicações às Ciências Físicas e à Engenharia volume 1: Fundamentos* - J. Pontes e N. Mangiavacchi

#### **COLEÇÃO OLIMPIADAS DE MATEMÁTICA**

- *Olimpíadas Brasileiras de Matemática, 1ª a 8ª* - E. Mega e R. Watanabe
- *Olimpíadas Brasileiras de Matemática, 9ª a 16ª* - C. Moreira e E. Motta, E. Tengan, L. Amâncio, N. C. Saldanha e P. Rodrigues
- *21 Aulas de Matemática Olímpica* - C. Y. Sh
- *Iniciação à Matemática: Um Curso com Problemas e Soluções* - K. I. M. Oliveira e A. J. C. Fernández
- *Olimpíadas Cearenses de Matemática 1981-2005 Nível Fundamental* - E. Carneiro, O. Campos e M. Paiva
- *Olimpíadas Cearenses de Matemática 1981-2005 Nível Médio* - E. Carneiro, O. Campos e M. Paiva
- *Olimpíadas Brasileiras de Matemática - 17ª a 24ª* - C. G. T. de A. Moreira, C. Y. Shine, E. L. R. Motta, E. Tengan e N. C. Saldanha

#### **COLEÇÃO FRONTEIRAS DA MATEMÁTICA**

- *Fundamentos da Teoria Ergódica* - M. Viana e K. Oliveira
- *Tópicos de Geometria Diferencial* - A. C. Muniz Neto
- *Formas Diferenciais e Aplicações* - M. Perdigão do Carmo

#### **COLEÇÃO MATEMÁTICA PARA O ENSINO**

- *Livro do Professor de Matemática na Educação Básica Volume I Números Naturais* - C. Ripoll, L. Rangel e V. Giraldo
- *Livro do Professor de Matemática na Educação Básica Volume II Números Inteiros* - C. Ripoll, L. Rangel e V. Giraldo

