



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Cristiano Marcos Agulhari
Engenheiro de Computação – FEEC - Unicamp

COMPRESSÃO DE ELETROCARDIOGRAMAS USANDO WAVELETS

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telemática e Telecomunicações.

Banca Examinadora:

Prof. Dr. Ivanil Sebastião Bonatti (Orientador)
FEEC - Unicamp

Prof. Dr. Pedro Luis Dias Peres
FEEC - Unicamp

Prof. Dr. Rodrigo Capobianco Guido
IFSC - USP São Carlos

Campinas
Fevereiro 2009

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

Ag95c Agulhari, Cristiano Marcos
 Compressão de eletrocardiogramas usando wavelets /
 Cristiano Marcos Agulhari. --Campinas, SP: [s.n.], 2009.

 Orientador: Ivanil Sebastião Bonatti.
 Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

 1. Compressão de dados (Computação). 2.
Eletrocardiografia. 3. Wavelet (Matemática). I. Bonatti,
Ivanil Sebastião. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Título em Inglês: Compression of electrocardiograms using wavelets

Palavras-chave em Inglês: Data compression, Electrocardiograms, Wavelet

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Rodrigo Capobianco Guido, Pedro Luís Dias Peres

Data da defesa: 11/02/2009

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Cristiano Marcos Agulhari

Data da Defesa: 11 de fevereiro de 2009

Título da Tese: "Compressão de Eletrocardiograma Usando Wavelets"

Prof. Dr. Ivanil Sebastião Bonatti (Presidente):



Prof. Dr. Rodrigo Capobianco Guido:



Prof. Dr. Pedro Luís Dias Peres:



Agradecimentos

Agradeço,

aos meus pais, Sueli e José, por serem os principais responsáveis por eu ter chegado até aqui, por sua constante presença e seu amor incondicional.

ao meu irmão Ricardo, mais do que um simples irmão de sangue, um amigo e confidente. E à Maria Angélica, minha cunhada preferida.

à Sandra, minha tia e *irmã*, sempre um grande exemplo nos mais diversos aspectos.

ao meu avô, Daniel, que nos deixa muitas saudades; e à minha avó, Felícia, que a cada dia nos ensina o significado de força, determinação e amor.

ao Prof. Ivanil, pelos quatro anos de valorosa orientação, pelos seus grandes ensinamentos nas mais variadas áreas e pelas centenas de cafés pagos ao longo desses anos.

ao Prof. Pedro, por sempre acreditar no meu potencial e por sua imensa ajuda desde os tempos de minha graduação.

aos professores da FEEC: Renato Lopes, Meloni, Von Zuben, Romis, Anésio e Takaaki, pelos ótimos cursos oferecidos.

aos membros da banca examinadora, pelos comentários, sugestões e contribuições que ajudaram a melhorar a qualidade e a redação final do manuscrito.

à agência FAPESP, pelo apoio financeiro concedido durante todo o período do mestrado.

ao Google, IEEEExplore e Wikipedia, pelo eficiente acesso ao conhecimento científico.

aos amigos do curso de graduação, por todos os *Nabus Iminentus* oferecidos e por me ensinarem o real valor do companheirismo. Em especial ao Anselmo, um amigo sempre presente e que me inspirou a começar a carreira acadêmica, e ao Thiago, por dividir quartos, amarguras e alegrias.

aos amigos da pós-graduação, por todo o apoio oferecido durante os cursos e as pesquisas. Em especial à Rosanna, que caminhou ao meu lado durante um bom período do meu mestrado.

aos demais amigos que sempre contribuíram de alguma forma para meu crescimento pessoal e profissional. Em especial ao Paulo, que acompanha minha trajetória desde antes de nos mudarmos para Campinas; ao Rodrigo, uma grande amizade de longa data, presente em momentos de alegrias e preocupações; à Laura, uma pessoa maravilhoa e de extrema bondade; ao Renato, meu *compadre*, pelas noites de filmes e pelas palavras certas nas horas incertas; e ao Bacalhau, pela sua incrível amizade fraterna, pela sua confiança, força e excentricidade.

e a Deus, por tudo.

Resumo

A principal contribuição desta dissertação é a proposta de dois métodos de compressão de eletrocardiogramas (ECGs). O primeiro método, chamado *Run Length Encoding Adaptativo* (RLEA), é baseado nas transformadas *wavelet* e consiste basicamente em utilizar uma função *wavelet*, obtida pela resolução de um problema de otimização, que se ajuste ao sinal a ser comprimido. O problema de otimização torna-se irrestrito com a parametrização dos coeficientes do filtro escala, que definem unicamente uma função *wavelet*. Após a resolução do problema de otimização é aplicado o procedimento de decomposição *wavelet* no sinal e os coeficientes de representação mais significativos são retidos, sendo que o número de coeficientes retidos é determinado de forma a satisfazer uma medida de distorção pré-especificada. Os coeficientes retidos são então quantizados e compactados, assim como o *bitmap* que indica as posições dos coeficientes retidos. A quantização é feita de forma adaptativa, utilizando diferentes números de *bits* de quantização para os diferentes subespaços de decomposição considerados. Tanto os valores dos coeficientes retidos quanto o *bitmap* são codificados utilizando uma variante do método *Run Length Encoding*. O segundo método proposto nesta dissertação, chamado *Zero Padding Singular Values Decomposition* (ZPSVD), consiste em primeiramente detectar os batimentos, equalizá-los pela inserção de zeros (*zero padding*) e então aplicar a decomposição SVD para obter tanto a base quanto os coeficientes de representação dos batimentos. Alguns componentes da base são retidos e então comprimidos utilizando os mesmos procedimentos aplicados aos coeficientes de decomposição do ECG no método RLEA, enquanto que os coeficientes de projeção dos batimentos nessa base são quantizados utilizando um procedimento de quantização adaptativa. Os dois métodos de compressão propostos são comparados com diversos outros métodos existentes na literatura por meio de experimentos numéricos.

Palavras-chave: compressão de dados, eletrocardiogramas, *wavelet*, *singular values decomposition*, *run length encoding*.

Abstract

The main contribution of the present thesis is the proposition of two electrocardiogram (ECG) compression methods. The first method, called *Run Length Encoding Adaptativo* (RLEA), is based on wavelet transforms and consists of using a wavelet function, obtained by the resolution of an optimization problem, which fits to the signal to be compressed. The optimization problem becomes unconstrained with the parametrization of the coefficients of the scaling filter, that define uniquely a wavelet function. After the resolution of the optimization problem, the wavelet decomposition procedure is applied to the signal and the most significant coefficients of representation are retained, being the number of retained coefficients determined in order to satisfy a pre-specified distortion measure. The retained coefficients are quantized and compressed, likewise the bitmap that informs the positions of the retained coefficients. The quantization is performed in an adaptive way, using different numbers of bits for the different decomposition subspaces considered. Both the values of the retained coefficients and the bitmap are encoded using a modified version of the Run Length Encoding technique. The second method proposed in this dissertation, called *Zero Padding Singular Values Decomposition* (ZPSVD), consists of detecting the beat pulses of the ECG, equalizing the pulses by inserting zeros (zero padding), and finally applying the SVD to obtain both the basis and the coefficients of representation of the beat pulses. Some components of the basis are retained and then compressed using the same procedures applied to the coefficients of decomposition of the ECG in the RLEA method, while the coefficients of projection of the beat pulses in the basis are quantized using an adaptive quantization procedure. Both proposed compression methods are compared to other techniques by means of numerical experiments.

Key-words: data compression, electrocardiograms, wavelet, singular values decomposition, run length encoding.

Siglas

CR	<i>Compression Ratio</i> (Taxa de Compressão)
DC	<i>Direct Current</i> (corrente contínua). Valor médio do sinal
DCT	<i>Discrete Cosine Transform</i>
ECG	Eletrocardiograma
FPA	Filtro Passa Altas
FPB	Filtro Passa Baixas
FIR	<i>Finite Impulse Response</i>
LZW	Lempel-Ziv-Welch
MAPRD	<i>Moving Average Percent Root-mean-square Distortion</i>
PRD	<i>Percent Root-mean-square Distortion</i>
QMF	<i>Quadrature Mirror Filter</i>
RLE	<i>Run Length Encoding</i>
RLEA	<i>Run Length Encoding</i> Adaptativo
RMSE	<i>Root Mean Square Error</i>
SPIHT	<i>Set Partitioning in Hierarchical Trees</i>
SQP	<i>Sequential Quadratic Programming</i>
SVD	<i>Singular Values Decomposition</i>
TRE	<i>Two-Role Encoder</i>
UPRD	PRD definido pelo usuário
WDD	<i>Weighted Diagnostic Distortion</i>
ZPSVD	<i>Zero Padding Singular Values Decomposition</i>

Notação

\mathbb{R}	conjunto dos números reais
\mathbb{Z}	conjunto dos números inteiros
$\mathbb{R}^{m \times n}$	conjunto de matrizes reais de dimensão $m \times n$
L^1	conjunto das funções reais absolutamente integráveis
L^2	conjunto das funções reais quadraticamente integráveis
A	notação para matrizes (letras maiúsculas)
I_n	matriz identidade de dimensão $n \times n$
A'	transposição da matriz A
\hat{A}	aproximação de menor <i>rank</i> da matriz A
r	<i>rank</i> de uma matriz
λ_k	k -ésimo auto-valor de uma matriz
u_k	notação para o k -ésimo auto-vetor de uma matriz U
σ_k	k -ésimo valor singular de uma matriz
$\ A\ $	norma de Frobenius da matriz A , dada por $\ A\ = \sqrt{\sum_i \sum_j a_{ij} ^2}$, sendo a_{ij} um elemento de A .

$F(\omega)$	transformada discreta de Fourier da função $f[n]$
$F(z)$	transformada Z da função $f[n]$
$\langle \cdot, \cdot \rangle$	produto interno de duas funções, dado por $\langle f(x), g(x) \rangle = \int_{-\infty}^{+\infty} f(x)g^*(x)dx$
*	operador convolução discreta, dada por $f[n] * g[n] = \sum_{k=-\infty}^{+\infty} f[k]g[n-k]$
$\{\cdot\}_{\downarrow 2}$	operador de subamostragem
$\{\cdot\}_{\uparrow 2}$	operador de superamostragem
\bar{f}	valor médio (valor DC) do sinal $f[n]$
$\delta[n]$	função delta de Kronecker
$\delta(x)$	função delta de Dirac
$y[n]$	saída de um filtro digital
$x[n]$	entrada de um filtro digital
$f[n]$	ECG original
$\hat{f}[n]$	ECG reconstruído
τ_n	número de amostras no ECG
τ_c	número de coeficientes de decomposição
F	matriz em que cada linha é um batimento do ECG $f[n]$
\mathcal{E}	energia do ECG original
$\hat{\mathcal{E}}$	energia do ECG reconstruído
β	número médio de amostras de um batimento de um ECG
ν	número de batimentos em um ECG
$\varphi(x)$	função escala
$\psi(x)$	função <i>wavelet</i>
$\ell[n]$	coeficientes do filtro escala
$h[n]$	coeficientes do filtro <i>wavelet</i>
η	resolução dos subespaços escala e <i>wavelet</i>
m	tamanho das funções escala e <i>wavelet</i> e número de coeficientes dos filtros escala e <i>wavelet</i>
V_η	subespaço escala de resolução η
W_η	subespaço <i>wavelet</i> de resolução η
$a_\eta[n]$	coeficientes de projeção de um sinal no subespaço escala de resolução η
$b_\eta[n]$	coeficientes de projeção de um sinal no subespaço <i>wavelet</i> de resolução η
$\mu[n]$	momento de ordem n de um par de funções escala e <i>wavelet</i>
θ	vetor de parâmetros que descreve um par de funções escala e <i>wavelet</i> ortogonais
γ	número de parâmetros do vetor θ
m_k	número de coeficientes de decomposição no k -ésimo subespaço
κ	número de amostras das funções escala e <i>wavelet</i>
$v_k[n]$	k -ésimo elemento de uma base de funções discretas
c_k	coeficiente de projeção de uma função no k -ésimo elemento de uma base
\mathcal{K}	conjunto de índices de coeficientes de transformação retidos
ρ	taxa de retenção de coeficientes de transformação
ζ	limiar que separa os coeficientes de transformação significativos dos não-significativos
ϵ	tolerância correspondente à distorção obtida em relação ao UPRD
ϵ_q	tolerância correspondente à distorção resultante da quantização em relação ao UPRD

$q[k]$	número de <i>bits</i> de quantização dos coeficientes do k -ésimo subespaço
q_z	número de <i>bits</i> usados para codificar os tamanhos das sequências de zeros de um <i>bitmap</i>
q_m	número máximo de <i>bits</i> de quantização dos coeficientes de projeção de um ECG em uma base
s	número de componentes singulares retidos
w	janela de observação considerada no cálculo do MAPRD
α_k	k -ésimo símbolo de uma sequência a ser compactada
$p(\alpha_k)$	frequência relativa de ocorrência do símbolo α_k
$s(\alpha_k)$	tamanho do código relacionado ao símbolo α_k

Sumário

Introdução	1
1 Compactação	5
1.1 Introdução	5
1.2 Método de Huffman	6
1.3 Lempel-Ziv-Welch	7
1.4 Run Length Encoding	9
1.5 Comparações entre os métodos	10
1.6 Conclusão	12
2 Eletrocardiogramas	15
2.1 Introdução	15
2.2 Características dos eletrocardiogramas	16
2.2.1 Base de dados	17
2.2.2 <i>Baseline</i>	17
2.3 Detecção de batimentos	17
2.4 Medidas de desempenho	19
2.5 Conclusão	22
3 Wavelets	23
3.1 Introdução	23
3.2 Funções escala e <i>Wavelets</i>	24
3.2.1 Funções escala	24
3.2.2 <i>Wavelets</i>	26
3.2.3 Coeficientes de projeção e bancos de filtros	28
3.3 Parametrização	30
3.4 Adaptação ao sinal	32
3.5 Conclusão	35
4 Compressão por Adaptação da <i>Wavelet</i>	37
4.1 Introdução	37
4.2 Método de compressão RLEA	38
4.3 Experimentos numéricos	40
4.4 Conclusão	45

5	Compressão por Decomposição em Valores Singulares	47
5.1	Introdução	47
5.2	Decomposição em Valores Singulares	47
5.3	Método de compressão ZPSVD	49
5.4	Experimentos numéricos	50
5.5	Comparação entre ZPSVD e RLEA	53
5.6	Conclusão	55
	Conclusões	57
	Bibliografia	59

Introdução

Os eletrocardiogramas (ECGs) são sinais clínicos capazes de fornecer aos cardiologistas informações essenciais sobre o diagnóstico de cardiopatias em um paciente [Bra97]. Em um sistema de monitoração ambulatorial, o volume de dados de ECGs obtido é grande, pois monitora-se o paciente por longos períodos de tempo [ZLP00]. O exame de monitoramento de Holter, por exemplo, tipicamente consiste na aquisição dos sinais cardíacos do paciente durante 24 horas. Com o crescente avanço da telemedicina [CKS⁺06], [GFAG08], [Kom05], os eletrocardiogramas também podem ser obtidos remotamente e transmitidos para o destino desejado. Assim, um esquema eficiente de compressão é um componente chave para sistemas tanto de armazenamento quanto de transmissão de eletrocardiogramas. Métodos de compressão de eletrocardiogramas cada vez mais eficazes têm sido desenvolvidos devido tanto à sua importância nos sistemas de armazenamento e transmissão de dados quanto à disponibilidade de bancos de dados de sinais fisiológicos, como os oferecidos pelo PhysioBank [Phy], [GAG⁺00].

Existem dois tipos de compressão: a compressão sem perdas ou compactação e a compressão com perdas ou simplesmente compressão. Os métodos de compactação são utilizados nos casos em que os dados após a descompactação devem ser idênticos aos dados originais, como por exemplo nos arquivos de texto. É possível também aplicar um método de compactação em conjunto com um método de compressão a fim de diminuir o tamanho do arquivo comprimido sem aumentar a distorção dos dados descomprimidos. Entre os métodos mais conhecidos de compactação estão o método de Huffman [Huf52], o método de Lempel-Ziv-Welch [ZL77], [Wel84] e o método *Run Length Encoding* [Gol66].

Os métodos de compressão são utilizados nos casos em que os dados após a descompressão não precisam ser idênticos aos dados originais, sendo possível tolerar pequenas distorções. Nesses casos, o usuário deve poder controlar o compromisso entre alta qualidade e maior compressão alterando parâmetros do algoritmo. Os métodos de compressão com perdas podem ser divididos em duas categorias: os métodos diretos e os métodos de transformadas. Os métodos diretos consistem na análise das amostras do sinal a fim de eliminar redundâncias [JHSC90], [CNFO68], [AT82] e [ISHS83]. Nos métodos de transformadas, o sinal é convertido para outra representação adequada à detecção e remoção das redundâncias [WCCJ01], [ZLP00], [BMB07], [SABP08], [Raj02], [KP04] e [BESS00].

Os métodos de compressão baseados em transformadas consistem na escolha da base de representação a ser utilizada, no cálculo dos coeficientes de projeção do sinal na base escolhida, na retenção dos coeficientes mais significativos e na quantização e codificação dos coeficientes. A transformada a ser utilizada deve ser escolhida de forma que a energia do sinal esteja concentrada em poucos coeficientes de projeção, e assim poucos coeficientes são suficientes para representar o sinal com uma distorção aceitável. Para aumentar a compressão, os coeficientes retidos podem

ser compactados. O arquivo comprimido deve conter os coeficientes de projeção codificados, a informação de quais coeficientes foram retidos e a base de representação escolhida. Em alguns casos a base não precisa ser armazenada, como no caso da aplicação da transformada DCT (*Discrete Cosine Transform*). Em outros, a base é descrita por alguns parâmetros, como no caso da transformada *Wavelet*.

Um dos primeiros métodos de compressão de eletrocardiogramas usando a transformada DCT foi proposto por Allen & Belina em 1992 [AB92]. A transformada DCT [ANR74] consiste na representação de um sinal utilizando como base a função cosseno. No caso de ECGs, a energia concentra-se nos primeiros coeficientes [AB92] e é comum reter uma certa quantidade de primeiros coeficientes e descartar os restantes, sem portanto necessitar armazenar informações sobre quais coeficientes foram retidos. A transformada DCT é ortogonal, o que simplifica o cálculo dos coeficientes e, conforme o Teorema de Parseval [OWN97], a análise da energia do sinal pode ser feita utilizando tais coeficientes.

Um dos métodos de compressão utilizando transformadas *wavelet* que se destaca é o desenvolvido por Lu & Pearlman, denominado SPIHT [ZLP00] (*Set Partitioning in Hierarchical Trees*). Sua principal característica é a forma de codificar tanto os valores dos coeficientes significativos quanto suas posições em uma única *string* de *bits*. A codificação garante que os coeficientes mais significativos sejam codificados antes dos coeficientes menos significativos e que a transmissão ou o armazenamento da *string* de *bits* possam ser interrompidos a qualquer momento sem comprometer a compressão. Normalmente o sinal é dividido em segmentos de tamanhos iguais e o método é aplicado a cada segmento.

Rajoub [Raj02] propôs um método de compressão de eletrocardiogramas baseado em transformadas *wavelet* que apresenta resultados satisfatórios. O método inicia-se com a retirada do valor médio do sinal e inserção de zeros nas bordas. A transformada *wavelet* é aplicada ao sinal e os coeficientes com valores absolutos maiores que um certo limiar são retidos. O valor de limiar é obtido a partir da análise da energia dos coeficientes em cada subespaço. Os coeficientes de projeção retidos são então quantizados e o *bitmap* que informa a posição dos coeficientes retidos é codificado utilizando uma versão do método *Run Length Encoding* que utiliza um número variável de *bits* para codificar cada sequência de *bits*.

O método desenvolvido por Benzid *et al.* [BMB07], também baseado em transformadas *wavelets*, distingue-se pela escolha dos coeficientes a serem retidos e na codificação do *bitmap* que informa a posição dos coeficientes retidos. Os coeficientes a serem retidos têm valor absoluto maior do que um limiar calculado a partir de um algoritmo de biseção. A codificação dos coeficientes e do *bitmap* é feita utilizando uma variação do método *Run Length Encoding*, denominado *Two-Role Encoder* (TRE).

Tanto o método SPIHT quanto o método TRE utilizam uma função *wavelet* fixa, independente do sinal a ser comprimido. Adequar a *wavelet* ao sinal pode trazer benefícios à compressão, uma vez que a aplicação de certas *wavelets* faz com que a energia do sinal transformado fique mais concentrada em poucos coeficientes de projeção. Chapa & Rao [CR00] determinam a *wavelet* que mais se aproxima de um dado sinal pela resolução de um problema de otimização formulado a partir do módulo e da fase da representação em frequência do sinal. Guido *et al.* [GSK⁺06] apresentam um método para adequar a *wavelet* ao sinal em um contexto diferente do da compressão de sinais. O método consiste na resolução de um sistema linear de equações composto pelas derivadas da energia de projeção do sinal no espaço escala e por um conjunto de restrições relacionadas aos momentos da função *wavelet*. O sistema é resolvido usando o método dos mínimos quadrados resultando em um filtro discreto passa-baixas que corresponde à função escala. A adequação da *wavelet* ao sinal também é estudada por Gupta *et al.* [GJP02], ainda

fora do contexto de compressão de sinais. O método consiste na resolução de um sistema linear de equações composto pelas derivadas da energia do erro de projeção do sinal no subespaço *wavelet* com relação aos coeficientes escala, fixando-se um dos coeficientes em 1. Para simplificar o cálculo dos coeficientes, ambos os artigos consideram uma versão relaxada da maximização da energia de projeção do sinal no subespaço escala e, assim, não garantem que a *wavelet* obtida maximiza a energia projetada.

O método de compressão apresentado por Wei *et al.* [WCCJ01] encontra uma base que minimiza o erro quadrático de projeção do sinal. Um pré-processamento é realizado no sinal, que consiste na detecção e isolamento de cada batimento e em sua normalização utilizando um método de interpolação, para que todos os batimentos apresentem o mesmo número de amostras. A base que minimiza a soma do erro quadrático de projeção é calculada pela aplicação da decomposição em valores singulares (*Singular Values Decomposition* - SVD) no conjunto de batimentos, sendo que apenas alguns elementos da base são retidos. A construção da base utilizando a decomposição SVD, em certas circunstâncias, também pode ser chamada de transformada de Karhunen-Loève [Say06].

Dois métodos de compressão de ECGs são propostos na presente dissertação. O método denominado *Run Length Encoding Adaptativo* (RLEA) consiste primeiramente em encontrar uma *wavelet* que se adapte ao sinal por meio de um processo de otimização para que a distorção seja minimizada. Os coeficientes de projeção mais significativos são retidos, utilizando o método de biseção apresentado em [BMB07] para a escolha dos coeficientes a serem retidos, e depois quantizados usando um número de *bits* diferente para os coeficientes de cada subespaço *wavelet* considerado. Para a compactação tanto dos coeficientes quanto do *bitmap* foram comparadas as técnicas de compactação de Huffman, de Lempel-Ziv-Welch e a técnica *Run Length Encoding* (RLE), sendo escolhida uma versão modificada da técnica RLE. A técnica de compactação proposta baseia-se na análise do *bitmap*: se um *bit* 1 ocorrer são inseridos na saída o *bit* 1 e o respectivo coeficiente quantizado; se um *bit* 0 ocorrer são inseridos o *bit* 0 e a codificação RLE da sequência de *bits* 0 que ocorre a partir do *bit* analisado. O número de *bits* utilizados para representar a codificação RLE varia com o subespaço considerado.

O segundo método proposto nesta dissertação, denominado *Zero Padding Singular Values Decomposition* (ZPSVD), é baseado no método apresentado em [WCCJ01] e consiste na utilização da SVD para obter uma base de representação adaptada ao sinal a ser comprimido. Apenas os elementos mais significativos da base são retidos de forma a satisfazer uma medida de distorção pré-especificada. Antes de aplicar o SVD no sinal é necessário realizar um pré-processamento que consiste na detecção e isolamento dos batimentos do ECG e na equalização dos tamanhos dos batimentos preenchendo-os com zeros (*zero padding*), sendo que o método de detecção e isolamento dos batimentos é baseado no método apresentado em [HT86], com modificações propostas nesta dissertação que melhoram a precisão da detecção. Os coeficientes de projeção dos batimentos na base encontrada são quantizados utilizando um número diferente de *bits* para os coeficientes referentes a cada um dos elementos da base. Os elementos retidos da base são codificados pela aplicação da transformada *wavelet* em cada elemento, retenção dos coeficientes mais significativos e quantização e codificação dos coeficientes utilizando os mesmos métodos aplicados aos coeficientes de decomposição no método RLEA.

A dissertação está organizada da seguinte forma. Três métodos de compactação de dados são descritos no capítulo 1. No capítulo 2 são apresentados uma descrição geral de eletrocardiogramas, um algoritmo de detecção de batimentos cardíacos e as medidas de desempenho utilizadas na compressão de eletrocardiogramas. Uma análise sobre as principais propriedades relacionadas à transformada *wavelet* é apresentada no capítulo 3, bem como o procedimento

de parametrização de funções *wavelet* e a formulação do algoritmo de otimização para determinação da *wavelet* que melhor se ajusta ao sinal. O método de compressão por *wavelets*, que consiste na aplicação da transformada *wavelet* utilizando como função base a *wavelet* que melhor se ajusta ao sinal e na quantização e codificação dos coeficientes de projeção, é apresentado em detalhes no capítulo 4. No capítulo 5 são apresentados o método de compressão baseado na decomposição em valores singulares e sua comparação com a compressão por *wavelets*. A seção de conclusões descreve as contribuições da dissertação.

Compactação

1.1 Introdução

Os métodos de compactação são utilizados para reduzir o tamanho de arquivos de forma que, quando reconstruídos, não apresentem distorção em relação aos dados originais. A compactação é utilizada, por exemplo, para reduzir o tamanho de textos e de programas computacionais cujos conteúdos precisam ser preservados. Os métodos de compactação são normalmente divididos em duas categorias [Sal07]: os métodos de dicionários, em que sequências de símbolos são codificadas utilizando um dicionário construído para diminuir a redundância dos dados compactados, como o método de Lempel-Ziv-Welch (LZW) [ZL77], [Wel84]; e os métodos estatísticos, em que um modelo estatístico é utilizado para representar os dados, como no método de Huffman [Huf52]. O método *Run Length Encoding* (RLE) [Gol66] pode ser classificado como estatístico, embora também tenha características dos métodos de dicionário.

Na compactação de dados considera-se que a entrada é formada por uma sequência de símbolos, de forma que um símbolo pode ser, por exemplo, uma letra (no caso de compactação de textos) ou uma sequência de *bits* de tamanho fixo (no caso de compactação de arquivos binários). A cada símbolo é atribuído um código cuja representação normalmente ocupa um espaço menor do que o espaço ocupado pelo símbolo correspondente, sendo que pode-se também atribuir códigos a conjuntos (*strings*) de símbolos. Basicamente os métodos de compactação consistem em encontrar uma codificação para os símbolos ou para conjuntos de símbolos de forma que a representação da sequência codificada ocupe um espaço menor do que a representação da sequência original. Neste capítulo são apresentados três métodos de compactação: O método de Huffman [Huf52], que define os códigos baseado na frequência relativa de ocorrência de cada símbolo na sequência de entrada; o método LZW [ZL77], [Wel84], que constrói um dicionário que relaciona símbolos e códigos pela análise dos dados de entrada; e o método RLE [Gol66], que codifica o número de vezes consecutivas que um mesmo símbolo aparece na sequência de entrada.

O capítulo está organizado da seguinte forma. Na Seção 1.2 é apresentado o método de Huffman, na Seção 1.3 é apresentado o método LZW e na Seção 1.4 é apresentado o método RLE. Uma comparação entre os métodos e a introdução de uma medida de desempenho para métodos de compactação e compressão são apresentadas na Seção 1.5. A Seção 1.6 conclui o capítulo.

1.2 Método de Huffman

No método de Huffman [Huf52], cada símbolo é codificado utilizando códigos cujos tamanhos variam de acordo com a frequência de ocorrência de cada símbolo na sequência de entrada de forma que, quanto maior a frequência de ocorrência de um símbolo, menor é o tamanho do código utilizado para representá-lo. Quando as frequências de ocorrência dos símbolos são potências negativas de dois, os tamanhos dos códigos produzidos pelo método de Huffman são os menores possíveis [Sal07].

Sejam $\alpha_1, \alpha_2, \dots, \alpha_n$ os símbolos de uma dada sequência a ser compactada. Cada símbolo possui uma frequência relativa de ocorrência $p(\alpha_k)$ tal que

$$\sum_{k=1}^n p(\alpha_k) = 1, \quad p(\alpha_k) > 0, \quad k = 1, \dots, n \quad (1.1)$$

e o tamanho do código utilizado no método de Huffman para representar cada símbolo é dado por $s(\alpha_k)$. Na prática, cada código é representado por uma sequência binária e, portanto, o tamanho do código é dado pelo número de *bits* utilizado para representá-lo. Sem perda de generalidade, suponha que as frequências de ocorrência dos símbolos α_k estão ordenados da seguinte forma:

$$p(\alpha_1) \geq p(\alpha_2) \geq \dots \geq p(\alpha_{n-1}) \geq p(\alpha_n). \quad (1.2)$$

Para que os códigos gerados pelo método sejam de mínima redundância [Huf52] as seguintes restrições devem ser satisfeitas:

1. Dois símbolos diferentes não podem ser codificados pelo mesmo código;
2. Símbolos com maior frequência de ocorrência devem ser representados com códigos de tamanho menor, ou seja:

$$s(\alpha_1) \leq s(\alpha_2) \leq \dots \leq s(\alpha_{n-1}) \leq s(\alpha_n); \quad (1.3)$$

3. Os códigos são construídos de forma que não sejam necessárias indicações de onde um código começa ou termina. Para isso, os primeiros k *bits* de um código (prefixo de ordem k) não podem ser iguais aos de outro código;
4. Os dois símbolos que apresentam as menores frequências de ocorrência possuem códigos de mesmo tamanho, ou seja,

$$s(\alpha_1) \leq s(\alpha_2) \leq \dots \leq s(\alpha_{n-1}) = s(\alpha_n). \quad (1.4)$$

Neste caso, os prefixos de ordem $s(\alpha_n) - 1$ de α_{n-1} e α_n são iguais e a única diferença entre α_{n-1} e α_n é o último *bit*.

No método de Huffman, primeiramente devem-se calcular as frequências relativas de ocorrência de cada símbolo na sequência a ser compactada. Seja $\alpha_1, \dots, \alpha_n$ a lista com os símbolos ordenados conforme a Equação (1.2). De acordo com a restrição 4 os dois últimos símbolos devem ser codificados com o mesmo código exceto pelo último *bit*. Uma vez definidos os dois códigos, os símbolos α_n e α_{n-1} são retirados da lista e o símbolo composto $\alpha_{n-1,n}$ é inserido na lista ordenada pelas frequências, sendo que o código de $\alpha_{n-1,n}$ é igual ao prefixo de ordem $s(\alpha_n) - 1$ dos símbolos α_n e α_{n-1} , ainda a ser definido, e sua frequência de ocorrência é igual à

Tabela 1.1: Exemplo do método de Huffman.

Iteração 0		Iteração 1		Iteração 2		Iteração 3		Final	
Sím	Freq	Sím	Freq	Sím	Freq	Sím	Freq	Sím	Cód
α_2	0.4	α_2	0.4	α_2	0.4	$\alpha_{3,4,5,1}$	0.6	α_1	00
α_1	0.2	α_1	0.2	$\alpha_{3,4,5}$	0.4	α_2	0.4	α_2	1
α_3	0.2	α_3	0.2	α_1	0.2	Sím	Cód	α_3	010
α_4	0.1	$\alpha_{4,5}$	0.2	Sím	Cód	α_1	$(\alpha_{3,4,5,1})0$	α_4	0110
α_5	0.1	Sím	Cód	α_3	$(\alpha_{3,4,5})0$	α_3	$(\alpha_{3,4,5,1})10$	α_5	0111
		α_4	$(\alpha_{4,5})0$	α_4	$(\alpha_{3,4,5})10$	α_4	$(\alpha_{3,4,5,1})110$		
		α_5	$(\alpha_{4,5})1$	α_5	$(\alpha_{3,4,5})11$	α_5	$(\alpha_{3,4,5,1})111$		

soma das frequências dos símbolos α_n e α_{n-1} . O procedimento é aplicado recorrentemente até que sobrem apenas dois símbolos na lista, sendo atribuído a cada um dos símbolos o código 0 e 1 e recorrentemente definindo todos os códigos referentes aos demais símbolos. A Tabela 1.1 mostra um exemplo com cinco símbolos, sendo as frequências de ocorrência de cada símbolo dadas por

$$p(\alpha_1) = 0.2, \quad p(\alpha_2) = 0.4, \quad p(\alpha_3) = 0.2, \quad p(\alpha_4) = 0.1, \quad p(\alpha_5) = 0.1. \quad (1.5)$$

Na primeira iteração os símbolos de menor frequência são α_4 e α_5 . De acordo com a restrição 4 os códigos dos dois símbolos possuem o mesmo prefixo $\alpha_{4,5}$ e apresentam diferenças apenas no último dígito. O prefixo $\alpha_{4,5}$ é considerado o novo símbolo e é inserido na lista. Na segunda iteração os símbolos $\alpha_{4,5}$ e α_3 são os que têm a menor frequência de ocorrência, portanto apresentam o mesmo prefixo $\alpha_{3,4,5}$. O símbolo α_3 é então codificado como $\alpha_{3,4,5}0$, o símbolo α_4 como $(\alpha_{4,5})0 \rightarrow (\alpha_{3,4,5}1)0$ e o símbolo α_5 como $(\alpha_{4,5})1 \rightarrow (\alpha_{3,4,5}1)1$. O procedimento continua até que sobrem apenas os símbolos $\alpha_{3,4,5,1}$ codificado como 0 e α_2 codificado como 1, sendo definidos também os códigos dos símbolos restantes.

O método de Huffman também pode ser implementado com o uso de árvores binárias, sendo que cada símbolo é uma folha da árvore, o antecessor de cada nó é a composição dos dois nós sucessores e o código de cada símbolo é a descrição do caminho entre a raiz da árvore e a folha correspondente ao símbolo, com 0 codificando o ramo esquerdo e 1 codificando o ramo direito.

Além dos símbolos codificados, a saída do algoritmo deve conter a tabela que relaciona cada símbolo com seu respectivo código uma vez que diferentes sequências podem gerar diferentes codificações para os mesmos símbolos. A descompactação se dá pela detecção de cada código (procedimento possível se a restrição 3 for seguida) e pela consulta ao dicionário para encontrar o respectivo símbolo.

1.3 Lempel-Ziv-Welch

Em muitas aplicações a sequência a ser codificada apresenta várias repetições de um mesmo conjunto de símbolos, como por exemplo a ocorrência de uma mesma palavra em diversas partes de um texto. As técnicas de compactação baseadas em dicionários se aproveitam desta característica para realizar a compactação dos dados. Basicamente, as técnicas de dicionários consistem na construção de uma tabela (dicionário) cujas entradas consistem tanto nos símbolos

quanto nos conjuntos de símbolos que ocorrem na sequência a ser compactada, sendo que cada entrada é relacionada com um código, comumente o índice da entrada no dicionário. O dicionário pode ser constante durante toda a aplicação do método (dicionário estático) ou pode ser alterado dependendo dos padrões que ocorrerem na sequência a ser compactada (dicionário adaptativo) [Sal07].

O método de Lempel-Ziv-Welch (LZW) [ZL77], [Wel84] é um método baseado em dicionários adaptativos. A grande vantagem do método LZW é construir, durante a análise da sequência de entrada, o dicionário de forma que este possa ser reconstruído no procedimento de descompactação sem que seja necessário armazenar qualquer informação sobre o dicionário durante a compactação. Em algumas implementações do método LZW o tamanho do dicionário é limitado a um certo número de entradas, porém na implementação mais usual o dicionário não apresenta limite de tamanho.

O primeiro passo do método é a inicialização do dicionário com todos os possíveis símbolos que podem aparecer na sequência de entrada. Por exemplo, se cada símbolo é representado por 8 *bits*, então o dicionário é inicializado com todos os 256 possíveis símbolos. Em seguida a variável p é inicializada com o primeiro símbolo da entrada e a variável α é inicializada com o símbolo seguinte. Verifica-se se a *string* $p\alpha$, formada pela concatenação de p com α , pertence ao dicionário. Se $p\alpha$ estiver no dicionário então $p \leftarrow p\alpha$ e α recebe o próximo símbolo da sequência de entrada. Caso contrário, o código referente a p é inserido na sequência de saída (sequência compactada), a *string* $p\alpha$ é inserida no dicionário, $p \leftarrow \alpha$ e α recebe o próximo símbolo da sequência de entrada. O algoritmo prossegue até que todos os símbolos da entrada sejam analisados.

A Tabela 1.2 ilustra a aplicação do método na compactação da sequência *TOBEORNOTTOBEORTOBEORNOT*. Como cada caractere é representado pelo seu código ASCII de 8 *bits* o dicionário inicial contém 256 entradas, com os códigos ASCII de cada caractere utilizado no exemplo dados por

$$B = \langle 66 \rangle, \quad E = \langle 69 \rangle, \quad N = \langle 78 \rangle, \\ O = \langle 79 \rangle, \quad R = \langle 82 \rangle, \quad T = \langle 84 \rangle.$$

Note que é necessário um certo número de iterações para que o dicionário contenha entradas que proporcionem uma boa compactação. Assim, a sequência de entrada precisa apresentar um tamanho mínimo para que a compactação pelo método LZW apresente um bom desempenho. Note também que no exemplo são necessários 9 *bits* para representar cada símbolo na sequência de saída, enquanto que cada símbolo na entrada é representado por 8 *bits*. O número reduzido de símbolos de saída compensa o aumento no número de *bits* usados para representar cada símbolo na saída.

Na compactação, toda vez que a *string* $p\alpha$ não pertencer ao dicionário são executados três procedimentos: o código da *string* p é inserido na sequência de saída, $p\alpha$ é inserida no dicionário e $p \leftarrow \alpha$. Note que, na compactação, a atualização no dicionário com a *string* $p\alpha$ ocorre sempre em conjunto com a inserção do código de p na sequência compactada, sendo que a próxima *string* codificada e inserida na saída se inicia com o símbolo α . Portanto, na descompactação, a atualização do dicionário se dá após o processamento de cada código da sequência compactada sendo que a nova entrada do dicionário consiste na *string* p decodificada concatenada com o primeiro símbolo da *string* decodificada seguinte.

O primeiro passo no procedimento de descompactação, assim como na compactação, é inicializar o dicionário com todos os possíveis símbolos. Em seguida, para cada código, obtém-se

Tabela 1.2: Exemplo do método LZW na codificação da sequência *TOBEORNOTTOBEORTOBEORNOTTO*.

Iteração	p	No dicionário?	Nova entrada	Saída
1	T	Sim		
2	TO	Não	$TO = \langle 256 \rangle$	$T = \langle 84 \rangle$
3	OB	Não	$OB = \langle 257 \rangle$	$O = \langle 79 \rangle$
4	BE	Não	$BE = \langle 258 \rangle$	$B = \langle 66 \rangle$
5	EO	Não	$EO = \langle 259 \rangle$	$E = \langle 69 \rangle$
6	OR	Não	$OR = \langle 260 \rangle$	$O = \langle 79 \rangle$
7	RN	Não	$RN = \langle 261 \rangle$	$R = \langle 82 \rangle$
8	NO	Não	$NO = \langle 262 \rangle$	$N = \langle 78 \rangle$
9	OT	Não	$OT = \langle 263 \rangle$	$O = \langle 79 \rangle$
10	TT	Não	$TT = \langle 264 \rangle$	$T = \langle 84 \rangle$
11	TO	Sim		
12	TOB	Não	$TOB = \langle 265 \rangle$	$TO = \langle 256 \rangle$
13	BE	Sim		
14	BEO	Não	$BEO = \langle 266 \rangle$	$BE = \langle 258 \rangle$
15	OR	Sim		
16	ORT	Não	$ORT = \langle 267 \rangle$	$OR = \langle 260 \rangle$
17	TO	Sim		
18	TOB	Sim		
19	$TOBE$	Não	$TOBE = \langle 268 \rangle$	$TOB = \langle 265 \rangle$
20	EO	Sim		
21	EOR	Não	$EOR = \langle 269 \rangle$	$EO = \langle 259 \rangle$
22	RN	Sim		
23	RNO	Não	$RNO = \langle 270 \rangle$	$RN = \langle 261 \rangle$
24	OT	Sim		
25	$OT \langle EOF \rangle$			$OT = \langle 263 \rangle$

a *string* p_1 correspondente ao código atual e a *string* p_2 correspondente ao código seguinte. Insire-se p_1 na sequência de saída e atualiza-se o dicionário com a entrada $p_1\alpha$, sendo α o primeiro símbolo da *string* p_2 . O algoritmo prossegue até que a última *string* seja decodificada. A Tabela 1.3 ilustra a aplicação do método na descompactação da sequência compactada *TOBEORNOTTOBEORTOBEORNOTTO*.

1.4 Run Length Encoding

No método *Run Length Encoding* (RLE) [Gol66], se um símbolo α_k aparecer n vezes consecutivas na entrada a ser compactada então toda a sequência de símbolos é codificada pelo par $n\alpha_k$ [Sal07]. Se a entrada for binária é possível informar apenas o número n de vezes consecutivas que um símbolo aparece, pois existem apenas dois símbolos diferentes. O método RLE é considerado um método de compactação estatístico pois em algumas variações do método o número de *bits* utilizados para representar os tamanhos das sequências de símbolos é variável

Tabela 1.3: Exemplo do método LZW na decodificação da sequência cuja codificação é mostrada na Tabela 1.2.

Iteração	Atual	Próximo	Entrada dicionário	Saída
1	$T = \langle 84 \rangle$	$O = \langle 79 \rangle$	$TO = \langle 256 \rangle$	T
2	$O = \langle 79 \rangle$	$B = \langle 66 \rangle$	$OB = \langle 257 \rangle$	O
3	$B = \langle 66 \rangle$	$E = \langle 69 \rangle$	$BE = \langle 258 \rangle$	B
4	$E = \langle 69 \rangle$	$O = \langle 79 \rangle$	$EO = \langle 259 \rangle$	E
5	$O = \langle 79 \rangle$	$R = \langle 82 \rangle$	$OR = \langle 260 \rangle$	O
6	$R = \langle 82 \rangle$	$N = \langle 78 \rangle$	$RN = \langle 261 \rangle$	R
7	$N = \langle 78 \rangle$	$O = \langle 79 \rangle$	$NO = \langle 262 \rangle$	N
8	$O = \langle 79 \rangle$	$T = \langle 84 \rangle$	$OT = \langle 263 \rangle$	O
9	$T = \langle 84 \rangle$	$TO = \langle 256 \rangle$	$TT = \langle 264 \rangle$	T
10	$TO = \langle 256 \rangle$	$BE = \langle 258 \rangle$	$TOB = \langle 265 \rangle$	TO
11	$BE = \langle 258 \rangle$	$OR = \langle 260 \rangle$	$BEO = \langle 266 \rangle$	BE
12	$OR = \langle 260 \rangle$	$TOB = \langle 265 \rangle$	$ORT = \langle 267 \rangle$	OR
13	$TOB = \langle 265 \rangle$	$EO = \langle 259 \rangle$	$TOBE = \langle 268 \rangle$	TOB
14	$EO = \langle 259 \rangle$	$RN = \langle 261 \rangle$	$EOR = \langle 269 \rangle$	EO
15	$RN = \langle 261 \rangle$	$OT = \langle 263 \rangle$	$RNO = \langle 270 \rangle$	RN
16	$OT = \langle 263 \rangle$	—	—	OT

com a entrada a ser compactada e é determinado pela análise estatística dos tamanhos das sequências detectadas. Pode-se utilizar, por exemplo, a média dos tamanhos das sequências para inferir o número de *bits* de representação dos tamanhos das sequências. Existem ainda as implementações em que a representação dos tamanhos de cada sequência utiliza um número diferente de *bits* e, nesse caso, é necessário informar o número de *bits* utilizados junto com o par $n\alpha_k$. A seguir é mostrado um exemplo de compactação utilizando o método RLE, considerando que são usados 3 *bits* para representar o tamanho de cada sequência, ou seja, no máximo pode-se representar sequências de 8 símbolos consecutivos usando um único código:

$$AAAAAABBBBAAABAAAAAABBBB \Rightarrow 6A 3B 3A 1B 8A 2A 4B \quad (1.6)$$

A descompactação consiste em aplicar o processo inverso da compactação: detectar o tamanho da repetição do símbolo, decodificar o símbolo em questão e reproduzir a repetição na saída descompactada. A detecção das informações necessárias depende de como tais informações foram codificadas, sendo que o número de *bits* utilizados para codificar os tamanhos das sequências ou é fixo ou é inserido no início do arquivo compactado. Note que, se a entrada não possuir repetições consecutivas, o método RLE, além de não compactar a entrada, pode resultar em uma codificação maior do que a própria entrada.

1.5 Comparações entre os métodos

Na presente seção os três métodos de compactação apresentados são comparados por meio de experimentos numéricos que consistem na compactação de dois textos e de um arquivo binário, sendo que os textos não possuem caracteres com acento e letras em maiúsculo para facilitar a

aplicação dos métodos. As entradas a serem compactadas nos experimentos foram escolhidas de forma a ressaltar as principais características de cada um dos métodos de compactação. Para comparar os métodos de compactação é utilizada como medida de desempenho a taxa de compressão (*Compression Ratio* — CR). A taxa de compressão é a medida de desempenho mais utilizada para comparações tanto entre métodos de compactação quanto em métodos de compressão e é dada por

$$CR = \frac{\text{Número de bits para representar os dados originais}}{\text{Número de bits para representar os dados compactados}}. \quad (1.7)$$

Para a execução dos experimentos tanto no método de Huffman quanto no método LZW foi considerado que cada símbolo é composto por 8 *bits*, sendo que no método LZW nenhum limite para o tamanho do dicionário foi imposto. No experimento cuja entrada é uma sequência binária (*bitmap*), cada grupo de 8 *bits* consecutivos compõe um símbolo. O método RLE foi implementado limitando em 6 o número de *bits* usados para representar o tamanho das sequências consecutivas de símbolos e a informação do tamanho de cada sequência de símbolos na saída compactada é precedida pela informação do próprio símbolo.

Nos experimentos executados, a primeira entrada a ser compactada consiste no seguinte texto:

“voila! in view, a humble vaudevillian veteran, cast vicariously as both victim and villain by the vicissitudes of fate. this visage, no mere veneer of vanity, is it vestige of the vox populi, now vacant, vanished, as the once vital voice of the verisimilitude now venerates what they once vilified. however, this valorous visitation of a bygone vexation stands vivified, and has vowed to vanquish these venal and virulent vermin vanguarding vice and vouchsafing the violently vicious and voracious violation of volition. the only veredict is vengeance, a vendetta held as a votive, not in vain, for the value and veracity of such shall one day vindicate the vigilant and the virtuous. verily, this vichyssoise of verbiage veers most verbose vis a vis an introduction, and so it is my very good honor to meet you and you may call me v.” (Trecho do filme “*V for Vendetta*”).

A segunda entrada a ser compactada consiste no seguinte texto:

“joao amava teresa, que amava raimundo, que amava maria, que amava joaquim, que amava lili, que amava marcelo, que amava karina, que amava abel, que amava barbara, que amava cesar, que amava daiane, que amava edilson, que amava telma, que amava sidnei, que amava rosana, que amava lauro, que amava eleonora, que amava adriano, que amava thais, que amava paulo, que amava rosana, que amava leonardo, que amava vivian, que amava renato, que amava aline, que amava gabriel, que amava valeria, que amava regis, que amava carla, que nao amava ninguem.” (Inspirado no poema “A Quadrilha”, de Carlos Drummond de Andrade).

A terceira entrada a ser compactada consiste no *bitmap* de representação da imagem mostrada na Figura 1.1. O *bitmap* é construído de forma que cada *pixel* da cor branca é representado por um *bit* 1 e cada *pixel* da cor preta é representado por um *bit* 0.

A Tabela 1.4 mostra as taxas de compressão obtidas com a aplicação de cada método em cada um dos experimentos. A entrada do primeiro experimento é marcada pela frequente ocorrência não consecutiva de alguns poucos símbolos, como os caracteres *v*, *e* e *i*. Tal configuração privilegia o método de Huffman, que aloca códigos que utilizam poucos *bits* para representar os símbolos que têm maior frequência de ocorrência, aumentando a taxa de compressão resultante. O método LZW apresenta um resultado um pouco pior uma vez que não há repetições significativas de sequências de símbolos, e o método RLE apresenta um resultado ruim já que não há a ocorrência de repetições consecutivas de símbolos. No segundo experimento, a entrada



Figura 1.1: Imagem cujo *bitmap* de representação é utilizado no Experimento 3.

Tabela 1.4: Taxas de compressão obtidas pela aplicação do método de Huffman, do método LZW e do método RLE em cada uma das entradas descritas.

Entrada	Huffman	LZW	RLE
1	1.78	1.50	0.81
2	1.92	2.25	0.89
3	3.36	3.23	4.12

é marcada pela frequente ocorrência de repetições de sequências de símbolos, como as palavras “que” e “amava” e o texto em si é relativamente grande, características que privilegiam o método LZW. Como não há uma notável frequência de ocorrência de um símbolo específico o método de Huffman apresenta um resultado um pouco pior e o método RLE novamente apresenta um resultado ruim devido à não ocorrência de repetições consecutivas de símbolos. No terceiro experimento, a entrada é uma sequência binária em que pode-se observar frequentes repetições consecutivas dos símbolos, situação que privilegia o método RLE.

1.6 Conclusão

Neste capítulo foram apresentados três métodos de compactação: o método de Huffman, o método LZW e o método RLE. O método de Huffman codifica os símbolos utilizando códigos de tamanho variável, sendo que o tamanho de cada código é inversamente proporcional à frequência de ocorrência do respectivo símbolo na entrada a ser compactada. No método LZW os símbolos são codificados a partir de um dicionário, construído pela análise dos símbolos da entrada de forma que o dicionário pode ser completamente reconstruído a partir da análise da sequência compactada, sendo desnecessário transmitir ou armazenar o dicionário em conjunto com a sequência compactada. O método RLE consiste na codificação das repetições consecutivas de um mesmo símbolo na entrada, informando a quantidade de repetições de cada símbolo. Uma comparação entre os três métodos foi realizada, concluindo-se que nenhum dos métodos é melhor do que o outro em um caso geral. Para escolher qual método de compactação deve ser usado em uma certa entrada é necessário ter um conhecimento *a priori* do comportamento da entrada. O método de Huffman é adequado para compactar dados que apresentam uma grande repetição não consecutiva de alguns símbolos, o método LZW apresenta bons resultados para entradas que apresentem repetições de conjuntos de símbolos e o método RLE se comporta melhor para entradas que contenham um grande número de repetições de símbolos consecutivos. Em geral

os métodos de Huffman e LZW apresentam bons resultados para praticamente todos os casos, sendo portanto considerados bons métodos de compactação para casos gerais. O método RLE, no entanto, apresenta resultados ruins se a entrada não for adequada, o que o torna um método de compactação mais específico para entradas que apresentem longas sequências de repetições de um mesmo símbolo.

Eletrocardiogramas

2.1 Introdução

Um eletrocardiograma (ECG) é o registro das atividades elétricas do coração durante um determinado período de tempo [Bra97]. Os ECGs são ferramentas clínicas muito utilizadas no diagnóstico de cardiopatias, sendo ocasionalmente os únicos indicadores de certos processos patológicos, podendo também serem utilizados como guias para terapias e tratamentos. As atividades elétricas do coração podem ser medidas por eletrodos colocados sobre a pele do paciente medindo a atividade de diferentes partes do músculo cardíaco.

A duração do ECG obtido e o número de eletrodos utilizados depende do tipo de exame a que o paciente é submetido. O ECG de repouso é o exame mais simples, realizado em poucos minutos, utilizando normalmente 10 a 12 eletrodos. É utilizado para uma avaliação cardíaca geral do paciente e para controle evolutivo de uma doença cardíaca já confirmada, enquanto que o ECG de esforço consiste na obtenção do ECG no momento em que o paciente realiza alguma atividade física, como corrida em uma esteira, e serve para avaliar o coração sob condições de estresse. Tanto o ECG de repouso quanto o ECG de esforço são exames que resultam em registros de ECG de pouca duração. Na monitoração de Holter [Ken94], utilizada para detecção de arritmias, armazenam-se os eletrocardiogramas por um período de 24 horas. Os ECGs obtidos são armazenados ou transmitidos para um destino apropriado, sendo a transmissão cada vez mais utilizada com o crescente avanço da telemedicina [CKS⁺06], [GFAG08], [Kom05].

A compressão de ECGs é, portanto, imprescindível para que se possa armazenar uma quantidade maior de informações ou para que se utilize melhor a banda de transmissão do sinal. A compressão deve apresentar uma distorção aceitável, para que o sinal resultante não gere diagnósticos equivocados.

O capítulo está organizado da seguinte forma. Na Seção 2.2 são descritas as características gerais de um ECG, uma breve análise sobre a *baseline* de um ECG é feita e o repositório *online* de bases de dados de sinais fisiológicos *PhysioBank* [Phy] é descrito. Um algoritmo para detecção e isolamento dos batimentos, procedimentos utilizados em alguns métodos de compressão de ECGs, é apresentado na Seção 2.3. Na Seção 2.4 é realizada uma análise das medidas de desempenho normalmente utilizadas nas compressões de ECGs e a Seção 2.5 conclui o capítulo.

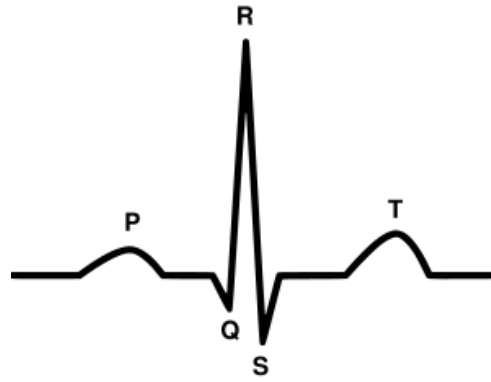


Figura 2.1: Representação esquemática de um batimento de um ECG (Fonte: [HTTP://EN.WIKIPEDIA.ORG/WIKI/ELECTROCARDIOGRAM](http://en.wikipedia.org/wiki/Electrocardiogram)).

2.2 Características dos eletrocardiogramas

Um ECG é tipicamente dividido nos seguintes componentes: a onda P, o intervalo PR, o complexo QRS, o segmento ST e a onda T. A Figura 2.1 mostra cada um desses componentes em um ECG típico. Cada componente está relacionado com uma atividade muscular que ocorre durante um batimento cardíaco. Dessa forma, é possível observar os distúrbios que podem ocorrer em cada grupo muscular do coração de um paciente através da análise do comportamento de cada um dos componentes do ECG. Algumas das características dos ECGs que são analisadas no diagnóstico de uma cardiopatia são:

- Diferenças nos tamanhos dos batimentos;
- Formatos dos componentes;
- Diferenças nas amplitudes dos componentes entre cada batimento;
- Relação entre a onda P e o complexo QRS;
- Diferenças na duração dos componentes.

Durante o batimento o coração realiza contrações (sístoles), geradas pela despolarização elétrica das respectivas seções, e relaxamentos (diástoles), geradas por repolarização. Um batimento cardíaco inicia-se com a geração de uma corrente elétrica pelo nó sinoatrial, que causa a despolarização atrial seguida da despolarização ventricular até que a corrente elétrica atinja o nó atrioventricular. Com o fim da despolarização iniciam-se os processos de repolarização atrial e ventricular. Após um pequeno intervalo outro batimento é iniciado [Bra97].

A despolarização atrial é representada pela onda P, enquanto que a despolarização ventricular é representada pelas ondas Q, R e S (complexo QRS), sendo que a onda Q representa a primeira deflexão negativa, a onda R representa a primeira deflexão positiva e a onda S representa a segunda deflexão negativa. A repolarização ventricular é representada pela onda T. O intervalo PR representa o atraso entre a chegada do impulso elétrico no nó atrioventricular e o processo de despolarização, enquanto que o segmento ST representa o início da repolarização ventricular [ASAZA03].

2.2.1 Base de dados

Um requisito importante na comparação entre métodos e algoritmos em geral é a disponibilidade de uma base de dados que contenha diversos exemplos de variados tipos do objeto de análise, sendo possível dessa forma comparar os métodos e algoritmos utilizando os mesmos exemplos. Na análise de ECGs a base de dados normalmente utilizada para medir o desempenho dos métodos de compressão está contida no conjunto de banco de dados denominado *PhysioBank* [Phy], [GAG⁺00].

O *PhysioBank* é um repositório contendo diversos sinais fisiológicos digitalizados. Os bancos de dados contendo ECGs são divididos por cardiopatias, duração dos sinais obtidos, tipo de exame realizado para a obtenção dos sinais, etc. O banco de dados mais utilizado nos trabalhos que apresentam métodos de compressão de ECGs é o banco de dados de sinais de arritmia, disponibilizado pelo MIT-BIH (*Massachusetts Institute of Technology - Beth Israel Hospital*). O banco de dados de arritmia contém 48 sinais, cada um com 30 minutos de duração, digitalizado com uma frequência de amostragem de 360 Hz. Cada amostra é quantizada utilizando 11 *bits*, sendo que cada amostra é um valor inteiro positivo. Para que todas as amostras dos sinais no banco de dados sejam positivas, facilitando o armazenamento, um valor DC de 1024 é somado a todas as amostras. O acesso aos bancos de dados do *PhysioBank* foi realizado pelos programas contidos no pacote WFDB [GAG⁺00], executados no Matlab, versão 7, *release* 14.

2.2.2 Baseline

Em alguns métodos de compressão o sinal passa por uma fase de pré-processamento, para que o sinal resultante apresente certas propriedades que simplificam sua compressão, como por exemplo a retirada de ruídos ou outros componentes do ECG. Um tipo de ruído muito comum em ECGs é a *baseline*, uma curva normalmente de baixa frequência que é somada ao ECG original. A Figura 2.2 mostra um exemplo de um ECG e sua *baseline*. A *baseline* é uma consequência do processo de aquisição do ECG e pode ser originada por movimentos do paciente, respiração, problemas de contato nos eletrodos ou interferência da rede elétrica [BFB03].

Existem na literatura métodos para remoção da *baseline* [BFB03], [KHO02]. O método proposto nesta dissertação consiste na retirada da *baseline* do sinal utilizando a transformada *Discrete Cosine Transform* (DCT). Como a *baseline* é de baixa frequência, basta retirar os primeiros coeficientes da transformação DCT. Experimentos com diversos ECGs mostram que o número de coeficientes DCT a ser retirado é igual ao número de batimentos [AB08b].

2.3 Detecção de batimentos

A detecção dos batimentos de um ECG é um processo importante em diversas aplicações, principalmente no diagnóstico de cardiopatias como as arritmias [HT86], [KHO02]. Técnicas de detecção de batimentos são também utilizadas em alguns métodos de compressão [WCCJ01].

Cada batimento é definido como o segmento entre picos de duas ondas R consecutivas [Bra97]. O método de detecção dos picos das ondas R descrito a seguir é baseado em [HT86]. O primeiro passo consiste no pré-processamento do ECG, aplicando-o a um conjunto de filtros e executando uma série de operações, conforme ilustrado na Figura 2.3. O filtro passa-baixas (FPB) é implementado pela equação a diferenças dada por

$$y[n] = 2y[n - 1] - y[n - 2] + x[n] - 2x[n - 6] + x[n - 12], \quad (2.1)$$

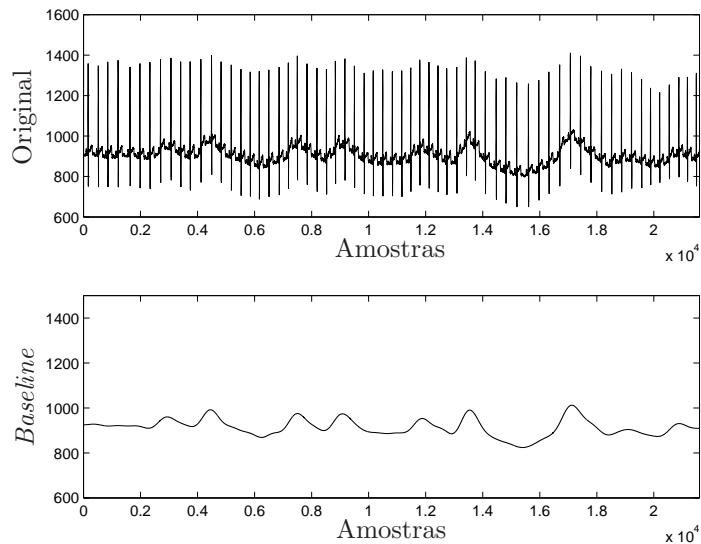


Figura 2.2: ECG original e sua respectiva *baseline*. O ECG mostrado corresponde ao primeiro minuto do ECG de número 115 da base de dados de arritmia do MIT-BIH [Phy].

o filtro passa-altas (FPA) é dado por

$$y[n] = y[n - 1] - \frac{x[n]}{32} + x[n - 16] - x[n - 17] + \frac{x[n - 32]}{32} \quad (2.2)$$

e a equação a diferenças aproxima a operação de diferenciação $\Delta[n]$ por

$$y[n] = \frac{2x[n] + x[n - 1] - x[n - 3] - 2x[n - 4]}{8}. \quad (2.3)$$

Na implementação das equações a diferenças considera-se que

$$x[n] = 0 \text{ e } y[n] = 0 \text{ para } n < 0.$$

A saída do bloco $(\)^2$ é igual à entrada elevada ao quadrado e a saída do bloco $\frac{1}{32} \sum_{k=1}^{32}$ é calculada somando-se os 32 valores mais recentes da entrada e dividindo-se o resultado por 32. Um exemplo de um ECG pré-processado é mostrado na Figura 2.4.

Note que um pico no sinal pré-processado corresponde a uma amostra próxima do pico de uma onda R no sinal original. O próximo passo no algoritmo consiste em detectar os picos do sinal pré-processado e então encontrar os correspondentes picos das ondas R. A detecção de picos do sinal pré-processado é realizada em duas fases. Na primeira fase considera-se que os valores dos picos são sempre maiores que o valor médio do sinal. No entanto, dependendo do nível de ruído, o sinal pré-processado pode apresentar picos que não correspondam a picos de ondas R. A segunda fase consiste em detectar os falsos picos. Para sua detecção calculam-se os tamanhos dos intervalos entre dois picos consecutivos e considera-se que a duração de um batimento não é menor que o valor médio das durações dos batimentos. Se for detectado um intervalo entre

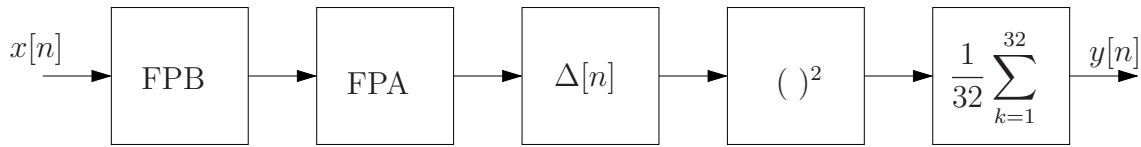


Figura 2.3: Diagrama de blocos do pré-processamento do ECG para a detecção das ondas R [HT86].

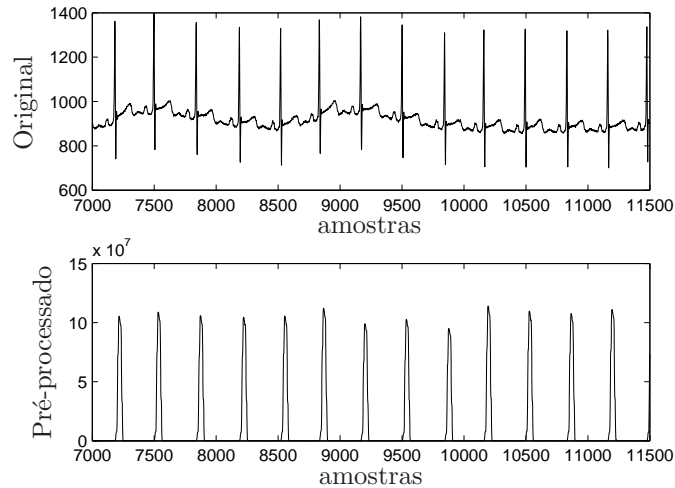


Figura 2.4: Trecho do ECG de arritmia 115 e do resultado do seu pré-processamento.

picos menor que o valor médio das durações dos batimentos, o menor pico é considerado um falso pico e é descartado. Os picos das ondas R são então determinados pela amostra de maior valor na janela que se inicia na amostra correspondente ao pico do sinal pré-processado e que possui duração de um quarto do tamanho médio dos batimentos.

2.4 Medidas de desempenho

Nos métodos de compactação a medida de desempenho é a taxa de compressão. Nos métodos de compressão os sinais reconstruídos após o processo de compressão podem apresentar distorções em relação aos sinais originais. Como o diagnóstico de cardiopatias é sensível à forma dos ECGs [Bra97], o método de compressão deve reduzir o tamanho do arquivo comprimido sem que o sinal resultante seja muito diferente do original.

Diversas medidas de distorção são propostas na literatura. Uma medida proposta em [ZCK00], denominada *Weighted Diagnostic Distortion* (WDD), é utilizada em alguns trabalhos de compressão de ECGs [ASAZA03], [ZCK00]. A medida WDD busca quantificar a preservação relativa do diagnóstico no sinal reconstruído e consiste na extração dos componentes de cada batimento dos sinais original e reconstruído e na comparação entre os valores dos componentes dos dois sinais. Entre as informações utilizadas para o cálculo do WDD estão o tamanho do

batimento, a duração do complexo QRS, a duração entre o início do complexo QRS e o final da onda T, duração da onda P, formato das ondas T, P e do segmento ST, amplitude das ondas P e T, etc. As informações obtidas tanto do ECG original quanto do ECG reconstruído são colocadas em vetores e a medida WDD é obtida por um cálculo ponderado da norma da diferença entre os dois vetores. A ponderação é feita pois algumas informações são consideradas mais importantes para o diagnóstico de uma cardiopatia.

O cálculo do WDD depende da aplicação de algoritmos de detecção dos componentes dos batimentos dos sinais original e reconstruído e, conseqüentemente, erros na detecção dos componentes podem causar um aumento artificial no valor da distorção que não foi causado pelo método de compressão. A própria detecção dos componentes de todos os batimentos nos dois sinais é dispendiosa e o cálculo do WDD pode requerer um tempo maior do que o próprio método de compressão. É necessária então uma medida de distorção de cálculo mais simplificada.

Uma medida de erro muito utilizada é o erro médio quadrático (*Root Mean Square Error - RMSE*) [AB92], [BESS00], [KP04], [KYL06], [SP96], dado por

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{\tau_n} (f[k] - \hat{f}[k])^2}{\tau_n}}, \quad (2.4)$$

sendo $f[n]$ o ECG original, $\hat{f}[n]$ o ECG reconstruído após o processo de compressão e τ_n o número de amostras no ECG. Nota-se, no entanto, que a medida RMSE apresenta um valor de distorção artificialmente menor para sinais de longa duração, fazendo com que a aplicação dos métodos de compressão seja privilegiada para tais sinais.

Outra medida de distorção muito utilizada é a medida chamada *Percent Root-mean-square Distortion* (PRD) [ABP09a], [SABP08], [HM04], [MYL02], [ASAZA03], [ZCK00], e é dada por

$$\text{PRD} = \sqrt{\frac{\sum_{k=1}^{\tau_n} (f[k] - \hat{f}[k])^2}{\sum_{k=1}^{\tau_n} (f[k] - \bar{f})^2}}, \quad (2.5)$$

sendo \bar{f} o valor médio ou valor DC do ECG original. Muitos autores não fazem a subtração do valor médio do ECG no denominador no cálculo do PRD [AASAZ00], [BESS00], [BLDF06], [CHY04], [CI98], [HM04], [IHP01], [KYL06], [NC93], [Raj02], enquanto outros subtraem o valor 1024 ao invés de subtrair o valor da média, dado que, nos bancos de dados de arritmia do MIT-BIH [Phy] o valor 1024 é somado em todos os sinais para que todas as amostras sejam positivas e facilitar o armazenamento dos dados [BMB07], [ZLP00], [WCCJ01]. A subtração de um valor menor do que o valor médio ou mesmo não realizar nenhuma subtração no cálculo do PRD causa uma diminuição artificial no valor do PRD e mascara a distorção do sinal resultante.

O PRD é uma das medidas mais utilizadas pois seu cálculo é muito simples. Se uma técnica de compressão baseada em transformações ortogonais é utilizada, o PRD pode ser calculado a partir da relação entre a energia de todos os coeficientes da transformação e a energia dos coeficientes retidos. Sejam $v_k[n]$ os elementos da base ortonormal utilizada na transformação do sinal $f[n]$ de forma que

$$f[n] = \sum_k c_k v_k[n], \quad \sum_k \sum_p v_k[n] v_p[n] = \begin{cases} 1 & k = p \\ 0 & k \neq p \end{cases}, \quad (2.6)$$

sendo c_k os coeficientes de transformação do sinal $f[n]$. O sinal reconstruído $\hat{f}[n]$ é obtido retendo-se apenas os coeficientes c_k mais significativos. Seja \mathcal{K} o conjunto dos índices dos coeficientes. O sinal reconstruído $\hat{f}[n]$ é escrito como

$$\hat{f}[n] = \sum_{k \in \mathcal{K}} c_k v_k[n]. \quad (2.7)$$

Considere $f[n]$ um sinal de média nula. A Equação (2.5) pode ser reescrita como

$$\text{PRD} = \sqrt{1 - \frac{\sum_k \hat{f}^2[k]}{\sum_k f^2[k]}} \quad (2.8)$$

pois

$$\sqrt{\frac{\sum_k (f[k] - \hat{f}[k])^2}{\sum_k f^2[k]}} \Rightarrow \sqrt{\frac{\sum_k \left(\sum_p c_p v_p[k] - \sum_{p \in \mathcal{K}} c_p v_p[k] \right)^2}{\sum_k f^2[k]}} \Rightarrow \sqrt{1 - \frac{\sum_k \hat{f}^2[k]}{\sum_k f^2[k]}}. \quad (2.9)$$

A energia do sinal recuperado está relacionada com a distorção resultante da compressão [SABP07b]. De acordo com o Teorema de Parseval [OWN97] a energia \mathcal{E} do sinal original e a energia $\hat{\mathcal{E}}$ do sinal reconstruído são dadas por

$$\mathcal{E} = \sum_k f^2[k] = \sum_k |c_k|^2, \quad \hat{\mathcal{E}} = \sum_k \hat{f}^2[k] = \sum_{k \in \mathcal{K}} |c_k|^2. \quad (2.10)$$

Assim, o PRD pode ser escrito em função dos coeficientes de transformação como

$$\text{PRD} = \sqrt{1 - \frac{\sum_{k \in \mathcal{K}} |c_k|^2}{\sum_k |c_k|^2}}. \quad (2.11)$$

O PRD é uma medida global da distorção do sinal, não enfatizando nenhuma informação local da distorção do ECG reconstruído. Uma medida denominada *Moving Average Percent-root-mean Square Distortion* (MAPRD) que permite caracterizar localmente a distorção do sinal pode ser definida por

$$\text{MAPRD}[n] = \sqrt{1 - \frac{\sum_{k=n}^{n+w} \hat{f}^2[k]}{\sum_{k=n}^{n+w} f^2[k]}}, \quad (2.12)$$

sendo w a janela de observação.

2.5 Conclusão

Neste capítulo foram apresentadas as características principais dos eletrocardiogramas (ECGs). Uma breve análise da *baseline* do ECG foi realizada em conjunto com a apresentação de um método de retirada da *baseline* utilizando a transformada DCT. O repositório que contém as bases de dados de sinais fisiológicos intitulado *PhysioBank* [Phy] foi também descrito, bem como um método para detecção dos batimentos cardíacos, baseado em [HT86], que apresenta melhorias em relação ao método original. Algumas medidas de distorção foram apresentadas, destacando-se as medidas PRD (*Percent Root-mean-square Distortion*) e MAPRD (*Moving Average Percent Root-mean-square Distortion*), que consiste no cálculo do PRD em janelas de duração fixa e permite uma análise local do resultado da compressão.

Wavelets

3.1 Introdução

As *wavelets* têm sido utilizadas em diversas aplicações, como por exemplo na análise e compressão de sinais [SABP07c], [SABP07a], [Say06], [BMB⁺03], [Bra96], [CI98], [IHP01], [IP00], [GJP02] [ZLP00], [MYL02], [Raj02], [RS97], [GSK⁺06]. Os coeficientes resultantes da aplicação da transformada *wavelet* em um dado sinal são de fato os coeficientes de projeção do sinal na base formada por funções *wavelet* e por funções escala, que são funções complementares às funções *wavelet* [BGG98]. A existência de infinitas funções *wavelet* que podem gerar uma base de representação para o sinal faz com que as *wavelets* sejam atraentes principalmente em sistemas adaptativos, em que a escolha da função a ser utilizada depende da aplicação, do sinal ou de propriedades que se desejam observar. No caso específico da compressão de sinais é importante que a função *wavelet* utilizada seja adaptada ao sinal, pois neste caso a energia do sinal se concentra em alguns poucos coeficientes de representação e a qualidade da compressão é melhorada [SABP07c], [SABP07a]. O objetivo do presente capítulo é apresentar as principais propriedades das funções *wavelet* e desenvolver um método que adapte a *wavelet* ao sinal a ser comprimido.

O capítulo está organizado da seguinte forma. Na Seção 3.2 são apresentadas as funções escala e as funções *wavelet*, suas principais propriedades e resultados, os coeficientes escala e *wavelet* que são utilizados para representar as respectivas funções e uma maneira de se obter os coeficientes de representação do sinal nos subespaços escala e *wavelet* utilizando filtros digitais. Um procedimento de parametrização dos coeficientes escala e *wavelet* é apresentado na Seção 3.3, de forma que os parâmetros resultantes do procedimento sejam irrestritos. Na Seção 3.4 é apresentado um procedimento de otimização cujo objetivo é encontrar a função *wavelet* que resulta na melhor compressão de um dado sinal. A conclusão do capítulo é realizada na Seção 3.5.

3.2 Funções escala e *Wavelets*

3.2.1 Funções escala

Toda função real não-nula $\varphi(2^\eta x)$ que pode ser expressa em termos de uma soma ponderada de deslocamentos inteiros da função $\varphi(2^{\eta+1}x)$ como

$$\varphi(2^\eta x) = \sum_k \ell[k] \sqrt{2^{\eta+1}} \varphi(2^{\eta+1}x - k), \quad k, \eta \in \mathbb{Z}, \quad x, \ell[k] \in \mathbb{R} \quad (3.1)$$

é uma função escala, sendo $\ell[n]$ os coeficientes escala e η o nível de resolução da função escala. Particularmente, para $\eta = 0$,

$$\varphi(x) = \sum_k \ell[k] \sqrt{2} \varphi(2x - k), \quad k, \eta \in \mathbb{Z}, \quad x, \ell[k] \in \mathbb{R}. \quad (3.2)$$

Normalmente as funções escala são construídas de forma a apresentar norma unitária, ou seja,

$$\int_{-\infty}^{+\infty} \varphi^2(x) dx = 1. \quad (3.3)$$

O termo $\sqrt{2^{\eta+1}}$ na Equação (3.1) garante que a norma da função escala permanece constante independente da resolução η .

Propriedade 1 Se $\varphi(x)$ satisfaz a Equação (3.2), então

$$\int_{-\infty}^{+\infty} \varphi(x) dx > 0. \quad (3.4)$$

Se a função escala $\varphi(x)$ possui norma unitária, tem-se que

$$\int_{-\infty}^{+\infty} \varphi(x) dx = 1. \quad (3.5)$$

□

Duas funções escala em um mesmo nível de resolução são ditas ortonormais ao deslocamento se

$$\langle \varphi(2^\eta x) \varphi(2^\eta x + k) \rangle = \delta[k] = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}. \quad (3.6)$$

Note que a ortonormalidade não é uma condição necessária para que uma função $\varphi(x)$ seja uma função escala. Existem funções escala que são biortogonais e funções escala que não são nem ortogonais nem biortogonais ao deslocamento [Dau92]. Neste texto, as funções escala são ortogonais.

Um exemplo simples de função escala é a função de Haar, que é um retângulo de largura e altura unitárias e satisfaz

$$\varphi(x) = \varphi(2x) + \varphi(2x - 1), \quad (3.7)$$

com $\ell[0] = \ell[1] = 1/\sqrt{2}$.

A seguinte aproximação pode ser considerada

$$\lim_{\eta \rightarrow +\infty} \varphi(2^\eta x) = \delta(x), \quad (3.8)$$

sendo $\delta(x)$ a função delta de Dirac [OWN97]. De fato

$$\int_{-\infty}^{+\infty} \delta(x) dx = 1 \quad (3.9)$$

e, para $\eta \rightarrow +\infty$, a largura de ambas as funções tende a zero. Note que quando $\eta \rightarrow +\infty$ o deslocamento da função $\varphi(2^\eta x)$ é infinitesimal. O parâmetro η é chamado de resolução da função escala pois, quando se utiliza o conjunto formado pelos deslocamentos inteiros de uma função escala em uma resolução η para representar uma função qualquer, aumentar o valor de η implica em aproximar ainda mais a função de um delta de Dirac com deslocamentos infinitesimais, resultando em uma melhor representação da função.

As funções escala estão contidas no conjunto Ω_φ dado por

$$\Omega_\varphi = [0, m - 1], \quad m \in \mathbb{Z} \quad (3.10)$$

e são denominadas funções com suporte compacto. O valor m é o tamanho da função escala e também corresponde ao número de coeficientes escala $\ell[n]$ correspondentes à função escala. A função escala de Haar, por exemplo, está contida em $[0, 1]$ e é descrita por $m = 2$ coeficientes escala.

Toda função escala de tamanho m pode ser unicamente definida por m coeficientes $\ell[n]$ e vice-versa. Os coeficientes $\ell[n]$ podem ser obtidos a partir da função escala $\varphi(x)$ por

$$\ell[n] = \sqrt{2} \int_{-\infty}^{+\infty} \varphi(x) \varphi(2x - n) dx. \quad (3.11)$$

Existem diversas maneiras de se obter a função escala a partir dos coeficientes $\ell[n]$ [BGG98]. O algoritmo de aproximações sucessivas consiste em definir inicialmente uma função arbitrária $\varphi_{(0)}(x)$ de norma unitária e aplicar recursivamente a equação

$$\varphi_{(k+1)}(x) = \sum_n \ell[n] \sqrt{2} \varphi_{(k)}(2x - n), \quad (3.12)$$

baseada na Equação (3.2), até que o algoritmo convirja para uma função $\varphi(x)$. O algoritmo de expansões diádicas define um vetor composto pelos valores da função escala nos pontos $k, k = [0, 1, \dots, m - 1]$ e encontra tais valores pela resolução de um sistema de equações montado a partir da Equação (3.2). Na segunda iteração são encontrados os valores da função escala nos pontos $k, k = [1/2, 3/2, \dots, m - 3/2]$ pela resolução de um sistema de equações também montado a partir da Equação (3.2). O procedimento continua até que se encontre um número suficiente de amostras que defina a função escala. Os dois métodos apresentados são os mais comuns; outros podem ser encontrados em [BGG98].

Os coeficientes escala $\ell[n]$ apresentam uma série de propriedades muito úteis no estudo das funções escala e *wavelet* [BGG98].

Propriedade 2 Se $\varphi(x) \in L^1$ é uma função escala, então

$$\sum_k \ell[k] = \sqrt{2}. \quad (3.13)$$

□

Propriedade 3 Se $\varphi(x)$ é uma função escala com

$$\int \varphi(x)dx = 1 \quad \text{e} \quad \langle \varphi(x)\varphi(x-k) \rangle = \delta[k], \quad (3.14)$$

então

$$\sum_k \ell[2k] = \sum_k \ell[2k+1] = \frac{1}{\sqrt{2}}. \quad (3.15)$$

□

Propriedade 4 Se $\varphi(x)$ é uma função escala e se os deslocamentos inteiros de $\varphi(x)$ são ortogonais conforme descrito na Equação (3.6), então

$$\sum_k \ell[k]\ell[k-2n] = \delta[n]. \quad (3.16)$$

□

Teorema 1 Se $\ell[n]$ possui suporte compacto ou decai rápido o suficiente e se as condições das Propriedades 2 e 4 são satisfeitas, então $\varphi(x) \in L^2$ existe e é uma função escala ortogonal.

O Teorema 1 [Law90] apresenta condições suficientes para que a função $\varphi(x)$ seja uma função escala e faz com que as Propriedades 2 e 4 sejam as principais propriedades dos coeficientes escala.

3.2.2 *Wavelets*

O conjunto composto pelos deslocamentos inteiros de uma função escala $\varphi(x)$ é uma base para o subespaço V_0 denominado subespaço escala. Genericamente, denomina-se V_η o subespaço cuja base é composta pelos deslocamentos inteiros da função escala $\varphi(2^\eta x)$.

Propriedade 5 Se $\varphi(2^\eta x)$ é uma função escala, então $\varphi(2^\eta x)$ gera o subespaço escala V_η e, como consequência da Equação (3.1), tem-se que

$$V_\eta \subset V_{\eta+1}. \quad (3.17)$$

Expandindo-se a Equação (3.17):

$$\{0\} \subset \dots V_{-1} \subset V_0 \subset V_1 \subset \dots \subset L^2. \quad (3.18)$$

□

Conforme o valor de η aumenta, a largura da função escala diminui e conseqüentemente qualquer função pode ser melhor aproximada no subespaço escala de resolução η . O número de funções $\varphi(2^\eta x - k)$ necessárias para aproximar a função também aumenta. No limite $\eta \rightarrow +\infty$, a largura da função escala $\varphi(2^\eta x)$ tende a zero e o erro de representação de uma função no subespaço escala V_η tende a zero.

O subespaço W é denominado de subespaço *wavelet* e é definido como o complemento ortogonal do subespaço escala em uma resolução η , ou seja,

$$V_{\eta+1} = V_{\eta} \oplus W_{\eta}, \quad (3.19)$$

resultando em

$$V_{\eta+2} = V_{\eta+1} \oplus W_{\eta+1} = V_{\eta} \oplus W_{\eta} \oplus W_{\eta+1} \quad (3.20)$$

e assim sucessivamente

$$L^2 = V_{\eta} \oplus W_{\eta} \oplus W_{\eta+1} \oplus \cdots \quad (3.21)$$

A Equação (3.21) apresenta a análise de multirresolução: qualquer função quadraticamente integrável pode ser representada projetando-se a função em um subespaço escala com uma resolução arbitrária η e em todos os subespaços *wavelet* com resolução maior ou igual a η .

De acordo com a Equação (3.19), o subespaço *wavelet* de resolução η está contido no subespaço escala na resolução $\eta + 1$. Portanto

$$\psi(2^n x) = \sum_k h[k] \sqrt{2^{\eta+1}} \varphi(2^{\eta+1} x - k), \quad k \in \mathbb{Z}, \quad (3.22)$$

sendo $\psi(x)$ a função *wavelet* e $h[n]$ os coeficientes *wavelet*. Note que a base do subespaço *wavelet* na resolução η é composta pelos deslocamentos inteiros da função escala $\varphi(2^{\eta+1} x)$. A função *wavelet* possui a mesma largura m da função escala e o número de coeficientes *wavelet* utilizados na relação de recorrência mostrada na Equação (3.22) é também igual a m .

Propriedade 6 Se $\psi(x)$ é uma função *wavelet*, então

$$\int_{-\infty}^{+\infty} \psi(x) dx = 0. \quad (3.23)$$

□

Propriedade 7 O subespaço escala V_{η} é ortogonal ao subespaço *wavelet*, ou seja,

$$\langle \varphi(2^n x - k) \psi(2^n x - p) \rangle = 0, \quad \forall k, p \in \mathbb{Z}. \quad (3.24)$$

Uma função *wavelet* é ortogonal a uma função escala independentemente da resolução ou do deslocamento considerado, isto é,

$$\langle \varphi(2^p x - k) \psi(2^r x - n) \rangle = 0. \quad (3.25)$$

□

Propriedade 8 A função *wavelet* é ortogonal com respeito à resolução e ao deslocamento:

$$\langle \psi(2^p x - k) \psi(2^r x - n) \rangle = E \delta[p - r] \delta[k - n], \quad (3.26)$$

sendo E a norma da função *wavelet*. Se $E = 1$, então a *wavelet* é ortonormal.

□

Propriedade 9 Os coeficientes *wavelet* podem ser obtidos a partir da função *wavelet* e da respectiva função escala por

$$h[n] = \sqrt{2} \int_{-\infty}^{+\infty} \psi(x) \varphi(2x - n) dx. \quad (3.27)$$

□

Propriedade 10 Os coeficientes $h[n]$ de uma função *wavelet* relacionam-se com os coeficientes $\ell[n]$ da função escala por

$$h[n] = (-1)^n \ell[m - n - 1], \quad n \in [0, m - 1]. \quad (3.28)$$

□

Propriedade 11 Se os coeficientes escala $\ell[n]$ satisfazem as condições das Propriedades 2 e 4, então

$$\langle \varphi(2^p x - k) \psi(2^r x - n) \rangle = 0 \quad \Leftrightarrow \quad \sum_k \ell[k] h[k - 2n] = 0. \quad (3.29)$$

□

Propriedade 12 A soma dos coeficientes *wavelet* $h[n]$ é igual a zero, ou seja,

$$\sum_{k=0}^{m-1} h[k] = 0. \quad (3.30)$$

□

3.2.3 Coeficientes de projeção e bancos de filtros

Dada uma função $f(x)$, os coeficientes de projeção $a_\eta[n]$ da função no subespaço V_η são dados por

$$a_\eta[k] = \frac{\langle f(x) \varphi(2^\eta x - k) \rangle}{\langle \varphi^2(2^\eta x) \rangle} \quad (3.31)$$

e os coeficientes de projeção $b_\eta[n]$ da função no subespaço W_η são dados por

$$b_\eta[k] = \frac{\langle f(x) \psi(2^\eta x - k) \rangle}{\langle \psi^2(2^\eta x) \rangle}. \quad (3.32)$$

Como as funções escala e *wavelet* são descritas pelos coeficientes escala $\ell[n]$ e *wavelet* $h[n]$, é possível obter os coeficientes de projeção por [GS97]

$$a_{\eta-1}[n] = \sum_p \ell[p - 2n] a_\eta[p] = \{\ell[-n] * a_\eta[n]\}_{\downarrow 2}, \quad (3.33)$$

$$b_{\eta-1}[k] = \sum_p h[p - 2k] a_\eta[p] = \{h[-n] * a_\eta[n]\}_{\downarrow 2} \quad (3.34)$$

e $a_\eta[n] = f[n]$, sendo $f[n]$ as amostras da função $f(x)$.

As operações apresentadas nas Equações (3.33) e (3.34) formam o procedimento de decomposição. Dados os coeficientes $a_\eta[n] = f[n]$ de projeção do sinal $f(x)$ no subespaço escala de resolução η é possível encontrar os coeficientes de representação do sinal nos subespaços escala e *wavelet* com resolução $\eta - 1$, conforme mostrado na Figura 3.1. O valor de η é igual ao número de níveis de decomposição.

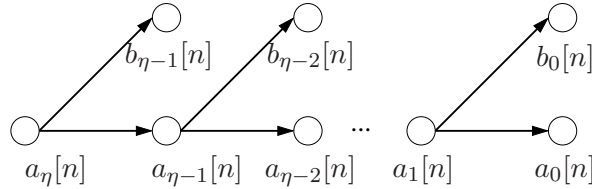


Figura 3.1: Procedimento de decomposição.

Note nas Equações (3.33) e (3.34) que os coeficientes de decomposição em uma resolução são obtidos pela convolução dos coeficientes de decomposição do subespaço escala de resolução maior com as sequências discretas $\ell[-n]$ ou $h[-n]$ e pela subamostragem do resultado. O procedimento de reconstrução é realizado pelos passos inversos do procedimento de decomposição: os coeficientes de decomposição de resolução mais baixa são sobreamostrados e convoluídos com as sequências discretas $\ell[n]$ e $h[n]$ para gerar os coeficientes de decomposição do subespaço escala de maior resolução, conforme a Equação (3.35) e a ilustração da Figura 3.2.

$$a_{\eta+1}[n] = \ell[n] * \{a_\eta[n]\}_{\uparrow 2} + h[n] * \{b_\eta[n]\}_{\uparrow 2}. \quad (3.35)$$

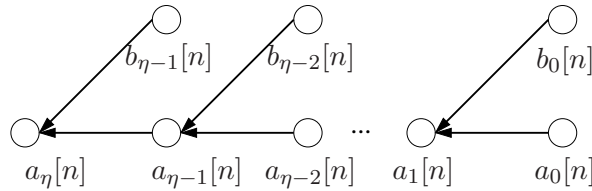


Figura 3.2: Procedimento de reconstrução.

Note que a convolução dos coeficientes $a_\eta[n]$ com as sequências $\ell[-n]$ e $h[-n]$ no procedimento de decomposição equivale a aplicar os coeficientes nos filtros digitais cujas respostas ao impulso são iguais a $\ell[-n]$ e $h[-n]$. Similarmente, a convolução dos coeficientes $a_\eta[n]$ e $b_\eta[n]$ com as sequências $\ell[n]$ e $h[n]$ equivale a aplicar os coeficientes nos filtros digitais cuja resposta ao impulso são iguais a $\ell[n]$ e $h[n]$. De fato, considerar que as sequências discretas $\ell[n]$ e $h[n]$ são respostas ao impulso de filtros discretos destaca propriedades interessantes na análise das funções escala e *wavelet*.

Sejam $L(\omega)$ e $H(\omega)$ as transformadas discretas de Fourier dos filtros digitais $\ell[n]$ e $h[n]$, respectivamente, dadas por [OS99]

$$L(\omega) = \sum_{k=0}^{m-1} \ell[k] \exp(-j\omega k) \quad , \quad H(\omega) = \sum_{k=0}^{m-1} h[k] \exp(-j\omega k). \quad (3.36)$$

Propriedade 13

$$\sum_{k=0}^{m-1} \ell[k] = L(\omega)|_{\omega=0} = \sqrt{2}. \quad (3.37)$$

□

Propriedade 14

$$\sum_{k=0}^{m-1} (-1)^k \ell[k] = L(\omega)|_{\omega=\pi} = 0. \quad (3.38)$$

□

Propriedade 15

$$\sum_{k=0}^{m-1} h[k] = H(\omega)|_{\omega=0} = 0. \quad (3.39)$$

□

Propriedade 16

$$\sum_{k=0}^{m-1} (-1)^k h[k] = H(\omega)|_{\omega=\pi} = \sqrt{2}. \quad (3.40)$$

□

De acordo com as Propriedades 13 e 14, o filtro digital correspondente aos coeficientes escala $\ell[n]$ é um filtro passa-baixas e, de acordo com as Propriedades 15 e 16, o filtro digital correspondente aos coeficientes *wavelet* $h[n]$ é um filtro passa-altas. Consequentemente, os coeficientes de projeção no subespaço escala correspondem aos componentes de menor frequência enquanto que os coeficientes de projeção no subespaço *wavelet* de mais alta resolução correspondem aos componentes de maior frequência. O valor DC de uma função, por exemplo, é um componente de baixa frequência e afeta os coeficientes de projeção no subespaço escala: um aumento no valor DC do sinal ocasiona um aumento nos valores dos coeficientes de projeção no subespaço escala, como se um valor DC fosse somado aos próprios coeficientes.

Os procedimentos de decomposição e de reconstrução utilizam os coeficientes de projeção do sinal $f(x)$, porém é necessário primeiro obter os coeficientes de projeção do sinal no subespaço escala de resolução mais alta para depois aplicar o procedimento de decomposição, assim como o procedimento de reconstrução retorna, no último passo, os coeficientes de projeção no subespaço escala de resolução mais alta.

3.3 Parametrização

A Tabela 3.1 mostra os coeficientes escala de alguns pares de funções escala e *wavelet* clássicos. Os coeficientes *wavelet* podem ser obtidos dos coeficientes escala pela Propriedade 10. Cada um dos pares clássicos de funções escala e *wavelet* foram obtidos de forma a apresentarem alguma característica importante ou então para que fossem utilizadas para algum fim específico. Por exemplo, os pares de funções de Daubechies de tamanho m foram construídos de forma que as funções apresentassem $m/2$ momentos nulos [Dau92], sendo o k -ésimo momento do par de funções escala e *wavelet* dado por

$$\mu[k] = \sum_{p=0}^{m-1} (-1)^p \ell[p] p^k. \quad (3.41)$$

Tabela 3.1: Coeficientes escala $\ell[n]$ ortogonais que geram alguns dos pares de funções escala e *wavelet* clássicos.

Nome	Coeficientes escala $\ell[n]$
Haar	[0.701 0.701]
Daubechies-2	[0.483 0.836 0.224 -0.129]
Daubechies-3	[0.333 0.807 0.460 -0.135 -0.085 0.035]
Coiflets-2	[0.016 -0.041 -0.067 0.386 0.813 0.417 -0.076 -0.059 0.024 0.006 -0.002 -0.001]

As *wavelets* de Daubechies são nomeadas nesta dissertação de acordo com o número de momentos nulos da função.

Em compressão de sinais, encontrar um par de funções escala e *wavelet* cujos formatos sejam parecidos com o formato do sinal a ser comprimido aumenta a concentração da energia do sinal em alguns poucos coeficientes de decomposição e pode aumentar a taxa de compressão. É importante, portanto, desenvolver um método para obter sequências discretas $\ell[n]$ e $h[n]$ que satisfaçam as propriedades necessárias para que possam ser utilizadas como coeficientes escala e *wavelet* [GS97], [ZT93].

De acordo com o Teorema 1, uma sequência real $\ell[n]$ de tamanho m é uma sequência válida de coeficientes escala se as Propriedades 2 e 4 forem satisfeitas. Tais propriedades podem ser vistas como restrições para um problema cujo objetivo é encontrar um conjunto de coeficientes escala que apresente propriedades relevantes para a aplicação em questão. A Propriedade 2 gera uma restrição e a Propriedade 4 gera $m/2$ restrições. Restam portanto γ graus de liberdade na escolha dos coeficientes escala, com γ dado por

$$\gamma = \frac{m}{2} - 1. \quad (3.42)$$

Ao invés de utilizar diretamente as restrições é possível parametrizar o conjunto de coeficientes escala de modo que as restrições estejam embutidas na parametrização, resultando em uma expressão com γ parâmetros.

Sejam $L(z)$ e $H(z)$ as transformadas unilaterais Z das sequências $\ell[n]$ e $h[n]$ dadas por

$$L(z) = \sum_{k=0}^{m-1} \ell[k]z^{-k}, \quad H(z) = \sum_{k=0}^{m-1} h[k]z^{-k}. \quad (3.43)$$

Definindo

$$E_{\text{par}}(z) = \sum_k \ell[2k]z^{-k} \quad (3.44)$$

e

$$E'_{\text{ímpar}}(z) = \sum_k \ell[2k+1]z^{-k}, \quad (3.45)$$

a Equação (3.43) pode ser expressa na forma

$$L(z) = E_{\text{par}}(z^2) + z^{-1}E'_{\text{ímpar}}(z^2) \quad (3.46)$$

que é conhecida como representação polifásica e que separa a transformadas em componentes de fase par e de fase ímpar.

De acordo com as Propriedades 13 e 14 o filtro $L(z)$ é um filtro passa-baixas, e de acordo com as Propriedades 15 e 16 o filtro $H(z)$ é um filtro passa-altas. A relação entre $\ell[n]$ e $h[n]$ mostrada na Propriedade 10 permite que ambos os filtros sejam considerados filtros QMF (*Quadrature Mirror Filters*) [GS97], o que significa que o espectro de um sinal pode ser separado em duas sub-bandas pela aplicação de ambos os filtros e, sob certas circunstâncias, o sinal pode ser reconstruído sem erros.

Os filtros $L(z)$ e $H(z)$ podem ser descritos de maneira conjunta utilizando uma notação matricial

$$\begin{bmatrix} L(z) \\ z^{-2(2m-1)}H(z) \end{bmatrix} = E(z^2) \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \quad (3.47)$$

com

$$E(z) = \begin{bmatrix} E_{\text{par},L}(z) & E_{\text{ímpar},L}(z) \\ E_{\text{par},H}(z) & E_{\text{ímpar},H}(z) \end{bmatrix}, \quad (3.48)$$

sendo $E_{\text{par},L}(z)$ a fase par do filtro $L(z)$.

Uma matriz de transferência $E(z)$ de um filtro FIR (*Finite Impulse Response*) causal de grau γ permite a reconstrução perfeita do sinal de entrada se e somente se a matriz pode ser expressa na forma [Vai90]

$$E(z) = V_\gamma(z)V_{\gamma-1}(z) \cdots V_1(z)V_0 \quad (3.49)$$

sendo V_k matrizes da forma

$$V_k(z) = I + (z^{-1} - 1)v_kv'_k, \quad \text{para } 1 \leq k \leq \gamma - 1 \quad (3.50)$$

e v_k é um vetor real de norma unitária. Em particular, não há perda de generalidade em assumir que v_k é da forma

$$v_k = \begin{bmatrix} \cos(\theta_k) \\ \text{sen}(\theta_k) \end{bmatrix}. \quad (3.51)$$

A matriz V_0 é dada por [GS97]

$$V_0 = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (3.52)$$

O grau γ da matriz $E(z)$ é igual ao número de atrasos (z^{-1}) necessários para implementar o filtro, sendo que no caso das *wavelets* o valor γ é igual ao número de graus de liberdade na escolha da função escala dado pela Equação (3.42). A definição das matrizes $V_k(z)$ conforme mostrado na Equação (3.50) garante que os coeficientes escala obtidos satisfaçam a condição imposta pela Propriedade 4, e a definição da matriz V_0 como mostrado na Equação (3.52) garante que os coeficientes satisfaçam a condição imposta pela Propriedade 2.

Portanto, dado um vetor de γ parâmetros θ , a função $L(z)$ é encontrada por meio da Equação (3.47), com $E(z)$ dado pela Equação (3.49), V_k dado pela Equação (3.50) e V_0 dado pela Equação (3.52). Note que não há restrições nos parâmetros e que pode-se determinar o espaço de busca dos parâmetros θ com $\theta_k \in [0, \pi]$, $k = 1, \dots, \gamma$, pois os coeficientes escala repetem-se para θ com período π .

3.4 Adaptação ao sinal

Os parâmetros θ estão contidos em um domínio limitado (devido à sua característica de periodicidade) e não têm restrições para gerar sequências com as propriedades de coeficientes

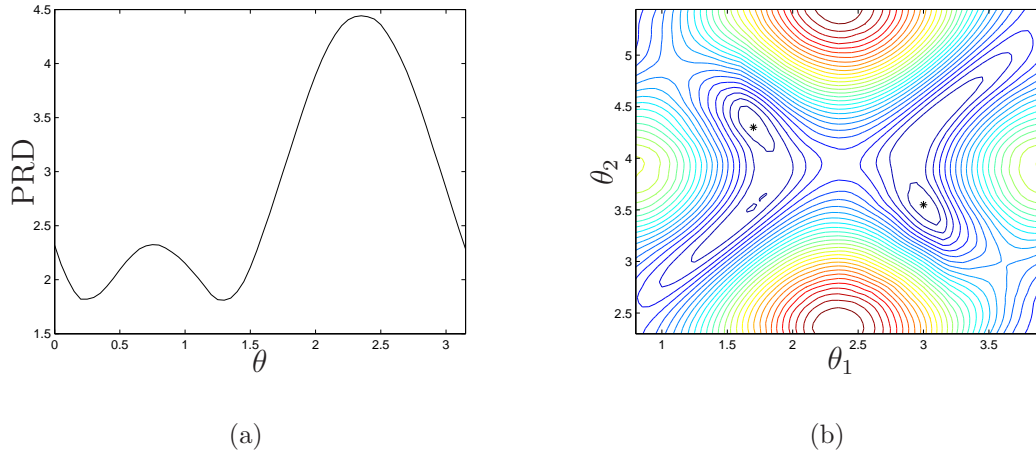


Figura 3.3: Distorção $\text{PRD}(\theta)$ para (a) $m = 4$ (b) $m = 6$. O sinal utilizado é o sinal de número 115 da base de arritmia do MIT-BIH. Na figura (b) os asteriscos representam as regiões de mínimo local de $\text{PRD}(\theta)$.

escala e *wavelet*. Tais características facilitam a busca por um par de funções escala e *wavelet* que satisfaça alguma propriedade desejada. Na compressão por transformadas, a taxa de compressão é maior se a energia do sinal se concentrar em alguns coeficientes da transformada [Sal07].

Uma das vantagens das transformadas *wavelet* na compressão de sinais é a utilização de uma base que não é fixa mas que pode ser descrita com poucos parâmetros. Para encontrar um par de funções escala e *wavelet* que gere uma base para a compressão, o seguinte problema de otimização é proposto

$$\min_{\theta} \text{PRD}(\theta), \quad (3.53)$$

sendo $\text{PRD}(\theta)$ a medida de distorção apresentada na Seção 2.4.

O problema de otimização é não convexo e apresenta diversos mínimos locais, conforme mostra a Figura 3.3 considerando funções escala com tamanho $m = 4$ ($\gamma = 1$ parâmetro θ) e $m = 6$ ($\gamma = 2$ parâmetros θ). O sinal utilizado é o sinal 115 da base de dados de arritmia do MIT-BIH [GAG⁺00], e os valores de distorção foram obtidos pela aplicação do procedimento de decomposição, retenção de um número fixo de coeficientes de decomposição e reconstrução do sinal.

Considerando que o ECG original $f[n]$ possui valor médio nulo, o PRD pode ser calculado da seguinte forma:

$$\text{PRD}(\theta) = \left(1 - \sum_k \hat{f}^2[k] \left(\sum_k f^2[k] \right)^{-1} \right)^{1/2}, \quad (3.54)$$

sendo $\hat{f}[n]$ o sinal reconstruído. Porém, não é necessário reconstruir o sinal após a retenção dos coeficientes significativos, uma vez que a medida de distorção $\text{PRD}(\theta)$ pode ser diretamente calculada a partir dos coeficientes de decomposição. Como as bases escala e *wavelet* são ortogonais, então o Teorema de Parseval fornece [OWN97]

$$\mathcal{E} = \sum_k f^2[k] = \sum_k |c_k|^2, \quad \hat{\mathcal{E}} = \sum_k \hat{f}^2[k] = \sum_{k \in \mathcal{K}} |c_k|^2 \quad (3.55)$$

sendo \mathcal{E} a energia do sinal original, $\hat{\mathcal{E}}$ a energia do sinal recuperado, c_k os coeficientes de decomposição e \mathcal{K} o subconjunto dos índices dos coeficientes significativos, resultando em

$$\text{PRD}(\theta) = \left(1 - \sum_{k \in \mathcal{K}} |c_k|^2 \left(\sum_k |c_k|^2 \right)^{-1} \right)^{1/2}. \quad (3.56)$$

O vetor c_k é arranjado da seguinte maneira:

$$\begin{aligned} c[n] &= a_0[n], \quad 1 \leq n \leq m_0, \\ c[\tau] &= b_k[n], \quad 1 \leq n \leq m_k, \quad \tau = n + 1 + \sum_{p=0}^k m_p, \end{aligned} \quad (3.57)$$

sendo m_k o número de coeficientes de decomposição no nível k .

O cálculo do valor do $\text{PRD}(\theta)$ no problema de otimização descrito pela Equação (3.53) é realizado pelos seguintes passos

1. Escolha uma taxa de retenção ρ ;
2. Encontre, para um vetor de parâmetros θ , os coeficientes dos filtros escala e *wavelet*;
3. Compute a resolução η por

$$\eta = \left\lfloor \log_2 \left(\frac{2(\beta - 1)}{m - 1} \right) \right\rfloor, \quad (3.58)$$

sendo β o número médio de amostras de um batimento do ECG. O valor de η é o número de níveis de decomposição que produz uma função escala cujo número de amostras κ seja aproximadamente igual a β . O valor de κ é dado por

$$\kappa = (m - 1)2^{\eta-1} + 1; \quad (3.59)$$

4. Decomponha o sinal em η subespaços conforme mostrado nas Equações (3.33) e (3.34), construa o vetor c_n conforme especificado na Equação (3.57) e retenha aproximadamente τ_c/ρ coeficientes, sendo τ_c o número total de coeficientes de decomposição obtidos. Os coeficientes não retidos são substituídos por zero;
5. Compute o $\text{PRD}(\theta)$ pela Equação (3.56).

O procedimento de otimização é repetido para diversos pontos iniciais e armazenam-se os parâmetros que resultam na menor distorção. O valor m do suporte das funções escala e *wavelet* resultantes da otimização, e conseqüentemente o número γ de parâmetros θ considerados, é fixo durante todo o procedimento. A Tabela 3.2 mostra os valores de PRD obtidos após a aplicação de um procedimento de compressão de cinco eletrocardiogramas considerando vários valores de m . O procedimento aplicado consiste na resolução do problema de otimização mostrado na Equação (3.53), cálculo dos coeficientes utilizando o procedimento de decomposição *wavelet*, retenção de 25% dos coeficientes mais significativos e recuperação do sinal com os coeficientes retidos. Note que os valores do PRD não se alteram significativamente para valores de m acima de 10. Assim, na implementação do procedimento de otimização nos métodos de compressão é utilizado o valor $m = 10$.

Tabela 3.2: Valores de PRD resultantes da compressão de cinco eletrocardiogramas variando o suporte m das funções *wavelet* utilizadas.

Sinal	$m = 4$	$m = 6$	$m = 8$	$m = 10$	$m = 12$	$m = 14$	$m = 16$
107	1.12	0.96	0.92	0.91	0.90	0.89	0.90
109	1.85	1.65	1.54	1.47	1.48	1.48	1.47
115	1.70	1.52	1.42	1.38	1.36	1.34	1.33
117	3.15	2.81	2.63	2.54	2.51	2.49	2.47
222	5.40	5.08	4.85	4.80	4.76	4.73	4.74

3.5 Conclusão

Neste capítulo foram apresentadas as principais propriedades e teoremas referentes à transformada *wavelet*. Os coeficientes da transformada *wavelet* são de fato os coeficientes provenientes do chamado procedimento de decomposição *wavelet*, que consiste na projeção do sinal original em diversos subespaços *wavelet* e no subespaço escala, este último gerado pelos deslocamentos inteiros da chamada função escala, que é uma função complementar à função *wavelet*. Tanto a função escala quanto a função *wavelet* podem ser descritas pelas sequências discretas formadas pelos coeficientes escala e coeficientes *wavelet*. Tais coeficientes podem também ser vistos como filtros digitais, sendo que o processo de decomposição se reduz à aplicação do sinal original nestes filtros digitais, gerando na saída os coeficientes de decomposição. Foi apresentado também um método de parametrização dos coeficientes escala e *wavelet* de forma que os parâmetros não tenham restrições e estejam contidos em um domínio finito. O procedimento de parametrização possibilita a busca, utilizando parâmetros irrestritos, de pares de funções escala e *wavelet* que satisfaçam alguma propriedade desejada. Um procedimento de otimização para encontrar o par de funções escala e *wavelet* adequado à compressão de eletrocardiogramas foi também proposto.

Compressão por Adaptação da *Wavelet*

4.1 Introdução

Muitos métodos de compressão de ECGs que utilizam a transformada *wavelet* têm sido desenvolvidos. Em [ZLP00], Lu & Pearlman apresentam o método SPIHT (*Set Partitioning in Hierarchical Trees*), que usa estruturas de árvores para codificar tanto os valores dos coeficientes significativos quanto suas posições em uma mesma *string* de *bits*. Rajoub [Raj02] propõe um método de compressão que faz um pré-processamento no sinal e, após a aplicação da transformada *wavelet* no sinal pré-processado, escolhe quais coeficientes serão retidos baseado no cálculo da energia dos coeficientes de cada subespaço *wavelet*. Os coeficientes retidos são então quantizados e codificados. O método desenvolvido por Benzid *et al.* [BMB07] define quais coeficientes serão retidos baseado em um algoritmo de biseção e codifica tanto os coeficientes retidos quanto suas posições usando um procedimento denominado *Two Role Encoder* (TRE). Nos três métodos citados é possível desenvolver mecanismos para que a distorção do sinal reconstruído possa ser controlado pelo usuário: no método SPIHT a distorção é controlada pela interrupção da *string* de saída em um certo ponto, no método de Rajoub a distorção é controlada definindo uma taxa de retenção dos coeficientes dos subespaços *wavelet* e no método TRE a distorção é controlada pela precisão do algoritmo de biseção.

Neste capítulo é apresentado o método de compressão denominado *Run Length Encoding Adaptativo* (RLEA). O método consiste na obtenção da função *wavelet* que minimiza a distorção do sinal comprimido, aplicação da transformada *wavelet* utilizando a função obtida e retenção dos coeficientes de projeção mais significativos. A escolha dos coeficientes significativos é feita por um algoritmo de biseção, como em [BMB07]. Os coeficientes retidos são quantizados utilizando um procedimento de quantização adaptativa e tanto os coeficientes quantizados quanto o *bitmap* que informa a posição dos coeficientes retidos são codificados pela aplicação da técnica de compactação denominada *Run Length Encoding*, porém utilizando palavras-código cujos tamanhos são adaptados para cada subespaço *wavelet* considerado. Assim como no método apresentado em [BMB07], a distorção do sinal reconstruído é controlada pelo algoritmo de biseção.

O capítulo está organizado da seguinte forma. A seção 4.2 apresenta o método de compressão RLEA. Na Seção 4.3 é apresentado um exemplo que ilustra as principais vantagens do método RLEA, assim como um conjunto de experimentos numéricos realizados para comparar o método RLEA com os métodos TRE, SPIHT e de Rajoub. A Seção 4.4 conclui o capítulo.

4.2 Método de compressão RLEA

O primeiro passo do método consiste na retirada do valor médio do sinal, uma vez que a parte importante em um ECG é a variação temporal da forma de onda e não o seu nível DC [AAF03]. O valor médio do sinal também afeta o cálculo do PRD do sinal reconstruído após a compressão, pois reduz o valor da distorção e mascara o real desempenho do método de compressão. Outra consequência observada quando se mantém o valor médio é o aumento do valor de todos os coeficientes de projeção no subespaço escala, fazendo com que um coeficiente que seria considerado não-significativo sem a presença do valor médio seja retido, diminuindo a taxa de compressão.

Após a retirada do valor médio, a função *wavelet* que melhor se adapta ao sinal é calculada. Tal procedimento aumenta a concentração da energia do sinal em alguns coeficientes. A função *wavelet* é calculada utilizando o modelo de otimização descrito na Seção 3.4. Após uma série de comparações entre algoritmos para a resolução de problemas de otimização o que apresentou melhores resultados foi o algoritmo de programação sequencial quadrática (*sequential quadratic programming* - SQP), implementado pela função FMINCON do *Toolbox* de Otimização do Matlab [BDS02]. Na função FMINCON é necessário indicar um vetor de parâmetros que atua como um ponto de partida do algoritmo, assim como a função objetivo considerada. Na implementação proposta são indicados quatro diferentes pontos de partida: os parâmetros da função de Haar, os parâmetros que geram a função de Daubechies com o mesmo tamanho m usado para gerar a função *wavelet* que melhor se adapta ao sinal e dois vetores de parâmetros aleatórios. A função FMINCON é executada para cada ponto de partida e o vetor de parâmetros resultante que gera o menor PRD é armazenado. Note que o número γ de parâmetros é sempre o mesmo durante o processo de otimização. As funções de Haar e de Daubechies são utilizadas como ponto de partida do algoritmo de otimização por serem as funções ortogonais clássicas que resultam em boas compressões de ECGs [BESS00].

Após a obtenção dos coeficientes dos filtros escala e *wavelet* que minimizam a distorção do sinal reconstruído, o procedimento de decomposição é aplicado para determinarem-se os coeficientes de projeção do sinal nas bases escala e *wavelet*, sendo que apenas os coeficientes mais significativos são retidos. Os coeficientes a serem retidos são os coeficientes cujos valores absolutos são maiores do que um valor de limiar ζ , definido de forma que a distorção do sinal resultante da compressão corresponda à distorção pré-definida pelo usuário (UPRD) com uma certa tolerância ϵ . O Algoritmo 1 é baseado no algoritmo de biseção apresentado em [BMB07] e é utilizado para determinar o valor ζ de limiar, sendo c o conjunto total de coeficientes escala e *wavelets* arranjados conforme mostrado na Equação (3.57).

Algoritmo 1 $[\zeta] = \text{Biseção}(c, \epsilon, UPRD)$

$$PRD \leftarrow 1; \quad \zeta_{min} \leftarrow 0; \quad \zeta_{max} \leftarrow \max_n |c_n|; \quad \mathcal{E} \leftarrow \sum_k |c_k|^2$$

while $|PRD - UPRD|/UPRD > \epsilon$ **do**

$$\zeta \leftarrow (\zeta_{min} + \zeta_{max})/2$$

$$\mathcal{K} \leftarrow \{\arg_k |c_k| \geq \zeta\}$$

$$PRD \leftarrow (1 - \frac{1}{\mathcal{E}} \sum_{k \in \mathcal{K}} |c_k|^2)^{1/2}$$

if $PRD < UPRD$ **then** $\zeta_{min} \leftarrow \zeta$ **else** $\zeta_{max} \leftarrow \zeta$

end while

Os coeficientes retidos são quantizados utilizando um procedimento adaptativo, que consiste em utilizar um número de *bits* diferente para cada subespaço de decomposição considerado. O número $q[n]$ de *bits* utilizado para quantizar o n -ésimo subespaço é igual ao mínimo número de *bits* que causa uma distorção correspondente ao UPRD com uma tolerância de ϵ_q . Para o cálculo do número de *bits* de quantização em um dado subespaço os coeficientes dos demais subespaços devem manter seus valores originais, de forma que a distorção calculada represente apenas a distorção causada pelos coeficientes do subespaço analisado. Após o cálculo do número de *bits* utilizados para quantizar os coeficientes em um subespaço, os valores dos coeficientes são restaurados a seus valores originais e o processo de cálculo do número de *bits* de quantização dos coeficientes do subespaço seguinte é iniciado. Ao fim do cálculo do número de *bits* usados para quantizar os coeficientes de todos os subespaços todos os coeficientes são quantizados e armazenados.

Os coeficientes quantizados são codificados para aumentar a compressão do sinal, assim como o *bitmap* que informa as posições dos coeficientes retidos. Cada *bit* do *bitmap* representa a posição de um coeficiente, de forma que se um coeficiente foi retido então o respectivo *bit* assume o valor 1, caso contrário assume o valor 0. O método de codificação por subespaço dos coeficientes quantizados e do *bitmap* consiste nos seguintes passos:

1. Para cada subespaço, faça:

- (a) No *bitmap*, se existir uma quantidade de *bits* 1 maior que de *bits* 0, insira um *bit* 1 na *string* de saída e inverta os valores dos *bits* do subespaço atual. Caso contrário, insira um *bit* 0 na *string* de saída;
- (b) Construa um vetor v que contenha os tamanhos das sequências de zeros do *bitmap*. O número de *bits* utilizados para codificar os tamanhos das sequências de zeros no subespaço atual é dado por

$$q_z = \lfloor \log_2(\text{mean} + \text{std}) \rfloor, \quad (4.1)$$

sendo “mean” e “std” a média e o desvio padrão do vetor v , respectivamente. A sequência de zeros que ocorre após o último coeficiente significativo (considerando todos os coeficientes de todos os subespaços) não é considerada no cálculo de q_z ;

- (c) Insira na *string* de saída o valor q_z usando um número fixo de *bits* (normalmente são usados três *bits*);
- (d) Analise o trecho do *bitmap* referente ao subespaço atual: se um *bit* 1 ocorrer, insira um *bit* 1 na *string* de saída e o valor do coeficiente de projeção quantizado referente a essa posição; se um *bit* 0 ocorrer, insira um *bit* 0 na *string* de saída e o valor do tamanho da sequência de zeros que se inicia com este *bit* 0 (técnica RLE) usando q_z *bits*. Caso os *bits* no presente subespaço estiverem invertidos, se um *bit* 1 ocorrer simplesmente insira o *bit* na *string* de saída; se um *bit* 0 ocorrer, insira o *bit* 0 na saída, o valor codificado do tamanho da sequência de zeros usando q_z *bits* e os valores quantizados dos coeficientes correspondentes à sequência de zeros.

O algoritmo prossegue até que o último coeficiente significativo seja incluído na *string*. Note que a sequência de zeros que ocorre após o último coeficiente significativo é desconsiderada.

São utilizados números diferentes de *bits* para codificar os tamanhos das sequências de zeros no *bitmap* pois, pela análise dos *bitmaps* obtidos em experimentos numéricos [AB08e], nota-se que uma grande porcentagem dos *bits* referentes ao primeiro subespaço é igual a 1, enquanto que quase todos os *bits* referentes ao último subespaço são iguais a 0. Como os *bits* 0 são codificados utilizando a técnica RLE e os tamanhos das sequências de zeros apresentam uma considerável variação dependendo do subespaço analisado, codificar os tamanhos das sequências de zeros utilizando números diferentes de *bits* para cada subespaço considerado aumenta a compressão obtida.

Além da *string* de saída é necessário também armazenar um cabeçalho que contenha as informações necessárias para reconstruir o sinal. O cabeçalho é composto das seguintes informações:

- Os valores dos parâmetros θ , quantizados com 11 *bits*;
- Os números de *bits* de quantização $q[n]$ usados em cada subespaço, codificados utilizando 4 *bits*;
- O valor absoluto do menor coeficiente e o intervalo de quantização considerado em cada subespaço, armazenados utilizando 11 *bits*.

4.3 Experimentos numéricos

Os resultados da aplicação do método de compressão RLEA em um ECG são apresentados e analisados em detalhes a fim de destacar as principais vantagens do método. O ECG utilizado é obtido da base de dados de arritmias do MIT-BIH [GAG⁺00] e corresponde a dois minutos (43202 amostras) do sinal identificado como 117. O sinal considerado apresenta 100 batimentos e aproximadamente 433 amostras por batimento, e é decomposto em $\eta = 6$ subespaços. A distorção especificada é $UPRD = 4\%$, a tolerância de retenção é $\epsilon = 1\%$ e a tolerância de quantização é $\epsilon_q = 10\%$. A Figura 4.1 mostra um trecho dos sinais original e reconstruído que contém a janela que apresentou o pior valor de MAPRD, também apresentado na figura. O tamanho da janela utilizada é $w = 433$.

A Tabela 4.1 mostra os resultados obtidos da aplicação do método RLEA no sinal 117. Note que a porcentagem dos coeficientes retidos e a energia retida diminuem com o aumento da resolução do subespaço, o que implica no aumento da frequência de ocorrência de zeros no *bitmap*. Tal aumento pode também ser observado pela análise do número de *bits* usados para representar os tamanhos das sequências de zeros nos subespaços. Note que os *bits* são invertidos nos três primeiros subespaços, portanto o número de zeros nos três primeiros subespaços é muito pequeno e aumenta à medida em que os demais subespaços são analisados, como se pode observar pela análise do número de *bits* usados para representar as sequências de zeros nos demais subespaços. O aumento do número de *bits* usados para representar as sequências de zeros é uma consequência direta do aumento do tamanho médio das sequências de zeros e, conseqüentemente, do aumento da frequência de ocorrência desses zeros. Por fim, o número de *bits* de quantização dos coeficientes retidos apresenta um desvio pequeno para os primeiros cinco subespaços e diminui consideravelmente nos dois últimos subespaços, o que significa que a amplitude dos coeficientes dos dois últimos subespaços é menor e tais coeficientes podem ser representados com um número menor de *bits*. A diminuição da amplitude dos coeficientes nos dois últimos subespaços também pode ser observada pela análise dos valores da média e do desvio padrão dos coeficientes retidos e pela porcentagem de energia retida. Tal característica

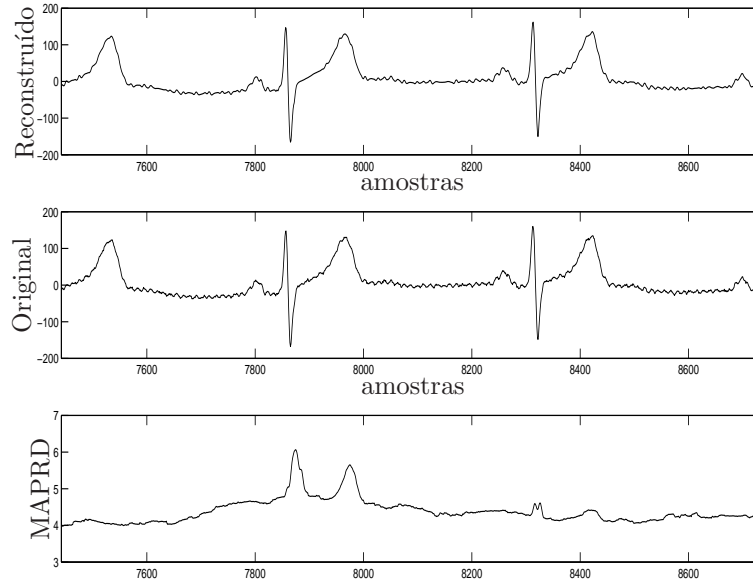


Figura 4.1: Parte dos sinais reconstruído (acima) e original (meio) do ECG de número 117 do banco de dados do MIT-BIH. A parte mostrada contém a janela de pior MAPRD, dado na parte de baixo. O pior valor de MAPRD ocorre para a janela que se inicia na amostra 7875.

não é considerada no método apresentado em [BMB07], em que o número de *bits* de quantização é constante para todos os coeficientes.

A seguir o método proposto (RLEA) é comparado com os métodos Rajoub [Raj02], *Two-Role Encoder* (TRE) [BMB07] e SPIHT [ZLP00]. Quatorze ECGs com duração de dois minutos (43202 amostras cada sinal) obtidos do banco de dados de arritmias do MIT-BIH [GAG⁺00] são utilizados nos experimentos, sendo os onze primeiros utilizados por Benzid *et al.* em [BMB07] e os três demais são considerados sinais de difícil compressão. No método RLEA são utilizados os valores de tolerâncias $\epsilon = 1\%$ e $\epsilon_q = 10\%$, os sinais são decompostos em $\eta = 6$ subespaços *wavelet* e o número de coeficientes dos filtros escala e *wavelet* é $m = 10$, resultando em $\gamma = 4$ parâmetros θ . O algoritmo SPIHT é implementado utilizando janelas de 1024 amostras, a *wavelet* utilizada é a biortogonal bior4.4 cujos filtros de análise escala e *wavelet* possuem tamanhos $m = 9$ e $m = 7$ respectivamente e o sinal é decomposto em $\eta = 6$ subespaços *wavelet*. No algoritmo TRE a *wavelet* bior4.4 é utilizada, o sinal é decomposto em $\eta = 6$ subespaços *wavelet* e os valores de tolerância utilizados são $\epsilon = 1\%$ e $\epsilon_q = 10\%$. No algoritmo de Rajoub é utilizada a *wavelet* bior4.4, o sinal é decomposto em $\eta = 5$ níveis, todos os coeficientes do subespaço escala são retidos e no máximo 99.95% da energia dos subespaços *wavelet* é retida. O número de coeficientes retidos varia com o sinal comprimido, conforme feito em [SABP08], para que a distorção do sinal reconstruído possa ser controlada. A distorção nominal é $UPRD = 4\%$ para todos os casos. O computador utilizado nas simulações é um Intel[®] Core[™] 2 Duo 1.66 GHz, com 1 GB de memória RAM.

Os valores das taxas de compressão obtidas utilizando os métodos de compressão descritos são apresentados na Tabela 4.2 e as Figuras 4.2, 4.3 e 4.4 são ilustrações das comparações do

Tabela 4.1: Resultados do método RLEA: sinal 117 do MIT-BIH.

Subespaço	a_0	b_0	b_1	b_2	b_3	b_4	b_5
Número de coeficientes	683	683	1358	2708	5408	10807	21605
Porcentagem de coeficientes retidos	97,80	92,24	70,77	43,35	50,94	4,26	0,001
Porcentagem da energia retida	99,99	99,99	99,96	99,89	99,00	66,57	0,75
Média dos valores absolutos dos coeficientes retidos	213,77	83,17	54,49	25,28	10,12	0,55	0,001
Desvio padrão dos valores absolutos dos coeficientes retidos	181,24	86,37	75,82	68,79	24,38	2,81	0,12
<i>Bits</i> de quantização	8	6	7	8	7	2	1
Média dos tamanhos das sequências de zeros nos <i>bitmaps</i>	41,75	13,12	6,81	3,57	1,94	8,95	9,27
Desvio padrão dos tamanhos das sequências de zeros nos <i>bitmaps</i>	45,64	11,55	3,74	3,21	1,50	24,98	28,19
Número de <i>Bits</i> de representação das sequências de zeros nos <i>bitmaps</i>	6	4	3	2	1	5	5
Inversão de <i>bits</i> nos <i>bitmaps</i>	Sim	Sim	Sim	Não	Sim	Não	Não
Porcentagem de coeficientes retidos	15,38						
Taxa de compressão (CR)	7,92						
Distorção (PRD)	4,07 %						

resultado obtido pela aplicação do RLEA com os resultados obtidos pela aplicação do método de Rajoub, do TRE e do SPIHT, respectivamente. Note que o método RLEA apresenta resultados melhores que os demais métodos de compressão para todos os sinais considerados. Uma comparação entre o método RLEA e o método de compressão por transformada DCT é realizada em [AB08c], sendo que o RLEA apresenta resultados melhores que o DCT.

A Tabela 4.2 também mostra os tempos utilizados para a execução dos métodos de compressão para cada ECG considerado. Note que nos métodos de Rajoub, TRE e SPIHT os tempos de execução são aproximadamente inversamente proporcionais às taxas de compressão obtidas. Isto ocorre porque quanto maior a taxa de compressão, menor o número de coeficientes armazenados e, portanto, menor o tempo necessário para realizar o procedimento de codificação. No entanto, os tempos de execução dos métodos de Rajoub e TRE são pequenos, ao contrário do que ocorre no método SPIHT. Já no método RLEA os tempos de execução são maiores que os observados nos métodos de Rajoub e TRE, mas também são menores que os observados no método SPIHT. Observa-se, também, que no método RLEA não há relação entre as taxas de compressão e os tempos totais de execução. Isso ocorre pois o tempo decorrido no procedimento de otimização, tempo esse sem relação com a taxa de compressão obtida, corresponde em média a 95% do tempo total de execução. O tempo decorrido após a otimização, que corresponde ao tempo de execução dos procedimentos de quantização e codificação e é sensível ao número de coeficientes retidos, apresenta um comportamento aproximadamente inversamente proporcional à taxa de compressão correspondente. Porém, como o tempo após a otimização é muito pequeno em relação ao tempo total, o número de coeficientes retidos, e conseqüentemente a taxa

de compressão obtida, não influi consideravelmente no tempo total de execução do método.

Tabela 4.2: Taxas de compressão (CR) obtidas e tempos de execução (TE) da aplicação dos métodos de Rajoub, TRE, SPIHT e o RLEA, considerando UPRD= 4%. No método RLEA os tempos de execução estão divididos no tempo decorrido da otimização (TE_o) e no tempo decorrido do restante do método (TE_r).

Sinal	Rajoub		TRE		SPIHT		RLEA	
	CR	TE (s)	CR	TE (s)	CR	TE (s)	CR	$TE_o + TE_r$ (s)
100	3.91	3.39	5.15	6.87	5.00	127.66	5.93	42.14 + 2.50
101	4.44	3.08	7.10	2.81	5.10	128.33	8.30	42.64 + 1.72
102	4.46	2.97	5.21	6.05	5.20	125.17	6.60	36.06 + 2.20
103	9.02	1.72	10.57	2.05	8.65	84.41	11.89	54.66 + 1.25
107	10.00	1.69	12.96	1.66	9.86	79.25	14.59	71.78 + 1.09
109	9.36	1.64	11.77	1.75	8.83	89.64	13.58	46.58 + 1.17
111	3.55	3.95	5.77	4.62	4.93	135.16	8.44	50.83 + 1.91
115	9.86	1.59	12.95	1.67	7.98	91.05	13.48	61.11 + 1.20
117	4.56	2.77	6.52	3.83	6.17	113.61	8.10	72.70 + 1.83
118	5.32	2.44	8.51	2.59	6.82	107.11	9.55	58.02 + 1.55
119	9.32	1.59	13.04	1.62	8.82	86.23	14.16	52.16 + 1.09
213	7.54	1.72	9.78	2.14	7.39	99.77	10.60	42.91 + 1.20
222	2.43	5.37	2.84	22.89	3.57	167.95	4.11	46.61 + 4.73
232	2.90	3.47	3.35	15.69	3.62	171.89	4.28	20.47 + 2.17

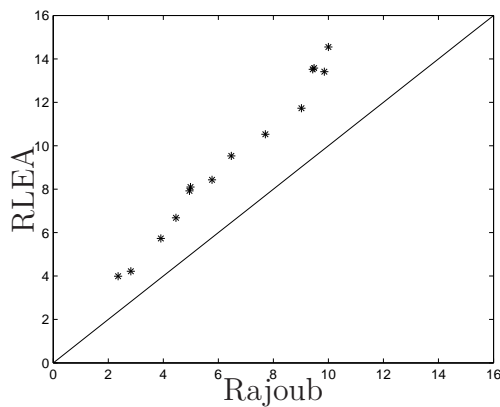


Figura 4.2: Taxas de compressão do RLEA e do método de Rajoub para quatorze ECGs com arritmia.

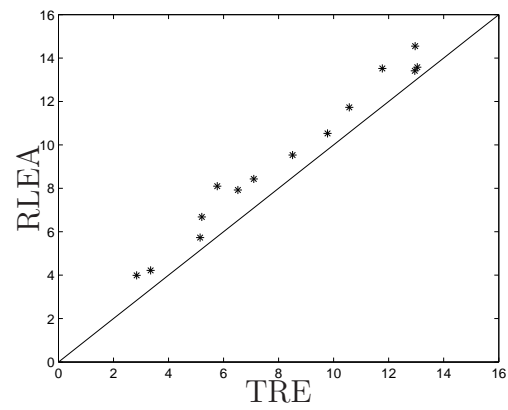


Figura 4.3: Taxas de compressão do RLEA e do TRE para quatorze ECGs com arritmia.

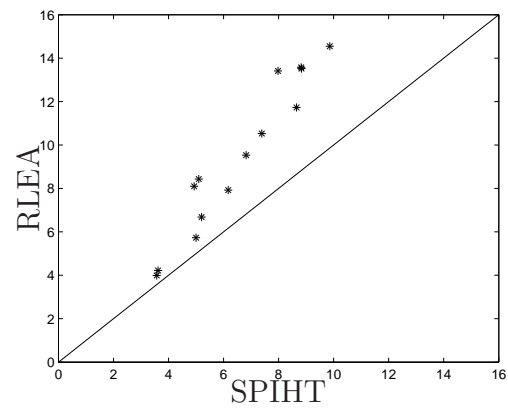


Figura 4.4: Taxas de compressão do RLEA e do SPIHT para quatorze ECGs com arritmia.

4.4 Conclusão

Nesse capítulo foi apresentado o método de compressão denominado *Run Length Encoding Adaptativo*, que consiste nos seguintes passos: i) escolha adaptativa da *wavelet* que melhor se ajusta ao sinal; ii) seleção dos coeficientes significativos que satisfazem uma medida de distorção pré-estabelecida; iii) quantização adaptativa dos coeficientes de projeção, definindo um número de *bits* de quantização diferente para cada subespaço; e iv) aplicação do método *Run Length Encoding* cujas palavras-código possuem tamanhos variados dependendo do subespaço considerado. Experimentos numéricos foram conduzidos para comparar o método RLEA com três outros métodos baseados em transformadas *wavelet*: Rajoub [Raj02], TRE [BMB07] e SPIHT [ZLP00]. Os experimentos mostraram que o método RLEA apresenta um desempenho superior aos outros métodos em todos os casos analisados.

Compressão por Decomposição em Valores Singulares

5.1 Introdução

O método de compressão apresentado por Wei *et al.* [WCCJ01] consiste na aplicação da Decomposição em Valores Singulares (*Singular Values Decomposition* - SVD) no conjunto de batimentos do ECG a fim de obter uma base de representação para os batimentos. O método apresenta bons resultados na compressão, porém existem diversos pontos em que pode ser melhorado.

O método proposto neste capítulo, denominado *Zero Padding Singular Value Decomposition* (ZPSVD) [ABP09b], consiste em preprocessar o sinal para detectar os batimentos, aplicar um procedimento de normalização para que todos os batimentos possuam o mesmo número de amostras, aplicar o SVD para obtenção da base de representação, reter alguns dos elementos da base, quantizar adaptativamente os coeficientes de projeção dos batimentos na base e codificar os elementos da base.

O capítulo está organizado da seguinte forma. Na Seção 5.2 são apresentadas as principais propriedades da decomposição em valores singulares. O método de compressão denominado ZPSVD é descrito na Seção 5.3. Na Seção 5.4 é apresentada uma série de experimentos numéricos para comparar o método ZPSVD com o descrito por Wei *et al.* [WCCJ01]. Uma série de comparações entre os métodos de compressão ZPSVD e RLEA, propostos nessa dissertação, é realizada na Seção 5.5. O capítulo é concluído na Seção 5.6.

5.2 Decomposição em Valores Singulares

Seja $F \in \mathbb{R}^{\nu \times n}$ a matriz dada por

$$F = [f'_1 \ f'_2 \ \dots \ f'_\nu]', \quad (5.1)$$

sendo f_k , $k = 1, \dots, \nu$ os batimentos de um dado eletrocardiograma. A decomposição da matriz F no produto de matrizes

$$F = USV' \ , \ U \in \mathbb{R}^{\nu \times \nu} \ , \ V \in \mathbb{R}^{n \times n} \ \text{e} \ S = \text{diag}(\sigma_1, \dots, \sigma_r) \ , \ S \in \mathbb{R}^{\nu \times n} \ , \ r \leq \min(\nu, n) \quad (5.2)$$

é denominada de Decomposição em Valores Singulares (*Singular Values Decomposition - SVD*), sendo U e V matrizes ortonormais, r o *rank* e σ_k os valores singulares da matriz F [AB06], [GL96]. Os valores singulares são tais que $\sigma_k \geq \sigma_{k+1}$.

As matrizes U , S e V são obtidas pelo cálculo dos autovalores e autovetores das matrizes FF' e $F'F$. Por construção, as colunas da matriz U são os autovetores de FF' e as colunas da matriz V são os autovetores de $F'F$. Como os autovalores não-nulos da matriz FF' são iguais aos autovalores não nulos da matriz $F'F$, tem-se que

$$FF'u_k = \lambda_k u_k \quad , \quad F'Fv_k = \lambda_k v_k \quad , \quad \lambda_k > 0, \quad (5.3)$$

sendo λ_k os autovalores não-nulos das matrizes FF' e $F'F$. Os valores singulares da matriz F são calculados a partir dos autovalores λ_k [AB06] e são dados por

$$\sigma_k = \sqrt{\lambda_k} \quad , \quad k = 1, \dots, r. \quad (5.4)$$

As colunas da matriz V podem ser vistas como os elementos de uma base para o subespaço formado pelos batimentos f_k e cada elemento p_{ij} da matriz $P = US$ pode ser visto como o coeficiente de projeção do batimento f_i no j -ésimo componente da base, no caso o vetor coluna v_j .

A matriz $F = USV'$ pode ser escrita como

$$F = \sum_{k=1}^r u_k \sigma_k v_k'. \quad (5.5)$$

Uma das principais propriedades da decomposição em valores singulares é que a solução do problema de otimização

$$\begin{aligned} \min_{\hat{F}} \|F - \hat{F}\|^2 \\ \text{s. a. } \text{rank}(\hat{F}) = s \quad , \quad s \leq r \end{aligned} \quad (5.6)$$

é a matriz \hat{F} [Gab78] dada por

$$\hat{F} = \sum_{k=1}^s u_k \sigma_k v_k', \quad (5.7)$$

sendo $\|\cdot\|$ a norma de Frobenius de uma matriz. Uma outra forma de representar a matriz \hat{F} é pelo produto

$$\hat{F} = CG \quad , \quad V = [G' \quad H'] \quad , \quad US = [C \quad B], \quad (5.8)$$

sendo $G' \in \mathbb{R}^{n \times s}$ a matriz que contém as s primeiras colunas da matriz V e $C \in \mathbb{R}^{\nu \times s}$ a matriz que contém as s primeiras colunas do produto US . A matriz \hat{F} resultante do problema mostrado na Equação (5.6) é denominada de aproximação de *rank* s da matriz F . Quanto maior o valor de s melhor a aproximação e mais componentes singulares são utilizados. O problema de otimização mostrado na Equação (5.6) representa o problema de encontrar uma base de funções com $s \leq r$ elementos que minimiza a soma do erro quadrático de projeção de cada batimento na base. Neste contexto, os valores singulares σ_k podem ser vistos como uma medida da contribuição de cada componente singular na representação do sinal: quanto maior o valor de σ_k , maior a energia concentrada nos coeficientes de projeção dos batimentos no k -ésimo componente da base e, portanto, melhor a representação dos batimentos quando

se usa o k -ésimo componente. Quando tem-se como motivação a compressão do conjunto de funções F é possível aplicar a decomposição em valores singulares e armazenar apenas os s elementos da base e seus respectivos coeficientes de projeção correspondentes aos s maiores valores singulares. O armazenamento de um número menor de componentes aumenta a taxa de compressão do método, porém também aumenta a distorção do sinal reconstruído.

Uma vez que o SVD resulta na base que minimiza a soma dos erros quadráticos da projeção ortogonal das funções nessa base, a compressão utilizando o SVD apresenta um melhor desempenho se todas as funções possuírem aproximadamente o mesmo formato. Neste caso, poucos componentes são necessários para que as funções sejam bem representadas. Considere, por exemplo, o caso extremo em que todas as funções são iguais a f_k

$$F = [f'_k \ f'_k \ \dots \ f'_k]' . \quad (5.9)$$

A matriz F possui *rank* $r = 1$, portanto apenas o primeiro valor singular σ_1 é diferente de zero e somente um componente (a própria função f_k) é necessário para representar o conjunto de funções. Por outro lado, considerando a situação oposta em que as funções são ortogonais:

$$FF' = I_\nu \ , \ F'F = I_n . \quad (5.10)$$

Como os valores singulares σ_k são iguais à raiz quadrada dos autovalores não-nulos das matrizes FF' e $F'F$ [AB06], tem-se que

$$\sigma_k = 1 \ , \ k = 1, \dots, r \ , \ r = \min\{\nu, n\} . \quad (5.11)$$

Como os valores singulares são uma medida da contribuição de cada elemento da base para a representação do sinal, nesse caso todos os elementos apresentam a mesma contribuição e a energia do sinal é dividida igualmente, sendo este o pior cenário para um método de compressão baseado em transformadas [Sal07].

5.3 Método de compressão ZPSVD

A primeira fase do método de compressão ZPSVD [ABP09b] é o pré-processamento do ECG. O primeiro passo do pré-processamento é realizar a detecção dos batimentos do eletrocardiograma. Cada batimento é o segmento entre picos de duas ondas R consecutivas, picos estes detectados utilizando o método descrito na Seção 2.3. O trecho entre a primeira amostra e o pico da primeira onda R é armazenado separadamente.

Os batimentos são normalizados para que tenham o mesmo número de amostras, procedimento necessário para a aplicação da decomposição em valores singulares. Cada batimento é rearranjado de forma que os picos R estejam na mesma amostra. Em [WCCJ01], a normalização é feita a partir da interpolação dos batimentos de forma que o número de amostras seja igual ao número médio de amostras dos batimentos. Inserir amostras no batimento por interpolação causa distorções no formato. A normalização realizada no método ZPSVD consiste em deslocar de forma circular cada batimento em um quarto do número de amostras para a direita. O número de amostras dos batimentos é equalizado pela inserção de zeros (*zero padding*) após a última amostra de cada batimento. Cada batimento é subtraído do valor de sua última amostra, que é armazenado para reconstrução. Um deslocamento circular adicional é realizado para que os picos das ondas R estejam na mesma posição.

Cada batimento normalizado é uma linha f_k da matriz F ,

$$F = [f'_1 \quad f'_2 \quad \cdots \quad f'_\nu]'. \quad (5.12)$$

O procedimento de decomposição em valores singulares é aplicado na matriz F como mostrado na Equação (5.2), gerando a matriz V , cujas colunas, também chamadas de componentes singulares, correspondem aos elementos da base de representação do conjunto de sinais, e a matriz $P = US$, cujas colunas correspondem aos coeficientes de projeção de cada batimento na base de representação. A matriz \hat{F} resultante do problema mostrado na Equação (5.6) é denominada de aproximação de *rank s* da matriz F .

Em [WCCJ01], os coeficientes de projeção são quantizados utilizando um número fixo de *bits*. Em geral a contribuição do k -ésimo elemento da base é menor que a contribuição do p -ésimo elemento para $k > p$. Assim, o número de *bits* para quantizar os coeficientes referentes ao k -ésimo elemento pode em geral ser menor do que o número de *bits* para quantizar os coeficientes referentes ao p -ésimo elemento.

O número de *bits* de quantização é determinado pelos seguintes passos:

- Determine c_m (coeficiente de projeção de maior valor absoluto). Todos os coeficientes de projeção no mesmo elemento da base relacionado ao coeficiente c_m são quantizados por um número q_m de *bits*;
- Para cada elemento k da base determine $c_m[k]$ (coeficiente de maior valor absoluto). O número de *bits* usados para quantizar os coeficientes de projeção no k -ésimo elemento da base é

$$q[k] = q_m - \left\lfloor \log_2 \left(\frac{c_m}{c_m[k]} \right) \right\rfloor. \quad (5.13)$$

Note que, se $c_m[k] = c_m/2$, então $q[k] = q_m - 1$.

A compressão da base no método ZPSVD consiste na aplicação da transformada *wavelet* a cada elemento da base e na retenção dos coeficientes mais significativos. A *wavelet* utilizada é a Db2 (Daubechies-2) com quatro níveis de decomposição. O número de coeficientes retidos é tal que a energia corresponda a 99.90% da energia total, exceto para os coeficientes referentes ao primeiro elemento da base, que são armazenados sem a aplicação de qualquer compressão, pois uma pequena distorção no primeiro componente resulta em grande distorção no sinal recuperado. Os coeficientes são quantizados e codificados junto com o *bitmap* que informa a posição dos coeficientes retidos utilizando o método de codificação por subespaços da Seção 4.2. Os parâmetros da compressão da base foram definidos a partir de análises de experimentos descritos em [AB08a].

As seguintes informações são armazenadas: os coeficientes quantizados de projeção, os números de *bits* utilizados para quantizar os coeficientes de projeção em cada elemento da base, a base codificada, os números de amostras dos batimentos originais, o trecho inicial do sinal e o valor subtraído de cada batimento antes de aplicar a normalização dos batimentos usando a técnica de *zero padding*.

5.4 Experimentos numéricos

Os sinais utilizados nos experimentos são os primeiros 10 minutos dos ECGs com arritmia do banco de dados do MIT-BIH [Phy]. O número máximo de *bits* de quantização dos coeficientes

de projeção é $q_m = 7$ e o número de componentes retidos é o mínimo número de coeficientes que garante $\text{PRD} \leq 6\%$. O número de *bits* de quantização não é especificado em [WCCJ01]. Assim, o método de Wei *et al.* foi implementado quantizando tanto os coeficientes de projeção quanto os elementos da base utilizando 13 *bits*, de forma que os resultados fossem aproximadamente iguais aos apresentados em [WCCJ01]. O computador utilizado nas simulações é um Intel® Core™ 2 Duo 1.66 GHz, com 1 GB de memória RAM.

A Tabela 5.1 mostra os resultados dos métodos ZPSVD e de [WCCJ01] e a Figura 5.1 mostra os valores das taxas de compressão. Também são mostradas na Tabela 5.1 a razão entre os números de amostras do maior e do menor batimento e os tempos de execução dos métodos. O valor do UPRD utilizado é diferente do valor utilizado nos experimentos apresentados na Seção 4.3 pois os métodos baseados na decomposição em valores singulares em geral não exibem uma boa taxa de compressão para valores muito pequenos de UPRD. Uma comparação entre o método ZPSVD com a compressão via DCT é apresentada em [AB08c], sendo que o ZPSVD em geral apresenta resultados melhores que o DCT.

Tabela 5.1: Taxas de compressão (CR) e distorções (PRD) obtidas pelo método de Wei *et al.* [WCCJ01] e pelo método ZPSVD. O número s de componentes é tal que a distorção é menor ou igual a 6% dentro de uma certa tolerância. Também são mostrados a razão entre os números de amostras do maior e do menor batimento e o tempo de execução (TE) dos métodos.

ECG	Razão	Wei <i>et al.</i>				ZPSVD			
		s	CR	PRD (%)	TE(s)	s	CR	PRD (%)	TE(s)
232	9.86	87	2.16	5.99	19.81	67	4.51	5.95	20.31
222	7.48	65	2.69	5.97	15.94	62	6.73	5.99	15.09
118	3.29	28	6.08	5.90	10.09	41	12.28	6.00	14.30
102	3.24	38	4.61	5.99	11.52	26	15.94	5.93	7.59
101	4.57	26	6.99	5.93	9.36	29	16.36	5.99	8.56
100	1.90	27	6.33	5.97	10.06	19	21.31	5.92	7.58
111	2.99	20	8.79	5.93	8.06	18	22.44	5.91	7.05
213	2.94	10	13.15	5.70	8.17	10	26.65	5.59	7.55
107	2.08	14	12.36	5.58	7.28	14	29.82	5.55	5.89
109	2.89	7	21.51	5.95	6.66	8	35.08	6.00	8.44
119	2.84	11	15.99	5.66	6.67	9	39.87	5.92	7.77
117	1.41	12	15.69	5.77	7.05	9	45.28	5.86	5.89
103	2.37	9	18.84	5.79	6.23	6	50.59	5.59	5.45
115	3.41	8	21.87	5.62	6.05	6	54.40	5.84	6.11

Note que os números s de componentes retidos são em geral inversamente proporcionais às taxas de compressão obtidas. De fato, um número menor de componentes retidos implica em um número menor de coeficientes de projeção, reduzindo o volume de dados que precisa ser armazenado para reconstruir o sinal e, conseqüentemente, aumentando a taxa de compressão. No entanto, o número de componentes a serem retidos para que a distorção especificada seja alcançada é muito dependente das características do sinal considerado. Note a correlação da razão entre os números de amostras do maior e do menor batimento e o número de componentes que precisam ser retidos para que a distorção do sinal reconstruído seja próxima da especifi-

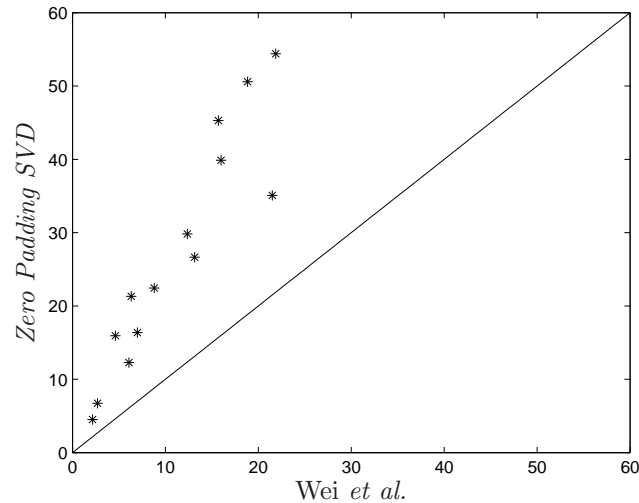


Figura 5.1: Valores da taxa de compressão de quatorze eletrocardiogramas para o ZPSVD e o método de Wei *et al.* usando o número mínimo de componentes tal que $\text{PRD} \leq 6\%$.

cada. Valores maiores para a razão implicam em uma maior irregularidade nos tamanhos dos batimentos detectados e, portanto, a base obtida a partir da aplicação do SVD depende de um número maior de componentes para representar adequadamente o sinal. Note também que as taxas de compressão obtidas pelo ZPSVD são maiores que as taxas de compressão obtidas pelo método de Wei *et al.* Uma possível evolução do método ZPSVD seria a determinação automática, por meio da análise da energia dos coeficientes de projeção e dos valores singulares, do número de componentes que produzem a distorção desejada para o sinal reconstruído, ao invés de considerar que o número de componentes é definido pelo usuário. Quanto ao tempo decorrido da execução dos métodos, pode-se notar que não há uma considerável diferença entre os tempos de execução dos métodos. A quantização e codificação tanto dos elementos da base quanto dos coeficientes de projeção têm um custo computacional maior no método ZPSVD, mas o procedimento de normalização por *zero padding* é menos custoso que o procedimento de normalização por interpolação realizado no método de Wei *et al.* Tais características fazem com que os tempos totais de execução sejam semelhantes para ambos os métodos.

A Figura 5.2 mostra dez batimentos após a aplicação do procedimento de *zero padding* nos sinais 117 e 232 que apresentaram, respectivamente, a menor e a maior razão entre os números de amostras do maior e do menor batimento. Note que os batimentos estão bem alinhados e que não são distorcidos pelo procedimento de normalização, como ocorre no procedimento de normalização do método de Wei *et al.* A Figura 5.2 também mostra, para cada sinal, o primeiro componente da base obtida pela aplicação do SVD nos batimentos, mostrando que o primeiro elemento da base apresenta aproximadamente o mesmo formato dos batimentos do ECG.

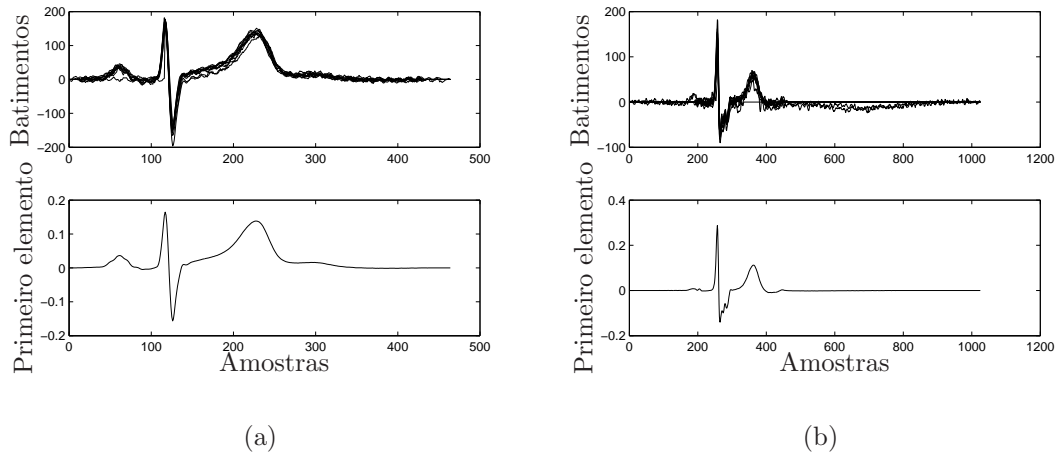


Figura 5.2: Dez batimentos e o primeiro componente da base dos sinais (a) 117 e (b) 232.

5.5 Comparação entre ZPSVD e RLEA

Os dois métodos de compressão de ECGs propostos, *Run Length Encoding Adaptativo* (RLEA) e o *Zero Padding SVD* (ZPSVD), são baseados em transformadas e buscam adequar a base de representação ao sinal a ser comprimido.

A Tabela 5.2 mostra os resultados dos métodos ZPSVD e RLEA para sinais com duração de dois minutos e sinais com duração de dez minutos. Os cinco sinais escolhidos dentre os 14 sinais de arritmia usados nos experimentos ilustram bem o comportamento geral dos sinais. O desempenho do RLEA não muda significativamente com a duração do sinal pois um sinal de maior duração implica em um número maior de coeficientes de projeção na base *wavelet* e, conseqüentemente, um número maior de coeficientes a serem retidos para que a distorção permaneça a mesma. Como houve um aumento tanto no tamanho do sinal original quanto no tamanho do sinal comprimido, a taxa de compressão permanece aproximadamente igual. No entanto, o desempenho do ZPSVD depende muito da duração do sinal considerado. Caso o sinal seja pequeno o tamanho da base é consideravelmente grande em relação ao tamanho do sinal e a taxa de compressão é pequena. Por outro lado, se o sinal é de longa duração, o tamanho da base não é significativo e, como o número de coeficientes de projeção no método ZPSVD é bem menor do que o número de amostras do sinal, a taxa de compressão é alta.

O método ZPSVD depende muito do sinal a ser comprimido, enquanto que o método RLEA apresenta uma maior robustez nesse ponto. Como no método ZPSVD cada batimento é descrito com um número pequeno de coeficientes de projeção, a qualidade do batimento reconstruído é muito sensível aos elementos da base obtida, que por sua vez é muito dependente do sinal a ser comprimido. Se o sinal apresentar batimentos altamente irregulares então são necessários muitos elementos na base para que o sinal seja bem representado, o que piora o desempenho do método de compressão. O método RLEA, por sua vez, utiliza uma maior quantidade de coeficientes para representar todo o sinal, o que o torna menos sensível à base obtida. A Figura 5.3 mostra o MAPRD (Equação (2.12)) de um trecho do sinal 232 comprimido para a aplicação dos dois métodos, utilizando uma janela de tamanho 357, que é o tamanho aproximado de um batimento. Note que em ambos os métodos os picos no MAPRD ocorrem para os batimentos de maior largura. Isso ocorre pois os dois métodos apresentam dificuldades na representação

Tabela 5.2: Taxas de compressão (CR) e distorções (PRD) obtidas pelo método ZPSVD e pelo método RLEA, para sinais de arritmia com 2 (43202 amostras) e 10 minutos (215000 amostras).

	ECG	ZPSVD			RLEA	
		s	CR	PRD (%)	CR	PRD (%)
Sinais com 2 minutos	107	12	18.93	5.80	18.36	6.05
	115	5	31.47	6.20	17.10	6.15
	117	9	16.10	6.01	13.73	6.10
	222	39	4.38	6.07	6.60	6.09
	232	55	1.43	6.04	6.38	6.09
Sinais com 10 minutos	107	13	31.30	6.02	18.10	6.05
	115	6	54.40	5.84	16.13	6.11
	117	8	49.11	6.26	14.90	6.10
	222	56	7.26	6.13	6.97	6.11
	232	66	4.58	6.02	8.67	6.11

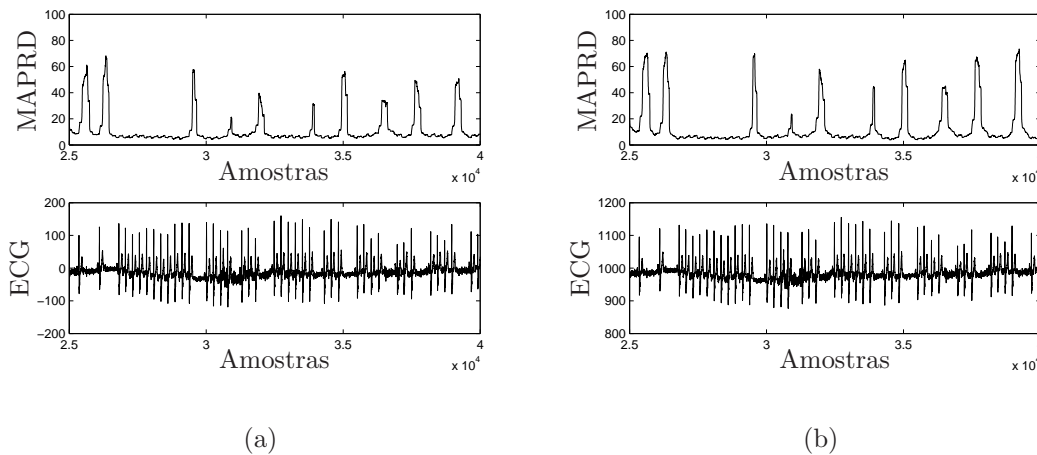


Figura 5.3: MAPRD (acima) após a compressão do sinal 232 (abaixo) utilizando as técnicas (a) RLEA e (b) ZPSVD.

da onda T, que possui uma maior influência no cálculo do MAPRD nos batimentos de maior largura. No entanto, os picos no MAPRD resultante da aplicação do ZPSVD são maiores do que os picos observados no MAPRD referente ao RLEA, indicando que o ZPSVD tem uma maior dificuldade de lidar com trechos ruidosos ou mesmo com batimentos com grandes variações de largura.

O método ZPSVD apresenta menor robustez devido também à sua dependência em relação ao pré-processamento do sinal, mais especificamente à detecção dos picos das ondas R do ECG. O desempenho do método depende consideravelmente da detecção dos batimentos pois a base retornada pelo SVD concentra mais energia em poucos coeficientes se os batimentos apresentarem uma maior semelhança entre si. Assim, falhas na detecção dos batimentos resultam em sinais muito diferentes entre si e prejudicam o desempenho do método. Além disso, a utilização do algoritmo de detecção de picos aumenta a complexidade e o tempo de execução do método de

compressão. No método RLEA o único pré-processamento consiste na retirada do valor médio do sinal. O algoritmo de detecção de batimentos é utilizado durante o procedimento de otimização, porém o resultado da otimização praticamente não se altera se o número de batimentos detectados é diferente do real. Com relação ao tempo de execução, a resolução do problema de otimização para obter a *wavelet* mais apropriada para a compressão do sinal é o passo com o maior tempo de execução do método, ao contrário da obtenção da base por SVD que leva um tempo relativamente curto para ser executado.

5.6 Conclusão

Neste capítulo foi apresentado o método de compressão *Zero Padding Singular Value Decomposition* (ZPSVD), que consiste em detectar e isolar os batimentos do ECG, normalizar os batimentos, aplicar a decomposição em valores singulares no conjunto de batimentos normalizados, reter parte dos elementos da base e dos respectivos coeficientes de projeção obtidos da aplicação do SVD, quantizar os coeficientes de projeção por um método de quantização adaptativa e codificar os elementos da base utilizando transformadas *wavelet*. Experimentos numéricos foram conduzidos a fim de comparar o método ZPSVD com o método descrito em Wei *et al.* [WCCJ01]. Os experimentos mostraram que o método ZPSVD apresenta um desempenho superior ao método apresentado em [WCCJ01] para todos os casos analisados. Uma comparação entre os métodos propostos ZPSVD e RLEA (*Run Length Encoding Adaptativo*) foi também realizada, explicitando as principais características e diferenças entre os dois métodos. O método RLEA apresenta uma maior robustez em relação ao formato do sinal a ser comprimido e resulta em melhores compressões para sinais de curta duração, enquanto que o método ZPSVD possui um desempenho mais dependente do formato do sinal e resulta em compressões melhores para sinais de longa duração.

Conclusões

As técnicas de compressão de eletrocardiogramas (ECGs) são o principal foco da presente dissertação, sendo a principal contribuição a proposta de dois métodos de compressão: *Run Length Encoding Adaptativo* (RLEA) e *Zero Padding Singular Values Decomposition* (ZPSVD). A técnica RLEA é baseada em transformadas *wavelet* e o primeiro passo é a obtenção, a partir da resolução de um problema de otimização, da função *wavelet* que melhor se adapta ao ECG. O problema de otimização define a *wavelet* que minimiza a distorção do sinal recuperado. A *wavelet* pode ser descrita por uma sequência discreta de coeficientes *wavelet*, e que devem satisfazer uma série de restrições que tornam o problema de otimização de difícil resolução. Uma parametrização dos coeficientes *wavelet* é apresentada de forma que o problema de otimização torna-se mais simples. Após a obtenção da função *wavelet*, o procedimento de decomposição é aplicado ao ECG, resultando nos coeficientes de projeção na base *wavelet*. Os coeficientes de projeção mais significativos são retidos, sendo o número de coeficientes retidos calculado a partir de um algoritmo de biseção para que uma distorção máxima especificada seja satisfeita. Tanto os coeficientes retidos quanto um *bitmap* que informa quais coeficientes foram retidos são codificados utilizando uma técnica de compactação. A compactação consiste primeiro na quantização adaptativa dos coeficientes retidos, que é realizada definindo-se números diferentes de *bits* para os diferentes subespaços *wavelet*. Em seguida, os coeficientes quantizados e o *bitmap* são codificados utilizando uma versão modificada do algoritmo *Run Length Encoding* (RLE), que se baseia na análise do *bitmap*: cada *bit* 1 é seguido do valor do respectivo coeficiente quantizado e cada *bit* 0 é seguido da codificação RLE da sequência de zeros subsequente, sendo que o número de *bits* utilizados para representar a codificação RLE também varia com o subespaço *wavelet*.

No método ZPSVD primeiramente é realizada a detecção e o isolamento dos batimentos do ECG, seguida pela normalização dos comprimentos dos batimentos. A normalização se dá pela inserção de zeros (*zero padding*) para que os tamanhos dos batimentos sejam iguais ao tamanho do maior batimento. Os batimentos normalizados são dispostos em uma matriz, que é decomposta em valores singulares (*Singular Values Decomposition* - SVD). A aplicação do SVD gera uma base ortonormal que minimiza a soma dos erros quadráticos de projeção de cada batimento na base. Os elementos da base mais significativos (concentram uma maior energia de projeção) são retidos, assim como os respectivos coeficientes de projeção, sendo que o número de elementos retidos é definido de forma a satisfazer uma distorção especificada. Em seguida são realizadas as codificações dos elementos da base e dos coeficientes de projeção. Os elementos da base são codificados aplicando o procedimento de decomposição *wavelet* (utilizando uma função *wavelet* fixa), retendo os coeficientes de decomposição mais significativos, quantizando os coeficientes utilizando o procedimento de quantização adaptativa utilizado no método RLEA e codificando os coeficientes quantizados em conjunto com o *bitmap* que informa as posições

dos coeficientes retidos utilizando a técnica RLE modificada. Os coeficientes de projeção são quantizados de forma adaptativa, utilizando um número de *bits* particular para os coeficientes de cada elemento da base.

Uma breve comparação entre métodos de compactação (compressão sem perdas) foi apresentada. Algumas técnicas de processamento de ECGs comumente utilizadas em algoritmos de compressão, como a extração da *baseline* e a detecção dos batimentos, foram também apresentadas.

Uma análise comparativa entre medidas de distorção foi realizada para escolher a mais apropriada para comparar as técnicas de compressão. As medidas usadas foram a medida *Percent Root-mean-square Distortion* (PRD), que quantifica globalmente a distorção do sinal e a medida *Moving Average Percent Root-mean-square Distortion* (MAPRD), utilizada para uma análise local da distorção.

Outra contribuição importante foi a formulação do problema irrestrito de otimização cujo resultado são os parâmetros referentes à função *wavelet* que gera a compressão com menor distorção. O problema de otimização foi resolvido pela aplicação da função FMINCON do *toolbox* de otimização do Matlab.

No algoritmo de compressão RLEA, foi desenvolvida uma quantização adaptativa aplicada aos coeficientes de decomposição de cada subespaço *wavelet*. A codificação dos coeficientes quantizados, em conjunto com o *bitmap* que informa a localização dos coeficientes retidos, utilizando uma técnica de compactação baseada no algoritmo RLE, é também uma contribuição desta dissertação.

No método ZPSVD, as principais inovações consistem na normalização dos batimentos isolados a partir da inserção de zeros (*zero padding*) e na quantização adaptativa, diferente da utilizada no RLEA, dos coeficientes de projeção dos batimentos na base obtida pelo SVD, de forma que os coeficientes de projeção em diferentes elementos da base sejam quantizados por números distintos de *bits*.

Por fim, os métodos de compressão propostos foram comparados por meio de experimentos numéricos com outros métodos conhecidos na literatura por apresentarem bons resultados, sendo que os métodos propostos nesta dissertação foram superiores aos demais em todos os casos analisados. Em seguida o RLEA e o ZPSVD foram comparados entre si e concluiu-se que a técnica RLEA comporta-se melhor para sinais de menor duração e é menos sensível ao sinal de entrada. A técnica ZPSVD, por sua vez, apesar de apresentar um desempenho muito dependente do ECG a ser comprimido, apresenta bons resultados em sinais de longa duração.

Programas utilizados no mestrado

Os programas criados e utilizados durante o período de mestrado, bem como uma breve descrição dos mesmos, estão disponíveis em [AB08d].

Trabalhos produzidos

- Rosanna M. R. Silveira, Cristiano M. Agulhari, Ivanil S. Bonatti e Pedro L. D. Peres, ECG Compression Based on Extracted Wavelets. International Workshop on Telecommunications, 2007, Santa Rita do Sapucaí - MG;
- Rosanna M. R. Silveira, Cristiano M. Agulhari, Ivanil S. Bonatti e Pedro L. D. Peres, A

- Genetic Algorithm to Compress Electrocardiograms using Parameterized Wavelets. 7th IEEE International Symposium on Signal Processing and Information Technology, 2007, Cairo, Egito;
- Rosanna M. R. Silveira, Cristiano M. Agulhari, Ivanil S. Bonatti e Pedro L. D. Peres, Compressão de sinais de eletrocardiogramas com wavelets determinadas por otimização genética. XXV Simpósio Brasileiro de Telecomunicações - SBrT, 2007, Recife - PE.
 - Cristiano M. Agulhari, Rosanna M. R. Silveira, Ivanil S. Bonatti e Pedro L. D. Peres, Compressing Electrocardiogram Signals Using Parameterized Wavelets. 23rd ACM Symposium on Applied Computing, 2008, Fortaleza - CE;
 - Cristiano M. Agulhari, Ivanil S. Bonatti e Pedro L. D. Peres, An adaptive RLE encoder to compress electrocardiograms. International Workshop on Telecommunications - IWT 2009, 2009, São Paulo - SP (Aceito).
 - Cristiano M. Agulhari, Ivanil S. Bonatti e Pedro L. D. Peres, A Zero Padding SVD Encoder to Compress Electrocardiograms. Data Compression Conference - DCC 2009, 2009, Salt Lake City, UT (Aceito).

Trabalhos relacionados

- Ivanil S. Bonatti, Cristiano M. Agulhari, Amauri Lopes e Pedro L. D. Peres, Projeção em Bases Uniformes Não-Ortogonais. Revista TEMA - Tendências em Matemática Aplicada e Computacional, 2007, Volume 8, Número 1, pp. 43-52.

Bibliografia

- [AAF03] ALSHAMALI, A. and AL-FAHOUM, A. S., 2003. Comments on “An efficient coding algorithm for the compression of ECG signals using the wavelet transform”. *IEEE Transactions on Biomedical Engineering*, 50(8):1034–1037.
- [AASAZ00] AHMEDA, S. M., AL-SHROUFB, A., and ABO-ZAHHAD, M., 2000. ECG data compression using optimal non-orthogonal wavelet transform. *Medical Engineering & Physics*, 22(1):39–46.
- [AB92] ALLEN, V. A. and BELINA, J., 1992. ECG data compression using the discrete cosine transform (DCT). In: *Computers in Cardiology 1992*. 687–690.
- [AB06] ANTON, H. and BUSBY, R. C., 2006. *Contemporary Linear Algebra* (Wiley, New Jersey), 1st edition.
- [AB08a] AGULHARI, C. M. and BONATTI, I. S., 2008. Analysis of the compression method “ECG data compression using truncated singular values decomposition”. Technical report, Unicamp, Brazil, http://www.dt.fee.unicamp.br/~ivanil/adaptive_svd_encoder-analise_metodo_wei.pdf.
- [AB08b] AGULHARI, C. M. and BONATTI, I. S., 2008. Análise da baseline de eletrocardiogramas. Technical report, Unicamp, Brazil, http://www.dt.fee.unicamp.br/~ivanil/analise_baseline_ecgs.pdf.
- [AB08c] AGULHARI, C. M. and BONATTI, I. S., 2008. Comparação entre os métodos de compressão via DCT, RLEA e ZPSVD. Technical report, Unicamp, Brazil, http://www.dt.fee.unicamp.br/~ivanil/comparacao_dct_rlea_zpsvd.pdf.
- [AB08d] AGULHARI, C. M. and BONATTI, I. S., 2008. Descrição dos programas utilizados. Technical report, Unicamp, Brazil, http://www.dt.fee.unicamp.br/~ivanil/descricao_programas_utilizados.pdf.
- [AB08e] AGULHARI, C. M. and BONATTI, I. S., 2008. Métodos de compactação aplicados a sequências de bits. Technical report, Unicamp, Brazil, http://www.dt.fee.unicamp.br/~ivanil/estudo_metodos_compressao_do_bitmap.pdf.
- [ABP09a] AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2009. An adaptive RLE encoder to compress electrocardiograms. In: *International Workshop on Telecommunications - IWT'09, São Paulo, Brazil*.

- [ABP09b] AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2009. A zero padding SVD encoder to compress electrocardiograms. In: *Data Compression Conference - DCC'09, Salt Lake City, UT*.
- [ANR74] AHMED, N., NATARAJAN, T., and RAO, R. K., 1974. Discrete cosine transform. *IEEE Transactions on Computers*, (C-23):90–93.
- [ASAZA03] AL-SHROUF, A., ABO-ZAHHAD, M., and AHMED, S. M., 2003. A novel compression algorithm for electrocardiogram signals based on the linear prediction of the wavelet coefficients. *Digital Signal Processing*, 13:604–622.
- [AT82] ABENSTEIN, J. P. and TOMPKINS, W. J., 1982. A new data-reduction algorithm for real-time ECG analysis. *IEEE Transactions on Biomedical Engineering*, BME-29(1):43–48.
- [BDS02] BUCK, J. R., DANIEL, M. M., and SINGER, A. C., 2002. *Computer Explorations in Signals and Systems Using MATLAB®*. Signal Processing (Prentice Hall), 2nd edition.
- [BESS00] BESAR, R., ESWARAN, C., SARIB, S., and SIMPSON, R., 2000. On the choice of the wavelets for ECG data compression. In: *Proceedings of Acoustics, Speech, and Signal Processing ICASSP '00*. volume 6, 3614–3617.
- [BFB03] BOUCHEHAM, B., FERDI, Y., and BATOUCHE, M. C., 2003. Piecewise linear correction of ECG baseline wander: a curve simplification approach. *Computer Methods and Programs in Biomedicine*, 78(1):1–10.
- [BGG98] BURRUS, C. S., GOPINATH, R. A., and GUO, H., 1998. *Introduction to Wavelets and Wavelet Transforms - A Primer* (Prentice Hall), 1st edition.
- [BLDF06] BRECHET, L., LUCAS, M.-F., DONCARLI, C., and FARINA, D., 2006. Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection. *IEEE Transactions on Biomedical Engineering*, 54(3):410–418.
- [BMB⁺03] BENZID, R., MARIR, F., BOUSSAAD, A., BENYOUCEF, M., and ARAR, D., 2003. Fixed percentage of wavelet coefficients to be zeroed for ECG compression. *Electronics Letters*, 39(11):830–831.
- [BMB07] BENZID, R., MARIR, F., and BOUGHECHAL, N.-E., 2007. Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder. *IEEE Signal Processing Letters*, 14(6):373–376.
- [Bra96] BRADIE, B., 1996. Wavelet packet-based compression of single lead ECG. *IEEE Transactions on Biomedical Engineering*, 43(5):493–501.
- [Bra97] BRAUNWALD, E., 1997. *Heart Disease: A Textbook of Cardiovascular Medicine*, volume 1 (W. B. Saunders Company), 5th edition.
- [CHY04] CHEN, W.-S., HSIEH, L., and YUAN, S.-Y., 2004. High performance data compression method with pattern matching for biomedical ECG and arterial pulse waveforms. *Computer Methods and Programs in Biomedicine*, 74(1):11–27.

- [CI98] CHEN, J. and ITOH, S., 1998. A wavelet transform-based ECG compression method guaranteeing desired signal quality. *IEEE Transactions on Biomedical Engineering*, 45(12):1414–1419.
- [CKS⁺06] CHOI, Y., KRAUSE, J., SEO, H., CAPITAN, K., and CHUNG, K., April 2006. Telemedicine in the USA: standardization through information management and technical applications. *IEEE Communications Magazine*, 44(4):41–48.
- [CNFO68] COX, J. R., NOLLE, F. M., FOZZARD, H. A., and OLIVER, G. C., April 1968. AZTEC, a preprocessing program for real-time ECG rhythm analysis. *IEEE Transactions on Biomedical Engineering*, BME-15(2):128–129.
- [CR00] CHAPA, J. O. and RAO, R. M., 2000. Algorithms for designing wavelets to match a specified signal. *IEEE Transactions on Signal Processing*, 48(12):3395–3406.
- [Dau92] DAUBECHIES, I., 1992. *Ten Lectures on Wavelets* (Society for Industrial and Applied Mathematics), 8th edition.
- [Gab78] GABRIEL, K. R., 1978. Least squares approximation of matrices by additive and multiplicative models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(2):186–196.
- [GAG⁺00] GOLDBERGER, A. L., AMARAL, L. A. N., GLASS, L., HAUSDORFF, J. M., IVANOV, P. C., MARK, R. G., MIETUS, J. E., MOODY, G. B., PENG, C.-K., and STANLEY, H. E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):215–220.
- [GFAG08] GONÇALVES, B., FILHO, J. G. P., ANDREÃO, R. V., and GUIZZARDI, G., 2008. ECG Data Provisioning for Telehomecare Monitoring. *23rd ACM Symposium on Applied Computing - ACM SAC 2008, Fortaleza, Brazil*:1374–1379.
- [GJP02] GUPTA, A., JOSHI, S. D., and PRASAD, S., 2002. On a new approach for estimating wavelet matched to signal. In: *Proceedings 8th National Conference on Communications* (Bombay), 180–184.
- [GL96] GOLUB, G. H. and LOAN, C. F. V., 1996. *Matrix Computations* (The Johns Hopkins University Press), 3rd edition.
- [Gol66] GOLOMB, S., 1966. Run-length encoding. *IEEE Transactions on Information Theory*, IT-12:399–401.
- [GS97] G. STRANG, T. N., 1997. *Wavelets and Filter Banks* (Wellesley-Cambridge Press), revised edition.
- [GSK⁺06] GUIDO, R., SLAETS, J., KÖBERLE, R., ALMEIDA, L., and PEREIRA, J., 2006. A new technique to construct a wavelet transform matching a specified signal with applications to digital, real time, spike, and overlap pattern recognition. *Digital Signal Processing*, 16:24–44.

- [HM04] HAO, Y. and MARZILIANO, P., 2004. An efficient wavelet-based pattern matching scheme for ecg data compression. *2004 IEEE International Workshop on Biomedical Circuits and Systems*:S2/4–S5–8.
- [HT86] HAMILTON, P. S. and TOMPKINS, W. J., 1986. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. *IEEE Transactions on Biomedical Engineering*, BME-33(12):1157–1165.
- [Huf52] HUFFMAN, D. A., 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40:1098–1101.
- [IHP01] ISTEPANIAN, R. S. H., HADJILEONTIADIS, L. J., and PANAS, S. M., 2001. ECG data compression using wavelets and higher order statistics methods. *IEEE Transactions on Information Technology in Biomedicine*, 5(2):108–115.
- [IP00] ISTEPANIAN, R. S. H. and PETROSIAN, A. A., 2000. Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system. *IEEE Transactions on Information Technology in Biomedicine*, 4(3):200–211.
- [ISHS83] ISHIJIMA, M., SHIN, S.-B., HOSTETTER, G. H., and SKLANSKY, J., Nov. 1983. Scan-along polygonal approximation for data compression of electrocardiograms. *IEEE Transactions on Biomedical Engineering*, BME-30(11):723–729.
- [JHSC90] JALALEDDINE, S. M. S., HUTCHENS, C. G., STRATTAN, R. D., and COBERLY, W. A., 1990. ECG data compression techniques - A unified approach. *IEEE Transactions on Biomedical Engineering*, 37:329–343.
- [Ken94] KENNEDY, H. L., 1994. *Ambulatory (Holter) Electrocardiography Recordings*. Cardiac Electrophysiology: From Cell to Bedside (W. B. Saunders Company, Philadelphia), 2nd edition.
- [KHO02] KOHLER, B.-U., HENNIG, C., and ORGLMEISTER, R., Jan/Feb 2002. The principles of software QRS detection. *Engineering in Medicine and Biology Magazine, IEEE*, 21(1):42–57.
- [Kom05] KOMIYA, R., 23-25 June 2005. A proposal for telemedicine reference model for future standardization. *Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005. Proceedings of 7th International Workshop on*:224–228.
- [KP04] KWAN, B. and PARAMESRAN, R., 2004. Comparison between legendre moments and DCT in ECG compression. *TENCON 2004. 2004 IEEE Region 10 Conference*, A:167–170.
- [KYL06] KIM, B., YOO, S., and LEE, M., Jan. 2006. Wavelet-based low-delay ECG compression algorithm for continuous ECG transmission. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):77–83.
- [Law90] LAWTON, W. M., 1990. Tight frames of compactly supported affine wavelets. *Journal of Mathematical Physics*:31(8):1898–1901.

- [MYL02] MIAOU, S. G., YEN, H. L., and LIN, C. L., 2002. Wavelet-based ECG compression using dynamic vector quantization with tree codevectors in single codebook. *IEEE Transactions on Biomedical Engineering*, 49(7):671–680.
- [NC93] NAVE, G. and COHEN, A., 1993. ECG compression using long-term prediction. *IEEE Transactions on Biomedical Engineering*, 40(9):877–885.
- [OS99] OPPENHEIM, A. V. and SCHAFER, R., 1999. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series (Prentice Hall, New Jersey), 2nd edition.
- [OWN97] OPPENHEIM, A. V., WILLSKY, A. S., and NAWAB, S. H., 1997. *Signals & Systems* (Prentice Hall, New Jersey), 2nd edition.
- [Phy] *PhysioBank - Physiologic signal archives for biomedical research*. <http://www.physionet.org/physiobank>.
- [Raj02] RAJOUB, B. A., 2002. An efficient coding algorithm for the compression of ECG signals using the wavelet transform. *IEEE Transactions on Biomedical Engineering*, 49(4):355–362.
- [RS97] RAMAKRISHNAN, A. G. and SAHA, S., 1997. ECG coding by wavelet-based linear prediction. *IEEE Transactions on Biomedical Engineering*, 44(12):1253–1261.
- [SABP07a] SILVEIRA, R. M. R., AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2007. Compressão de sinais de eletrocardiogramas com *wavelets* determinadas por otimização genética. *Simpósio Brasileiro de Telecomunicações - SBrT 2007, Recife, Brasil*.
- [SABP07b] SILVEIRA, R. M. R., AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2007. ECG compression based on extracted wavelets. *International Workshop on Telecommunications - IWT'07*.
- [SABP07c] SILVEIRA, R. M. R., AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2007. A genetic algorithm to compress electrocardiograms using parameterized wavelets. *IEEE International Symposium on Signal Processing and Information Technology - ISSPIT 2007, Cairo, Egypt*.
- [SABP08] SILVEIRA, R. M. R., AGULHARI, C. M., BONATTI, I. S., and PERES, P. L. D., 2008. Compressing electrocardiogram signals using parameterized wavelets. *23rd ACM Symposium on Applied Computing - ACM SAC 2008, Fortaleza, Brazil*:1348–1352.
- [Sal07] SALOMON, D., 2007. *Data Compression - The Complete Reference* (Springer), 4th edition.
- [Say06] SAYOOD, K., 2006. *Introduction to Data Compression* (Morgan Kaufmann), 3rd edition.
- [SP96] SAID, A. and PEARLMAN, W. A., 1996. A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250.

- [Vai90] VAIDYANATHAN, P. P., 1990. Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial. *Proceedings of the IEEE*, 78(1):56–93.
- [WCCJ01] WEI, J.-J., CHANG, C.-J., CHOU, N.-K., and JAN, G.-J., Dec 2001. ECG data compression using truncated singular value decomposition. *IEEE Transactions on Information Technology in Biomedicine*, 5(4):290–299.
- [Wel84] WELCH, T. A., 1984. A technique for high-performance data compression. *Computer*, 17(6):8–19.
- [ZCK00] ZIGEL, Y., COHEN, A., and KATZ, A., 2000. The weighted diagnostic distortion (WDD) measure for ECG signal compression. *IEEE Transactions on Biomedical Engineering*, 47(11):1422–1430.
- [ZL77] ZIV, J. and LEMPEL, A., 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3):337–343.
- [ZLP00] Z. LU, D. Y. K. and PEARLMAN, W. A., 2000. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. *IEEE Transactions on Biomedical Engineering*, 47(7):849–856.
- [ZT93] ZOU, H. and TEWFIK, A. H., 1993. Parametrization of compactly supported orthonormal wavelets. *IEEE Transactions on Signal Processing*, 41(3):1428–1431.