
Projeto e Avaliação de Dicionários para Quantização Vetorial de Voz e Imagem

Francisco Madeiro Bernardino Junior

Tese de Doutorado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no domínio da Engenharia Elétrica.

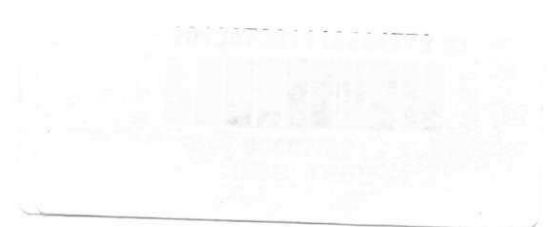
Área de Concentração: Processamento da Informação

Benedito Guimarães Aguiar Neto, Dr.-Ing., UFPB
Orientador

Marcelo Sampaio de Alencar, Ph.D., UFPB
Orientador

Campina Grande, Paraíba, Brasil

©Francisco Madeiro Bernardino Junior, Dezembro de 2001





B523p Bernardino Junior, Francisco Madeiro

Projeto e avaliação de dicionários para quantização vetorial de voz e imagem/Francisco Madeiro Bernardino Junior.-Campina Grande: UFPB: DEE, 2001.

208p.:il.-

Inclui bibliografia

Tese (doutorado) - UFPB/CCT/DEE

1-Quantização vetorial 2-Processamento digital de voz 3-Processamento digital de imagem
4-Processamento da informação

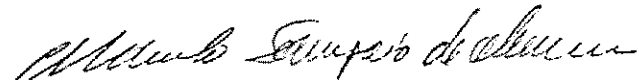
CDU 621.3:681.3

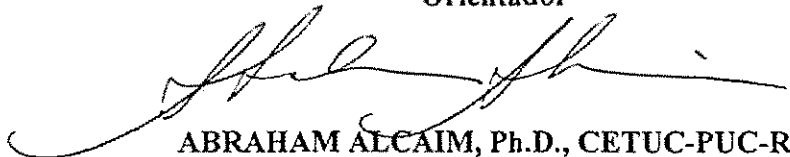
PROJETO E AVALIAÇÃO DE DICIONÁRIOS PARA QUANTIZAÇÃO VETORIAL
DE VOZ E IMAGEM

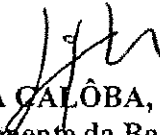
FRANCISCO MADEIRO BERNARDINO JUNIOR


Tese Aprovada em 14.12.2001


BENEDITO GUIMARÃES AGUIAR NETO, Dr.-Ing., UFPB
Orientador


MARCELO SAMPAIO DE ALENCAR, Ph.D., UFPB
Orientador


ABRAHAM ALCAIM, Ph.D., CETUC-PUC-Rio
Componente da Banca


LUIZ PEREIRA GALÔBA, Ph.D., UFRJ
Componente da Banca


RICARDO MENEZES CAMPELLO DE SOUSA, Ph.D., UFPE
Componente da Banca


JOÃO MARQUES DE CARVALHO, Ph.D., UFPB
Componente da Comissão

CAMPINA GRANDE - PB
Dezembro - 2001

Dedicatória

Dedico este trabalho aos meus pais, Madeiro e Leni,
aos meus irmãos, Charles e Ricardo,
e à minha noiva, Ana Cristina.

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos a algumas pessoas e instituições que contribuíram para a realização deste trabalho.

Aos meus pais, a quem devo minhas principais conquistas, cujo carinho foi de fundamental importância nos momentos difíceis e cuja compreensão constituirá sempre minha referência de sublimidade.

À minha noiva Ana Cristina, cuja presença alegria meu dia a dia e motiva minha obstinação, pelo seu amor, companheirismo e apoio.

Aos meus irmãos, Charles e Ricardo, pelo inestimável afeto, e à vovó Fransquinha, por suas orações e por representar um referencial de sabedoria e força de vontade.

À Professora Rosângela Maria Vilar, principal responsável pela direção tomada por este trabalho, a quem expressei meus votos de gratidão por suas críticas e sugestões, pelo estímulo e, principalmente, por sua amizade.

Ao professor Benedito Guimarães Aguiar Neto, pela amizade e pela orientação do trabalho, por haver despertado em mim o interesse por Processamento de Sinais de Voz e por ter possibilitado a consecução de realizações importantes.

Ao professor Marcelo Sampaio de Alencar, pela orientação do trabalho, pelo apoio dado a iniciativas relacionadas à vida acadêmica e por sua amizade.

Ao professor Francisco Marcos de Assis, pelo incentivo dado ao estudo de *wavelets*.

Aos meus amigos Waslon Terlizzi Araújo Lopes, Joseana Macêdo Fachine, Marcos Ricardo Alcântara Moraes e Madhavan Srinivasan Vajapeyam, com os quais tive a satisfação de interagir para o desenvolvimento de trabalhos interessantes.

À Professora Maria de Fátima Farias, pela colaboração dada à aquisição de alguns artigos.

Aos professores Ricardo Menezes Campello de Souza, João Marques de Carvalho, Luiz Pereira Calôba e Abraham Alcaim, pelas contribuições dadas ao trabalho.

A Rodrigo de Lamare, pelo provimento dos conjuntos de treino e de avaliação referentes aos parâmetros LSF.

Ao meu amigo Geovany Araújo, por seu incentivo e sua presteza, e a Antônio Campos, pelo apoio.

Aos companheiros do LAPS e LABCOM: Luciana, Edmar, Leonel, Rinaldo, Paulo Márcio, Suzete, Juraci, Leocarlos, Ronaldo, Vânia, Robson, Walter, Josemar, Sérgio, Wamberto, Gustavo, Lidiano, Eustáquio e George.

À família Ribeiro Veloso, por contribuir sobremaneira para tornar agradáveis os dias em Campina Grande, longe de minha família.

Aos meus grandes amigos Murilo, Fábio, Daniel, Lucas, Cassandra, Maryanne e Marcus, que sempre me apoiaram e incentivaram.

Aos funcionários da COPELE, Ângela, Pedro e Marcos, e aos funcionários da ATE-CEL, Joab e Toinho, pela presteza.

À Universidade Federal da Paraíba - Campus II e à CAPES, órgão financiador deste trabalho.

A Deus, por minha vida ser feliz e repleta de realizações.

Resumo

Neste trabalho são apresentadas técnicas para projeto de dicionários aplicados à quantização vetorial (QV) de sinais de voz e imagens. A primeira, denominada SOA (*self-organizing algorithm*), é inspirada no algoritmo de Kohonen. Utiliza, no entanto, um paradigma de vizinhança diferente do proposto por Kohonen para atualização dos vetores-código. A segunda, denominada SSC (*synaptic space competitive*), corresponde a um algoritmo que utiliza aprendizagem competitiva simples. A terceira, denominada FS-SSC (*frequency sensitive SSC*), é resultante da introdução do princípio de consciência de Grossberg no algoritmo SSC. A quarta técnica, denotada por PCA (como alusão a *principal component analysis*), utiliza os autovalores e os autovetores (componentes principais) da matriz de covariância de um sinal de voz para calcular os vetores-código do dicionário. São apresentados resultados concernentes à codificação de sinais de voz e imagem baseada em QV simples (convencional) como também em QV no domínio *wavelet*, além de resultados referentes ao reconhecimento de locutores baseado em QV paramétrica. Os resultados mostram que os algoritmos SOA, SSC, FS-SSC e PCA apresentam-se como alternativas ao tradicional algoritmo LBG (Linde-Buzo-Gray).

É investigada a complexidade computacional dos algoritmos SSC e LBG. São obtidas expressões analíticas (em função do tamanho do dicionário, da dimensão dos seus vetores-código, do número de vetores do conjunto de treino e do número de iterações desses algoritmos) que estabelecem as condições que devem ser obedecidas para que o algoritmo SSC seja mais eficiente que o algoritmo LBG quanto ao número de operações (divisões, multiplicações, comparações, adições e subtrações) realizadas durante o projeto do dicionário.

O presente trabalho também apresenta um método para redução da complexidade computacional da etapa de codificação da QV. O método proposto utiliza a organização estruturada dos dicionários PCA para reduzir significativamente o número de operações realizadas na etapa de determinação do vizinho mais próximo de cada vetor a ser codificado bem como para minimizar os requisitos de memória exigida para armazenar o dicionário.

Abstract

This work presents techniques for designing codebooks applied to vector quantization (VQ) of speech signals and images. The first technique, referred to as SOA (self-organizing algorithm), is inspired on Kohonen's algorithm. The unsupervised learning algorithm SOA, however, uses a neighborhood paradigm which differs from that of Kohonen for updating the codevectors. The second, referred to as SSC (synaptic space competitive), corresponds to an algorithm that uses competitive learning. The third, referred to as FS-SSC (frequency sensitive SSC), introduces Grossberg's conscience principle on SSC algorithm. The fourth technique, referred to as PCA (as an allusion to principal component analysis), computes the VQ codebooks taking into account the eigenvalues and the eigenvectors (principal components) of the covariance matrix of a speech signal. This work presents results concerning speech and image coding based upon simple (conventional) VQ and based upon wavelet VQ, as well as results regarding speaker recognition based upon parametric VQ. Results show that the algorithms SOA, SSC, FS-SSC and PCA are alternatives to the traditional LBG (Linde-Buzo-Gray) algorithm.

The computational complexity of the algorithms SSC and LBG is investigated. Analytical expressions (as a function of the codebook size, the dimension of the codevectors, the number of vectors in the training set and the number of iterations executed for codebook design) are derived for the number of operations (multiplications, divisions, additions, subtractions and comparisons) executed by SSC and LBG. Constraints are obtained under which the SSC algorithm is more efficient than the LBG algorithm in terms of number of operations executed in codebook design.

The work also presents a method for reducing the computational complexity of the minimum distortion encoding (MDE) of VQ. The proposed method uses the structured organization of the PCA codebooks for significantly reducing the number of operations executed in the process of determining the nearest neighbor for each source vector to be coded, as well as for reducing the memory requirements for codebook storage.

Lista de Símbolos e Abreviaturas

LSF	freqüências espectrais de linha (<i>line spectral frequencies</i>) ¹
LAPS	Laboratório de Automação e Processamento de Sinais
LABCOM	Laboratório de Comunicações
COPELE	Coordenação de Pós-Graduação em Engenharia Elétrica
ATECEL	Associação Técnico-científica Ernesto Luiz de Oliveira Junior
CAPES	Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
QV	quantização vetorial
SOA	<i>self-organizing algorithm</i>
SSC	algoritmo competitivo no espaço sináptico (<i>synaptic space competitive</i>)
FS-SSC	algoritmo SSC sensível à freqüência (<i>frequency sensitive SSC</i>)
PCA	análise de componentes principais (<i>principal component analysis</i>)
LBG	Linde-Buzo-Gray
VQ	<i>vector quantization</i>
MDE	codificação por distorção mínima (<i>minimum distortion encoding</i>)
DWT	transformada <i>wavelet</i> discreta (<i>discrete wavelet transform</i>)
DWT+QV	quantização vetorial dos coeficientes <i>wavelet</i>
MOS	escore médio de opinião (<i>mean opinion score</i>)
LPC	codificação por predição linear (<i>linear prediction coding</i>)
KLT	transformada de Karhunen-Loève (<i>Karhunen-Loève transform</i>)
ISDN	redes digitais de serviços integrados (<i>integrated services digital networks</i>)
Q	mapeamento da quantização vetorial

¹Na literatura, o símbolo LSP (*line spectral pairs*) também é usado em lugar de LSF.

\mathbf{x}	vetor de entrada, vetor de treino, padrão de entrada, padrão de treino, vetor a ser quantizado ²
K	dimensão do quantizador vetorial, ou ainda, número de coeficientes delta cepstrais calculados em um bloco de amostras, ou então, tamanho de uma janela de ponderação ³
R^K	espaço euclidiano K -dimensional
W	Dicionário do quantizador vetorial
w_i	i -ésimo vetor-código do dicionário W , ou ainda, vetor de pesos sinápticos do i -ésimo neurônio de um algoritmo não-supervisionado ⁴
N	tamanho do dicionário (número de vetores-código)
S_i	i -ésima célula de Voronoi
$d(\mathbf{x}, w_i)$	distorção/distância entre os vetores \mathbf{x} e w_i
b_i	palavra-binária ⁵ correspondente ao vetor-código w_i
$C(\mathbf{x})$	regra de codificação aplicada ao vetor \mathbf{x}
$b_l(I)$	l -ésimo bit ($1 \leq l \leq m$) da palavra-binária b_I , de comprimento m bits
$D(b_I)$	regra de decodificação aplicada à palavra-binária b_i
R	taxa de codificação
bpp	bits por <i>pixel</i>
PNN	<i>pair-wise nearest neighbor</i>
GLA	algoritmo de Lloyd generalizado (<i>generalized Lloyd algorithm</i>)
W_0	dicionário inicial
\mathbf{x}_m	m -ésimo vetor do conjunto de treino \mathbf{X}
M	número de vetores do conjunto de treino, ou ainda, número de linhas de uma imagem com $M \times M$ <i>pixels</i> ⁶
W_n	dicionário na n -ésima iteração
D_n	distorção ao final da n -ésima iteração do algoritmo LBG

²O símbolo \mathbf{x} pode ser usado para denotar um vetor pertencente a um conjunto de treino utilizado por um algoritmo de projeto de dicionário. Pode denotar também um vetor a ser codificado. O significado fica claro no contexto.

³O significado de K fica claro no contexto em que esse símbolo é utilizado.

⁴Ao se utilizar um algoritmo não supervisionado (Kohonen, SSC, SOA) para projeto de quantizadores vetoriais, o vetor de pesos sinápticos w_i corresponde ao i -ésimo vetor-código (vetor de reconstrução) do dicionário.

⁵Em se tratando de transmissão por canal ruidoso, b_i denota a palavra-binária enviada pelo codificador, enquanto que b_j denota a palavra-binária recebida pelo decodificador.

⁶O significado de M fica claro no contexto.

D_n	distorção média ao final da n -ésima iteração do algoritmo LBG
ϵ	limiar de distorção pré-estabelecido (critério de parada do algoritmo LBG), ou então, probabilidade de erro de bit em um canal binário simétrico ⁷
w_{ij}	j -ésima componente do vetor-código w_i , ou ainda, j -ésimo peso sináptico do vetor de pesos do i -ésimo neurônio de um algoritmo não-supervisionado ⁸
x_j	j -ésima componente do vetor de entrada (ou vetor de treino, padrão de treino, padrão de entrada) x
TSVQ	quantização vetorial estruturada em árvore (<i>tree-structured vector quantization</i>)
PDS	algoritmo de busca por distância parcial (<i>partial distance search</i>)
d_{min}	distância/distorção mínima
PDS-O	algoritmo de busca por distância parcial com ordenamento
p_i	probabilidade de que um dado vetor de entrada x pertença à região de Voronoi S_i ; probabilidade <i>a priori</i> de o vetor w_i ser selecionado como a reconstrução de x
S	longo conjunto representativo de vetores para treinamento do quantizador vetorial
x_s	s -ésimo vetor do conjunto de treino S
m	número de vetores do conjunto de treino ⁹
m_i	número de vetores de treino pertencentes à célula de Voronoi S_i
1-D	unidimensional
2-D	bidimensional
$h_0(n)$	filtro <i>wavelet</i> passa-baixa de análise
$h_1(n)$	filtro <i>wavelet</i> passa-alta de análise
$g_0(n)$	filtro <i>wavelet</i> passa-baixa de síntese
$g_1(n)$	filtro <i>wavelet</i> passa-alta de síntese
$x(n)$	sinal de entrada ou ainda, simplesmente, sinal original

⁷O significado de ϵ fica claro no contexto em que esse símbolo é utilizado.

⁸Ao se utilizar um algoritmo não supervisionado (Kohonen, SSC, SOA) para projeto de quantizadores vetoriais, o j -ésimo peso sináptico do vetor de pesos sinápticos w_i corresponde à j -ésima componente do vetor-código (vetor de reconstrução) do dicionário.

⁹O símbolo m também é usado para denotar o número de bits de uma palavra-binária que representa um vetor-código. O significado de m fica claro no contexto em que é usado.

$a_i(n)$	componente de aproximação (obtida por meio de DWT) no i -ésimo nível de resolução
$d_i(n)$	componente de detalhe (obtida por meio de DWT) no i -ésimo nível de resolução
IDWT	transformada <i>wavelet</i> discreta inversa (<i>inverse discrete wavelet transform</i>)
$y(n)$	signal reconstruído/recuperado (obtido por meio de IDWT), ou ainda, simplesmente, signal processado
L	número de níveis de decomposição (número de níveis de resolução) da DWT
L	filtro passa-baixa, ou ainda, número de componentes de f (com $L < K$) usadas na representação do vetor x , ou então, número de vetores z_i ao longo de cujas direções os vetores-código devem ser alocados no algoritmo PCA ¹⁰
H	filtro passa-alta, ou ainda, entropia dos vetores-código ¹¹
HL_1	sub-banda correspondente à direção horizontal do 1 ^o nível de resolução, também denotada por S_{11}
LH_1	sub-banda correspondente à direção vertical do 1 ^o nível de resolução, também denotada por S_{12}
HH_1	sub-banda correspondente à direção diagonal do 1 ^o nível de resolução, também denotada por S_{13}
HL_2	sub-banda correspondente à direção horizontal do 2 ^o nível de resolução, também denotada por S_{21}
LH_2	sub-banda correspondente à direção vertical do 2 ^o nível de resolução, também denotada por S_{22}
HH_2	sub-banda correspondente à direção diagonal do 2 ^o nível de resolução, também denotada por S_{23}
HL_3	sub-banda correspondente à direção horizontal do 3 ^o nível de resolução, também denotada por S_{31}
LH_3	sub-banda correspondente à direção vertical do 3 ^o nível de resolução, também denotada por S_{32}

¹⁰O significado de L fica claro no contexto em que é usado.

¹¹O significado de H fica claro no contexto.

HH_3	sub-banda correspondente à direção diagonal do 3 ^o nível de resolução, também denotada por S_{33}
LL_3	sub-banda correspondente à componente de aproximação (resultante de filtragens passa-baixa) do 3 ^o nível de resolução, também denotada por S_{30}
S_{ij}	sub-banda correspondente à j -ésima direção ($j = 1$ denota direção horizontal, $j = 2$ denota direção vertical, $j = 3$ denota direção diagonal e $j = 0$ denota direção LL) do i -ésimo nível de resolução (nível de decomposição)
DCT	transformada discreta de cosseno (<i>discrete cosine transform</i>)
JPEG	<i>joint photographics experts group</i>
b_{ij}	número de bits alocados para cada sub-banda S_{ij} da decomposição <i>wavelet</i>
Y_p	p -ésima ($1 \leq p \leq P$) variável aleatória
R_p	taxa alocada para Y_p
D_p	distorção média para o p -ésimo quantizador ¹²
HVS	sistema visual humano (<i>human visual system</i>)
PQS	<i>picture quality scale</i>
ECVQ	quantização vetorial com restrição de entropia (<i>entropy-constrained vector quantization</i>)
MSVQ	quantização vetorial multi-estágio (<i>multistage vector quantization</i>)
PVQ	quantização vetorial preditiva (<i>predictive vector quantization</i>)
CVQ	quantização vetorial classificada (<i>classified vector quantization</i>)
HVQ	quantização vetorial hierárquica (<i>hierarchical vector quantization</i>)
$K_1 \times K_2$	tamanho do bloco ¹³ ; define a dimensão do quantizador vetorial
QVR	quantização vetorial robusta
AI	atribuição de índices
X	seqüência de vetores x ou fonte a ser codificada
\hat{x}	versão reconstruída (quantizada vetorialmente) do vetor x
p_x	função densidade de probabilidades K -dimensional da fonte X

¹² $D_p = E[(Y_p - Q(Y_p))^2]$, em que $Q(Y_p)$ denota a saída do quantizador para Y_p .

¹³Em QV convencional de imagens, denota o tamanho do bloco de *pixels*. Em codificação de imagens utilizando DWT+QV, do tipo intra-banda, denota o tamanho do bloco de coeficientes *wavelet* de uma sub-banda.

$P_{j i}$	probabilidade de o decodificador receber b_j dado que b_i foi enviado pelo codificador
D	distorção total introduzida ao se transmitir a fonte X através de um canal ruidoso
D_Q	distorção de quantização
D_C	distorção devida aos erros de canal
s	arranjo (ordenamento) de dicionário
$I_{\text{dis}}(s)$	índice de desordem do dicionário com ordenamento s
π	função de atribuição de índices do dicionário W
Π_N	conjunto de todas funções de permutação $\pi : \{0, 1\}^m \rightarrow \{0, 1\}^m$
b_i'	palavra-binária resultante do mapeamento introduzido com a aplicação de uma permutação π , isto é, $b_i' = \pi(b_i)$
B	conjunto ordenado de palavras-binárias
s'	ordenamento resultante da aplicação de uma atribuição de índices (aplicação de uma permutação π) no dicionário com ordenamento s , isto é, $s' = \pi(s)$
B'	conjunto ordenado de palavras-binárias resultante da aplicação de uma atribuição de índices (aplicação de uma permutação π) no dicionário com conjunto ordenado de palavras-binárias B , isto é, $B' = \pi(B)$
NP	não polinomial
SA	<i>simulated annealing</i>
SNR	relação sinal-ruído (<i>signal-to-noise ratio</i>)
SNRseg	relação sinal-ruído segmental (<i>segmental signal-to-noise ratio</i>)
SD	distorção espectral (<i>spectral distortion</i>)
PSNR	relação sinal-ruído de pico (<i>peak signal-to-noise ratio</i>)
s_l	escore atribuído pelo l -ésimo avaliador ($1 \leq l \leq L$) no teste MOS
$e(n)$	n -ésima amostra do sinal erro, correspondente à diferença entre a n -ésima amostra do sinal original $x(n)$ e a n -ésima amostra do sinal processado $y(n)$, ou ainda, n -ésima amostra do sinal erro de predição, correspondente à diferença entre a n -ésima amostra do sinal de voz original $s(n)$ e a n -ésima amostra do sinal de voz estimado $\hat{s}(n)$ no processo de predição ¹⁴
E_x	energia contida no sinal original

¹⁴O significado de $e(n)$ fica claro no contexto em que é utilizado.

E_e	energia contida no sinal erro
$\text{SNR}(j)$	relação sinal-ruído convencional (total) para o j -ésimo segmento (janela de tempo) do sinal
m_j	instantes finais para a j -ésima janela de tempo ($0 \leq j \leq J-1$) utilizada para determinação de SNRseg
N_A	número de amostras de cada janela de tempo utilizada para determinação de SNRseg
SNRtot	relação sinal-ruído total
v_p	valor de pico da amplitude de entrada
MSE	erro médio quadrático (<i>mean square error</i>)
$F(l, c)$	valor de <i>pixel</i> referente à l -ésima linha e c -ésima coluna da imagem original
$\hat{F}(l, c)$	valor de <i>pixel</i> referente à l -ésima linha e c -ésima coluna da imagem reconstruída
HMMs	modelos de Markov escondidos (<i>hidden Markov models</i>)
DTW	alinhamento dinâmico no tempo (<i>dynamic time warping</i>)
M_L	número de locutores cadastrados pelo sistema de identificação de locutores
$\text{DICI}(m)$	dicionário projetado para o m -ésimo locutor ($1 \leq m \leq M_L$) cadastrado pelo sistema de reconhecimento de locutores
$\text{ERRO}(m)$	erro de quantização introduzido ao se representar uma seqüência de vetores acústicos pelo dicionário $\text{DICI}(m)$
CELP	codificação por excitação linear preditiva (<i>code-excited linear prediction</i>)
VSELP	codificação com excitação linear preditiva por soma de vetores (<i>vector sum excited linear prediction</i>)
$s(n)$	sinal de voz discreto no tempo
$\hat{s}(n)$	sinal de voz estimado por meio da predição linear
c_k	k -ésimo coeficiente ($k = 1, 2, \dots, K$) do preditor de ordem K , ou k -ésimo coeficiente LPC
$h(n)$	resposta ao impulso do preditor
$H_K(z)$	função de transferência do preditor
σ_e^2	variância do erro de estimação
R_{ss}	função de autocorrelação do sinal $s(n)$

$c_i(n)$	n -ésimo coeficiente LPC calculado no i -ésimo bloco de amostras
$ce_i(n)$	n -ésimo coeficiente cepestral calculado no i -ésimo bloco de amostras no qual os coeficientes LPC foram previamente determinados
$\Delta ce_i(n)$	n -ésimo coeficiente delta cepestral calculado no i -ésimo bloco de amostras no qual os coeficientes cepestrais foram previamente determinados
G	termo de ganho utilizado na determinação dos coeficientes delta cepestrais
$2Q + 1$	número de blocos de amostras sobre os quais o cálculo dos coeficientes delta cepestrais é realizado
$\Delta cp_i(n)$	n -ésimo coeficiente delta cepestral ponderado calculado no i -ésimo bloco de amostras no qual os coeficientes delta cepestrais foram previamente determinados
$jp(n)$	janela de ponderação utilizada para determinação dos coeficientes delta cepestrais ponderados
BPL	<i>bandpass liftering</i>
VLSI	<i>very-large-scale-integrated</i>
w_i	neurônio/vetor vencedor
$\mathcal{N}_{w_{i^*}}$	vizinhança em torno do nó (no arranjo topológico utilizado pelo algoritmo de Kohonen) correspondente ao neurônio vencedor
$r_g(n)$	raio de vizinhança (no n -ésimo passo) medido em um arranjo topológico de nós do algoritmo de Kohonen
$d_g(\cdot)$	distância medida em um arranjo topológico de nós do algoritmo de Kohonen
Δw_{ij}	modificação introduzida na j -ésima componente (peso sináptico) do neurônio w_i
$\eta(n)$	taxa de aprendizagem ou ganho de adaptação (com $0 < \eta < 1$) no n -ésimo passo (do algoritmo de Kohonen) ou na n -ésima iteração (dos algoritmos SOA e SSC)
$\mathcal{O}_i(i^*, n)$	função que define a vizinhança em torno do nó (no arranjo topológico) correspondente ao neurônio vencedor w_{i^*} (do algoritmo de Kohonen)
\mathcal{N}_x	vizinhança (hiper-esfera do algoritmo SOA) centrada em torno do vetor de treino x , definida no espaço de padrões R^K
$r_e(n)$	raio de vizinhança ou, mais precisamente, raio (na n -ésima iteração) da hiper-esfera do algoritmo SOA, medido no espaço de padrões R^K

$d_e(\cdot)$	distância euclidiana, medida no espaço de padrões R^K
O_x	função que define a vizinhança (do algoritmo SOA) em torno do vetor de treino x
n_{tot}	número total de passagens do conjunto de treino (número total de iterações)
n_f	iteração a partir da qual o raio de vizinhança do algoritmo SOA é zero, ou seja, número de iterações da primeira fase do algoritmo SOA
$\eta(1)$	taxa de aprendizagem inicial
$\eta(n_{tot})$	taxa de aprendizagem final
$\tau_c(1)$	raio de vizinhança inicial do algoritmo SOA
SSN	algoritmo <i>synaptic space neighborhood</i> ¹⁵
VES	algoritmo de Kohonen no espaço sináptico (V denota vizinhança)
\mathcal{N}_c	vizinhança (hiper-esfera do algoritmo SSN) centrada em torno do vetor vencedor, definida no espaço de padrões R^K
SSN-TV	<i>Kohonen's algorithm with spherical neighborhood around training vector; synaptic space neighborhood - training vector</i>
VES-VT	algoritmo de Kohonen com vizinhança no vetor de treino; vizinhança no espaço sináptico - vetor de treino
KMTAU	algoritmo de Kohonen modificado com taxa de aprendizagem uniforme
MKOH	<i>modified Kohonen's algorithm</i>
KMVVT	algoritmo de Kohonen modificado com vizinhança centrada no vetor de treino
n_{SSC}	número de iterações do algoritmo SSC
$w_{i^*}(n, m)$	vencedor quando da apresentação do m -ésimo vetor de treino na n -ésima iteração
i^*	índice do vencedor
$x(m)$	m -ésimo vetor do conjunto de treino
$w_i(n, m)$	i -ésimo vetor-código quando da apresentação do m -ésimo vetor de treino na n -ésima iteração
$\Delta w_{i^*j}(n, m)$	modificação introduzida no vencedor quando da apresentação do m -ésimo vetor de treino na n -ésima iteração
$x_j(m)$	j -ésima componente do vetor $x(m)$

¹⁵Esse algoritmo também é denotado por SSN-SV, em que SV é uma alusão a *synaptic winning vector*.

$w_{i \cdot j}(n, m)$	j -ésima componente do vetor $w_{i \cdot}(n, m)$
$\tilde{w}_{i \cdot j}$	versão atualizada da j -ésima componente do vencedor
$\eta(n_{SSC})$	taxa de aprendizagem final do algoritmo SSC
CES	algoritmo competitivo no espaço sináptico
FSCCL	aprendizagem competitiva sensível à frequência (<i>frequency sensitive competitive learning</i>)
f_i	número de vezes em que o i -ésimo vetor-código foi até então escolhido como vencedor no algoritmo FS-SSC
$\hat{d}[\cdot]$	medida de distorção utilizada pelo algoritmo FS-SSC
CV	coeficiente de variação
\bar{X}	média de um conjunto de n observações
σ_X	desvio-padrão de um conjunto de n observações
db6	família <i>wavelet</i> Daubechies 6
db12	família <i>wavelet</i> Daubechies 12
R_{d_i}	taxa de codificação do quantizador vetorial alocada para o sinal $d_i(n)$, com $1 \leq i \leq \mathcal{L}$
$R_{a_{\mathcal{L}}}$	taxa de codificação do quantizador vetorial alocada para o sinal $a_{\mathcal{L}}(n)$
R_T	taxa de codificação total
N_x	número de multiplicações por amostra
$S_{\mathcal{L}0}$	componente (sub-banda) de aproximação no nível de resolução mais baixa (nível de resolução mais grosseira)
$R_{S_{ij}}$	taxa de codificação alocada para a sub-banda S_{ij}
$R_{S_{\mathcal{L}0}}$	taxa de codificação alocada para a sub-banda $S_{\mathcal{L}0}$
QE	quantização escalar
SPIHT	<i>set partitioning in hierarchical trees</i>
EZW	<i>embedded zerotree wavelet</i>
AR(1)	fonte autoregressiva de 1 ^a ordem
$\{X(n)\}$	processo discreto de Gauss-Markov
$\{W(n)\}$	seqüência de variáveis aleatórias com distribuição gaussiana, independentes e identicamente distribuídas, com média zero
$h_a(n)$	resposta ao impulso do filtro utilizado para obtenção do processo de Gauss-Markov
a	coeficiente de correlação do processo de Gauss-Markov
$s_p(n)$	signal de voz obtido após a pré-ênfase

$w_h(n)$	janela de Hamming
f_s	freqüência de amostragem
SRL	sistema de reconhecimento de locutor
CEP	coeficientes cepestrais
CEP-P	coeficientes cepestrais ponderados
DCEP	coeficientes delta cepestrais
DCEP-P	coeficientes delta cepestrais ponderados
LF	(grupo de) de locutores femininos
LM	(grupo de) de locutores masculinos
TI	taxa de identificação
TFA	taxa de falsa aceitação
n_{LBG}	número total de iterações realizadas pelo algoritmo LBG
\tilde{w}_i	centróide da i -ésima classe
M_i	número de vetores de treino alocados em S_i
\tilde{w}_{ij}	j -ésima componente do centróide \tilde{w}_i
x_{mj}	j -ésima componente do vetor de treino x_m
ad.	adições
sub.	subtrações
div.	divisões
mult.	multiplicações
comp.	comparações
$A(K, N)$	expressão correspondente a $\frac{[(K-1)(1+N)+3]}{(K-1)N+K}$
z_i	i -ésimo vetor ($1 \leq i \leq K$), pertencente a um conjunto de vetores independentes, ou ainda, i -ésimo autovetor
f	vetor de atributos ¹⁶ , isto é, vetor que contém os escalares que contribuem (que multiplicam os vetores z_i) para a representação de um determinado vetor observado x
f_i	i -ésimo atributo ($1 \leq i \leq K$) do vetor f
Z^T	matriz quadrada de ordem K cujas colunas são os vetores z_i
δ_{ij}	função de Kronecker
I_K	matriz identidade de ordem K
C_{xx}	matriz de autocovariância de x
λ_i	i -ésimo autovalor ($1 \leq i \leq K$)

¹⁶O vetor f é uma transformação ortogonal do vetor x .

E_{\min}	erro médio quadrático mínimo
σ_x^2	variância de x
R_{ff}	matriz de autocorrelação de f
R_{xx}	matriz de autocorrelação de x
T	operação de transposição de uma matriz
C_{xx}	matriz de autocovariância do sinal de voz X
$\hat{\lambda}_i$	valor percentual relativo do autovalor λ_i
\tilde{z}_i	vetor obtido de z_i por meio de uma operação de normalização
τ_i	escalar utilizado para normalização de z_i - corresponde ao recíproco do valor absoluto da componente de maior valor absoluto do vetor z_i
N_i	Número de vetores-código a serem alocados (pelo algoritmo PCA) ao longo da direção definida pelo vetor \tilde{z}_i
w_{i,n_i}	n_i -ésimo vetor alocado (pelo algoritmo PCA) ao longo da i -ésima direção
$f(n_i, \lambda_i)$	escalar determinado (pelo algoritmo PCA) assumindo-se que cada direção principal apresenta distribuição gaussiana com variância $\sigma_i^2 = \lambda_i$
PCA-ORI	versão original do algoritmo PCA
\tilde{H}	entropia normalizada dos vetores-código
D_{PCA}	inicialização do algoritmo LBG usando dicionários PCA
N_{POS}	número de vetores-código com média positiva
N_{ZERO}	número de vetores-código com média zero
N_{NEG}	número de vetores-código com média negativa
$w_{i,PZ}$	vetor-código cujas componentes têm valor médio é positivo ou zero
PZ	(valor médio) positivo ou zero
$w_{i,NZ}$	vetor-código cujas componentes têm valor médio é negativo ou zero
NZ	(valor médio) negativo ou zero
$w_{N+1-i,NZ}$	vetor do tipo NZ cujo índice é $N + 1 - i$
média(x)	média das componentes do vetor x
1/2PCA	método proposto para redução da complexidade computacional da etapa de codificação da QV, utilizando para tanto apenas metade do dicionário previamente projetado com o algoritmo PCA
soma(x)	soma das componentes do vetor x
BT	algoritmo de busca total
d_8	distância entre os vetores x e w_8

d_g	distância entre os vetores \mathbf{x} e \mathbf{w}_g
BSC	canal binário simétrico (<i>binary symmetric channel</i>)
$RP_{\text{SNRseg}}(\epsilon)$	redução percentual (em relação a uma transmissão sem erro) da relação sinal-ruído segmental (SNRseg) do sinal reconstruído após transmissão por um canal BSC com probabilidade de erro de bit igual a ϵ
$\text{SNRseg}(\text{tsr})$	SNRseg do sinal reconstruído após transmissão sem erro
$\text{SNRseg}(\epsilon)$	SNRseg do sinal reconstruído após transmissão por um canal BSC com probabilidade de erro de bit igual a ϵ
$b_1(i)$	primeiro bit ou bit mais significativo da palavra \mathbf{b}_i
MSB	bit mais significativo (<i>most significant bit</i>)
$b_m(i)$	bit menos significativo da palavra \mathbf{b}_i de $m = \log_2 N$ bits
LSB	bit menos significativo (<i>least significant bit</i>)

Lista de Figuras

2.1	Partição do espaço euclidiano bidimensional, R^2 , introduzido pelo mapeamento dos vetores de entrada \mathbf{x} nos vetores-código \mathbf{w}_i . As coordenadas x_1 e x_2 representam a primeira e a segunda componentes do vetor $\mathbf{x} \in R^2$, respectivamente.	8
2.2	Codificação/decodificação em um sistema de codificação baseado em quantização vetorial simples.	9
2.3	Algoritmo PDS.	14
2.4	Estrutura básica do banco de filtros e das operações de decimação e interpolação envolvidos na DWT.	16
2.5	Estrutura piramidal de uma decomposição <i>wavelet</i> multiresolucional em 3 níveis: (a) operação de análise ou decomposição; (b) operação de síntese ou reconstrução.	18
2.6	Decomposição <i>wavelet</i> de 3 níveis de uma imagem 256×256 . A filtragem passa-baixa é denotada por L e a passa-alta por H	19
2.7	Sub-bandas obtidas em uma decomposição <i>wavelet</i> de 3 níveis de uma imagem 256×256 . A sub-banda S_{ij} diz respeito à j -ésima direção do i -ésimo nível de decomposição, $i, j \in \{1, 2, 3\}$. Observe que $j = 0$ diz respeito à sub-banda correspondente a LL	20
2.8	Sistema de compressão de imagens que utiliza DWT e QV.	24
2.9	Exemplo de QV inter-banda do tipo mesma orientação.	26
2.10	Exemplo de QV inter-banda do tipo mesmo nível.	26
2.11	Modelo do sistema de comunicação.	28
2.12	Etapas do processo de reconhecimento de locutor baseado em quantização vetorial paramétrica.	39
2.13	Diagrama de blocos da fase de treinamento do sistema de reconhecimento (identificação) de locutor.	39

2.14	Diagrama de blocos da fase de identificação do sistema de reconhecimento de locutor.	41
2.15	Estrutura geral do processo de predição linear: $h(n)$ representa a resposta ao impulso do preditor, $s(n)$ representa o sinal de voz discreto no tempo, $\hat{s}(n)$ denota o sinal de voz estimado por meio da predição linear e $e(n)$ denota o erro de estimação.	44
3.1	Aprendizagem supervisionada.	50
3.2	Vizinhança $\mathcal{N}_{\mathbf{x}}$ centrada no vetor de treino \mathbf{x} . A vizinhança no algoritmo SOA é uma hiper-esfera definida no próprio espaço de padrões, neste caso o espaço R^3	54
3.3	Uma vizinhança quadrada $\mathcal{N}_{w_i^*}$ em torno do nó i^* que corresponde ao neurônio vencedor w_i^* . A vizinhança do algoritmo de Kohonen é definida em um arranjo topológico (neste caso, uma grade bidimensional) de nós.	54
3.4	Função taxa de aprendizagem $\eta(n)$ no algoritmo SOA.	56
3.5	Função raio de vizinhança $r_c(n)$ no algoritmo SOA.	56
3.6	Definição de vizinhança no algoritmo de Kohonen e no algoritmo SOA.	58
3.7	Definição de vizinhança no algoritmo de SSN e no algoritmo SOA. A vizinhança (hiper-esfera) centrada no vetor de treino é denotada por $\mathcal{N}_{\mathbf{x}}$, enquanto que a hiper-esfera centrada no neurônio vencedor é denotada por \mathcal{N}_c	59
4.1	Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 2$	70
4.2	Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 4$	71
4.3	Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 8$	71
4.4	Diagrama de blocos do sistema de codificação de voz baseado em <i>wavelets</i> e quantização vetorial.	74
4.5	Imagens originais 256×256 utilizadas nas simulações.	79

4.6	Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Mandrill (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$	85
4.7	Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Gull (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$	86
4.8	Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Frog (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$	86
4.9	Imagens originais 256×256	93
4.10	Imagens reconstruídas após a exclusão das sub-bandas S_{11} , S_{12} e S_{13}	94
4.11	Diagrama de blocos do sistema de codificação de imagem considerado	97
4.12	Um exemplo de estratégia de alocação de bits para as sub-bandas de uma decomposição <i>wavelet</i> (em 3 níveis) de uma imagem. Para QV das sub-bandas S_{ij} , com $1 \leq i \leq 3$ e $1 \leq j \leq 3$, os coeficientes <i>wavelet</i> são lidos em blocos quadrados (2×2 ou 4×4) de dimensão K . A sub-banda S_{30} é submetida a QE com 8,0 bpp.	99
4.13	Imagem Airplane original, 256×256 pixels, 8,0 bpp.	100
4.14	Imagem Lena codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	101
4.15	Imagem Goldhill codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	102
4.16	Imagem Peppers codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	103
4.17	Imagem Lena codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	106
4.18	Imagem Goldhill codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	107
4.19	Imagem Peppers codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	108
4.20	Conjunto de treino, correspondente à palavra “aplausos” (0,89 s, 3560 vetores). As coordenadas x_1 and x_2 representam a primeira e a segunda componentes dos vetores de treino \mathbf{x} , respectivamente.	113

4.21	Dicionário obtido com o algoritmo SOA: $K = 2$ e $N = 16$. As coordenadas w_1 and w_2 representam, respectivamente, a primeira e a segunda componentes dos vetores-código w_i , $i = 1, 2, \dots, 16$	114
4.22	Distribuição de Gauss-Markov (coeficiente de correlação $a = 0,9$): (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$	115
4.23	Distribuição gaussiana: (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$	116
4.24	Distribuição uniforme: (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$	116
4.25	Padrões distribuídos sob a forma de uma "borboleta": (a) Conjunto de treino, (b) Dicionário SSC, $K = 2$ e $N = 32$	117
4.26	Padrões distribuídos sob a forma de uma elipse: (a) Conjunto de treino, (b) Dicionário SSC, $K = 2$ e $N = 32$	117
4.27	Número de vezes em que cada vetor-código de um dicionário SSC com $K = 4$ e $N = 64$ é selecionado como vencedor.	124
4.28	Número de vezes em que cada vetor-código de um dicionário FS-SSC com $K = 4$ e $N = 64$ é selecionado como vencedor.	125
4.29	Desempenho dos algoritmos SSC, FS-SSC e LBG em QV de forma de onda de voz: SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 2$	126
4.30	Desempenho dos algoritmos SSC, FS-SSC e LBG em QV de forma de onda de voz: SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 4$	127
5.1	Gráfico de $A(K, N) = \frac{[(K-1)(1+N)+3]}{(K-1)N+K}$	136
6.1	Histograma de um sinal de voz.	148
6.2	Sinal de voz consistindo de 10 frases foneticamente balanceadas (18,76s, 75040 vetores). As coordenadas x_1 e x_2 representam a primeira e a segunda componentes dos vetores de treino $x \in R^2$, respectivamente.	149
6.3	Dicionário obtido com o algoritmo PCA: $K = 2$ e $N = 16$. As coordenadas w_1 e w_2 representam, respectivamente, a primeira e a segunda componentes dos vetores-código $w_i \in R^2$, $i = 1, 2, \dots, 16$	149

6.4	Desempenho dos algoritmos PCA e PCA-ORI em termos de SNRseg e SNRtot do sinal reconstruído para diversas taxas em QV com $K = 2$.	150
6.5	Desempenho dos algoritmos PCA e PCA-ORI em termos de SNRseg e SNRtot do sinal reconstruído para diversas taxas em QV com $K = 4$.	151
6.6	Desempenho dos algoritmos PCA, SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com $K = 2$.	153
6.7	Desempenho dos algoritmos PCA, SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com $K = 4$.	153
6.8	Dicionário projetado com o algoritmo PCA: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.	158
6.9	Dicionário projetado com o algoritmo LBG: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.	159
6.10	Dicionário projetado com o algoritmo PCA-ORI: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.	159
6.11	Organização de um dicionário PCA com $N = N_{POS} + N_{ZERO} + N_{NEG}$ vetores-código.	162
6.12	Exemplo em que os algoritmos 1/2PCA e BT levam a representações diferentes para um dado vetor de entrada \mathbf{x} .	169
6.13	Relação sinal-ruído segmental do sinal de voz reconstruído, em função da probabilidade de erro de bit, para os dicionários LBG e PCA com $K = 2$ e $N = 64$.	176

Lista de Tabelas

2.1	Notação utilizada.	20
2.2	Escala para o teste MOS.	32
4.1	Sensibilidade do algoritmo SOA a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.	67
4.2	Sensibilidade do algoritmo SSC a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.	68
4.3	Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.	69
4.4	Importância das componentes de aproximação e de detalhe (de uma decomposição <i>wavelet</i> multiresolucional de 3 níveis) na qualidade do sinal de voz reconstruído. <i>Wavelet</i> utilizada: Daubechies 6 (db6).	73
4.5	Um exemplo de alocação de taxa para as componentes $a_3(n)$, $d_3(n)$, $d_2(n)$ e $d_1(n)$. Os valores de N e K do quantizador vetorial definem a taxa para cada componente.	75
4.6	Desempenho, fora do conjunto de treino, para diversas estratégias de alocação de bits – valores de SNRseg e SNRtot do sinal reconstruído para diversas taxas de codificação total, R_T (em bit/amostra), com uso de dicionários LBG. Foi realizada DWT com 3 níveis de decomposição. Família <i>wavelet</i> utilizada: Daubechies 6 (db6).	81

4.7	Desempenho, fora do conjunto de treino, em termos de SNRseg e SNRtot do sinal de voz reconstruído para diversas taxas de codificação total R_T (em bit/amostra). Algoritmo utilizado para projeto de dicionários destinados à quantização vetorial simples: LBG.	82
4.8	Desempenho fora do conjunto de treino – resultados, em termos de percentagens, de testes de preferência realizados com 20 avaliadores. São consideradas várias taxas de codificação R . Algoritmo utilizado para projeto de dicionários: LBG.	82
4.9	Desempenho, fora do conjunto de treino, para diversas estratégias de alocação de bits – valores de SNRseg e SNRtot do sinal reconstruído para diversas taxas de codificação total, R_T (em bit/amostra), com uso de dicionários SSC. Foi realizada DWT com 3 níveis de decomposição. Família <i>wavelet</i> utilizada: Daubechies 6 (db6).	83
4.10	Sensibilidade do algoritmo SOA a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.	84
4.11	Sensibilidade do algoritmo SSC a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.	84
4.12	Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.	84
4.13	Complexidade do algoritmo PDS em termos de número de multiplicações por amostra para a imagem Lena. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.	89
4.14	Complexidade do algoritmo PDS-O em termos de número de multiplicações por amostra para a imagem Lena. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.	89

4.15	Complexidade do algoritmo PDS em termos de número de multiplicações por amostra para a imagem Mandrill. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total. . . .	90
4.16	Complexidade do algoritmo PDS-O em termos de número de multiplicações por amostra para a imagem Mandrill. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total. . .	91
4.17	PSNR da imagem Lena reconstruída após a exclusão de determinada(s) sub-banda(s) da decomposição multiresolucional. <i>Wavelet</i> utilizada: Daubechies 6 (db6).	95
4.18	Importância das direções (sub-bandas) do nível de decomposição (resolução) 1 para a imagem Lena. <i>Wavelet</i> utilizada: Daubechies 6 (db6). .	96
4.19	Um exemplo de alocação de taxa para as sub-bandas S_{ij} , com $1 \leq i \leq 3$ e $1 \leq j \leq 3$. Os valores de N e K do quantizador vetorial definem a taxa para cada sub-banda.	98
4.20	Valores de PSNR (dB) apresentados pelas imagens reconstruídas, à taxa de 0,3125 bpp, para diversas técnicas de codificação. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	105
4.21	Valores de PSNR (dB) apresentados pelas imagens reconstruídas, à taxa de 0,5 bpp, para diversas técnicas de codificação. Em QV simples e DWT + QV, foram utilizados dicionários LBG.	109
4.22	Desempenho dos dicionários LBG e SSC em codificação de imagens usando DWT + QV. Valores de PSNR de algumas imagens reconstruídas à taxa de 0,3125 bpp.	109
4.23	Desempenho dos dicionários LBG ao ser reconstruído o conjunto de treino por meio de SPLIT VQ (4,6).	111
4.24	Desempenho dos dicionários SSC ao ser reconstruído o conjunto de treino por meio de SPLIT VQ (4,6).	111
4.25	Desempenho dos dicionários LBG ao ser reconstruído o conjunto de avaliação por meio de SPLIT VQ (4,6).	112
4.26	Desempenho dos dicionários SSC ao ser reconstruído o conjunto de avaliação por meio de SPLIT VQ (4,6).	112

4.27	Taxas médias de identificação para o SRL baseado em QV paramétrica. Avaliação realizada (em um conjunto de 20 locutores cadastrados) sem uso de limiar de rejeição.	123
4.28	CV para as taxas médias de identificação para o SRL baseado em QV paramétrica. Avaliação realizada (em um conjunto de 20 locutores cadastrados) sem uso de limiar de rejeição.	124
4.29	Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) fixada a dimensão $K = 2$ para os dicionários SSC e FS-SSC.	126
5.1	Número de operações requeridas pelo algoritmo LBG.	132
5.2	Número de operações requeridas pelo algoritmo SSC.	133
5.3	Número total de operações requeridas pelos algoritmos LBG e SSC.	134
5.4	Número de iterações realizadas pelos algoritmos LBG e SSC para alguns valores de K e N (com a correspondente taxa de codificação R indicada). São consideradas três fontes diferentes.	137
5.5	Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de número de iterações (n_{LBG}).	138
5.6	Expressões utilizadas para a avaliação comparativa de complexidade dos algoritmos LBG e SSC para projeto de dicionários aplicados a QV de forma de onda de voz, utilizando dimensão $K = 2$	141
5.7	Expressões utilizadas para a avaliação comparativa de complexidade dos algoritmos LBG e SSC para projeto de dicionários aplicados a QV de forma de onda de voz, utilizando dimensão $K = 4$	141
5.8	Número de operações requeridas pelo algoritmo LBG ao serem projetados dicionários destinados à codificação de voz baseada em QV.	142
5.9	Número de operações requeridas pelo algoritmo SSC ao serem projetados dicionários destinados à codificação de voz baseada em QV.	142
6.1	SNRseg e SNRtot dos sinais reconstruídos utilizando-se dicionários PCA, para diversos valores de K e N	155
6.2	SNRseg e SNRtot dos sinais reconstruídos utilizando-se dicionários LBG, para diversos valores de K e N	156

6.3	Sensibilidade do algoritmo LBG a quatro dicionários iniciais diferentes (D_I , D_{II} , D_{III} e D_{PCA}) em termos de número total de iterações para diversos valores de tamanho do dicionário (N) e dimensão (K). O subscrito PCA denota a inicialização do algoritmo LBG usando dicionários PCA.	157
6.4	Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) e dimensão (K) para os dicionários PCA.	157
6.5	Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) e dimensão (K) para os dicionários SOA.	158
6.6	Dicionário PCA com 16 vetores-código bidimensionais $w_i = [w_{i1} \ w_{i2}]^T$, com $1 \leq i \leq 16$. A palavra-binária do i -ésimo vetor-código, w_i , é denotada por b_i , enquanto que w_{ij} representa a j -ésima componente do vetor w_i	163
6.7	Vetores-código bidimensionais efetivamente armazenados na memória de referência do codificador.	165
6.8	Número de operações requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, em função de K e N	166
6.9	Número de adições requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de N fixada a dimensão $K = 4$. Entre parênteses está indicada a economia do algoritmo 1/2PCA com relação ao algoritmo BT.	167
6.10	Número de comparações requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de N . Entre parênteses está indicada a economia do algoritmo 1/2PCA com relação ao algoritmo BT.	167
6.11	SNRseg (dB) do sinal de voz reconstruído, obtidas ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de K e N	168
6.12	Dicionário LBG com 16 vetores-código bidimensionais $w_i = [w_{i1} \ w_{i2}]^T$, com $1 \leq i \leq 16$. A palavra-binária do i -ésimo vetor-código, w_i , é denotada por b_i , enquanto que w_{ij} representa a j -ésima componente do vetor w_i	171
6.13	Índice de desordem para os dicionários LBG e PCA, para diversos valores de K e N	173

6.14	SNRseg (dB) em função da probabilidade de erro de bit, ϵ , para os dicionários LBG e PCA com dimensão $K = 2$ e tamanho N	174
6.15	Redução percentual de SNRseg em relação à transmissão sem erro, para alguns valores de probabilidade de erro de bit, ϵ , para dicionários LBG e PCA com dimensão $K = 2$ e tamanho N	175
6.16	Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 16$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.	177
6.17	Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 32$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.	177
6.18	Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 64$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.	178
6.19	Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 128$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.	178

Conteúdo

1	Introdução	1
1.1	Organização do Trabalho	3
2	Quantização Vetorial	6
2.1	Visão Geral	7
2.2	A Complexidade Computacional da Quantização Vetorial	11
2.2.1	Algoritmo PDS	13
2.2.2	Algoritmo PDS-O	13
2.3	Quantização Vetorial no Domínio da Transformada <i>Wavelet</i> Discreta	15
2.3.1	Introdução	15
2.3.2	Transformada <i>Wavelet</i> Discreta	16
2.3.3	Transformada <i>Wavelet</i> Discreta Aplicada a Imagens	17
2.3.4	Codificação de Imagens Usando QV dos Coeficientes <i>Wavelet</i>	24
2.4	Quantização Vetorial em Canais Ruidosos	27
2.5	Avaliação de Desempenho de Quantizadores Vetoriais Aplicados à Com- pressão de Sinais	30
2.5.1	Escore Médio de Opinião	32
2.5.2	Testes de Preferência	32
2.5.3	Relação Sinal-Ruído (SNR)	33
2.5.4	Relação Sinal-Ruído Segmental (SNRseg)	34
2.5.5	Distorção Espectral	35
2.5.6	Relação Sinal-Ruído de Pico (PSNR)	35
2.6	Quantização Vetorial Aplicada ao Reconhecimento de Locutor	36
2.6.1	Preliminares	36
2.6.2	Técnicas para Reconhecimento de Locutor	37
2.6.3	Descrição do Sistema de Reconhecimento de Locutor	38

2.6.4	Análise por Predição Linear	42
3	Algoritmos Não-Supervisionados Aplicados à Quantização Vetorial	48
3.1	Introdução	48
3.2	Algoritmo de Kohonen	51
3.3	Algoritmo SOA	52
3.3.1	Descrição do Algoritmo	53
3.3.2	O Paradigma de Vizinhança	57
3.3.3	Considerações Adicionais	59
3.4	Algoritmo SSC	60
3.5	Algoritmo FS-SSC	63
3.6	Considerações Finais	64
4	Avaliação dos Dicionários SOA, SSC e FS-SSC	65
4.1	Quantização Vetorial de Forma de Onda de Voz – Domínio Temporal	65
4.1.1	Sensibilidade ao Dicionário Inicial	66
4.1.2	SNRseg versus Taxa de Codificação	70
4.2	Quantização Vetorial de Forma de Onda de Voz – Domínio <i>Wavelet</i>	72
4.2.1	A Importância das Componentes de Aproximação e de Detalhe	72
4.2.2	Resultados de Codificação	74
4.3	Quantização Vetorial de Imagem – Domínio Espacial	78
4.3.1	Sensibilidade ao Dicionário Inicial	78
4.3.2	PSNR versus Taxa de Codificação	85
4.3.3	Complexidade Computacional da Etapa de Codificação da QV	88
4.4	Quantização Vetorial de Imagem – Domínio <i>Wavelet</i>	91
4.4.1	A Importância do Nível de Decomposição 1	92
4.4.2	A Importância das Sub-Bandas do Nível de Decomposição 1	92
4.4.3	Resultados de Codificação	96
4.5	Quantização Vetorial dos Parâmetros LSF	110
4.6	Capacidade de Obtenção de Padrões Representativos do Conjunto de Treino	113
4.6.1	Desempenho do Algoritmo SOA em Sinais de Voz	113
4.6.2	Desempenho do Algoritmo SOA em Sinais com Distribuição Conhecida	114
4.6.3	Desempenho do Algoritmo SSC	115

4.7	Quantização Vetorial Aplicada ao Reconhecimento de Locutor	118
4.7.1	Resultados	119
4.8	Algoritmo FS-SSC Aplicado ao Projeto de Dicionários para QV de Voz	122
5	Complexidade Computacional do Algoritmo SSC	128
5.1	Avaliação de Complexidade	128
5.1.1	Algoritmo LBG	129
5.1.2	Algoritmo SSC	130
5.1.3	LBG versus SSC	131
5.2	Condições para as quais SSC é mais Eficiente que LBG em Termos de Multiplicações, Subtrações, Adições e Comparações	132
5.2.1	Número de Multiplicações	132
5.2.2	Número de Subtrações	133
5.2.3	Número de Adições	134
5.2.4	Número de Comparações	135
5.2.5	Considerações Gerais	135
5.3	Resultados	137
5.4	Conclusões	139
6	Análise de Componentes Principais Aplicada à Quantização Vetorial	143
6.1	Introdução	143
6.2	Projeto de Dicionários Usando Análise de Componentes Principais . . .	146
6.2.1	Resultados	147
6.3	Um Método para Redução da Complexidade Computacional da Etapa de Codificação da QV	160
6.3.1	Algoritmo para Redução da Complexidade Computacional da Etapa de Codificação da QV	162
6.3.2	Resultados	166
6.4	Avaliação dos Dicionários PCA para Uso em Transmissão de Voz por Canal Ruidoso	170
6.4.1	Robustez Inerente aos Erros de Canal	170
6.4.2	Resultados	170
7	Conclusões	179
7.1	Introdução	179

7.2	Comentários sobre as Avaliações Realizadas	180
7.3	Contribuições Mais Relevantes	182
7.4	Propostas de Trabalhos Futuros	183
A	Entropia Normalizada dos Vetores-Código	185

Capítulo 1

Introdução

A compressão de sinais, cujo objetivo fundamental é reduzir o número de bits necessários para representar adequadamente os sinais (voz, imagem, áudio, vídeo), desempenha um papel importante em aplicações que necessitam minimização dos requisitos de largura de faixa e/ou de capacidade de armazenamento, tais como: sistemas multimídia, redes digitais de serviços integrados, videoconferência, sistemas de resposta vocal, telefonia móvel, sistemas de armazenamento de imagens médicas e de impressões digitais e transmissão de imagens de sensoriamento remoto obtidas por satélites. Nesse cenário, a quantização vetorial (QV) apresenta-se como uma poderosa técnica, bastante utilizada em diversos sistemas de codificação de sinais.

Em sistemas de reconhecimento de locutor, um dos méritos da quantização vetorial reside no fato de que essa técnica dispensa a necessidade de alinhamento temporal, uma vez que permite ao sistema de reconhecimento flexibilidade quanto ao tamanho das sentenças proferidas pelos locutores.

O desempenho dos sistemas de compressão de sinais que utilizam QV e dos sistemas de reconhecimento de locutor baseados em QV depende da qualidade dos dicionários projetados. Dentre as diversas técnicas utilizadas para projeto de dicionários, o algoritmo LBG (Linde-Buzo-Gray) destaca-se por sua ampla utilização.

Neste trabalho são apresentadas técnicas para projeto de dicionários aplicados a QV de sinais de voz e imagens. A primeira, denominada SOA (*self-organizing algorithm*) e inspirada no algoritmo de Kohonen, corresponde a um algoritmo de aprendizagem não-supervisionada que utiliza um paradigma de vizinhança diferente do proposto por Kohonen para atualização dos vetores-código: no algoritmo SOA a vizinhança é definida no próprio espaço de padrões, enquanto que no algoritmo de Kohonen a vizinhança

é definida em um arranjo topológico de nós (em geral, um mapa bidimensional). A segunda técnica, denominada SSC (*synaptic space competitive*), corresponde a um algoritmo que utiliza aprendizagem competitiva simples. A terceira, denominada FS-SSC (*frequency sensitive SSC*), é resultante da introdução do princípio de consciência de Grossberg no algoritmo SSC. A quarta técnica, denotada por PCA (como alusão a *principal component analysis*), destaca-se pela simplicidade e rapidez – os vetores-código são determinados de acordo com os autovalores e autovetores (componentes principais) da matriz de covariância de um sinal de voz, diferindo consideravelmente de abordagens tradicionais, tais como os algoritmos LBG e de Kohonen, bem como dos algoritmos SOA, SSC e FS-SSC, que necessitam de um conjunto de treino para atualizar iterativamente os vetores-código.

São apresentados resultados de simulações envolvendo codificação de forma de onda de voz e codificação de imagens, utilizando QV simples (isto é, quantização vetorial convencional) bem como DWT + QV (isto é, quantização vetorial dos coeficientes presentes nas sub-bandas resultantes de uma decomposição *wavelet* multiresolucional). São também apresentados resultados concernentes à aplicação de QV para codificação dos parâmetros LSF (*line spectral frequencies*) de voz e resultados obtidos ao ser considerado um sistema de reconhecimento de locutor baseado em QV paramétrica.

O trabalho contempla uma avaliação comparativa de desempenho dos algoritmos SOA, SSC, FS-SSC e LBG tendo em vista diversos aspectos: a qualidade dos dicionários projetados com respeito à qualidade dos sinais reconstruídos, em sistemas de codificação de sinais; a qualidade dos dicionários de padrões acústicos (utilizados para representar a identidade vocal dos locutores) projetados com relação às taxas médias de identificação obtidas no sistema de reconhecimento de locutores considerado; a influência do dicionário inicial no desempenho dos algoritmos de projeto de dicionários.

É investigada a complexidade computacional dos algoritmos SSC e LBG. São obtidas expressões analíticas (em função do tamanho do dicionário, da dimensão dos seus vetores-código, do número de vetores do conjunto de treino e do número de iterações desses algoritmos) que estabelecem as condições que devem ser obedecidas para que o algoritmo SSC seja mais eficiente que o algoritmo LBG quanto ao número de operações (divisões, multiplicações, comparações, adições e subtrações) realizadas durante o projeto do dicionário.

Diversos aspectos do algoritmo PCA são apresentados, dentre os quais destacam-se sua capacidade de gerar dicionários organizados e sua adequação para quantização

vetorial com dimensão 2, com a qual são obtidos sinais reconstruídos com excelente qualidade. O algoritmo PCA, além de ser um método simples e intuitivo para projeto de dicionários destinados à quantização vetorial de forma de onda de voz, não apresenta o problema de ajuste de parâmetros (característico de vários algoritmos de redes neurais) e não requer a escolha de um dicionário inicial, como ocorre nos algoritmos LBG, SOA, SSC, FS-SSC e de Kohonen.

O presente trabalho também apresenta um método para redução da complexidade computacional da etapa de codificação da QV. O método proposto utiliza a organização estruturada dos dicionários PCA para reduzir significativamente o número de operações realizadas na etapa de codificação da QV (etapa de determinação do vizinho mais próximo de cada vetor a ser codificado) bem como para minimizar os requisitos de memória exigida para armazenar o dicionário. Além disso, mostra-se que os dicionários PCA apresentam uma robustez inerente aos erros de canal.

As avaliações realizadas neste trabalho e os diversos resultados obtidos mostram que os algoritmos SOA, SSC, FS-SSC e PCA apresentam-se como alternativas para o tradicional algoritmo LBG.

1.1 Organização do Trabalho

Este trabalho é organizado em 7 capítulos e 1 apêndice.

No Capítulo 2 é apresentada uma visão geral da quantização vetorial, com a devida formulação matemática desta técnica. Por se tratar do método mais amplamente utilizado para projeto de quantizadores vetoriais, o algoritmo LBG é descrito. A questão da complexidade computacional na fase de codificação da QV é abordada, sendo enfocados dois algoritmos clássicos para otimização do processo de procura do vizinho mais próximo. São apresentados alguns fundamentos da transformada *wavelet* discreta (DWT, *discrete wavelet transform*), com ênfase à decomposição *wavelet* multiresolucional, sendo apontadas algumas características que justificam sua ampla utilização em sistemas de codificação de imagens e sendo brevemente discutidos alguns fatores que afetam o desempenho de sistemas de codificação que utilizam DWT; nesse contexto, apresenta-se uma abordagem sucinta da codificação de imagens baseada em QV dos coeficientes *wavelet*. O Capítulo 2 também aborda o problema da QV em canais ruidosos, destacando a técnica de quantização vetorial robusta como alternativa para minimizar o impacto dos canais ruidosos no desempenho dos sistemas de comuni-

cação baseados em QV. Em se tratando da avaliação de desempenho de quantizadores vetoriais aplicados à compressão de voz e imagem, são abordadas duas metodologias de avaliação subjetiva da qualidade dos sinais reconstruídos – o escore médio de opinião (MOS, *mean opinion score*) e os testes de preferência, além de quatro medidas de avaliação objetiva da qualidade dos sinais reconstruídos – a relação sinal-ruído, a relação sinal-ruído segmental, a distorção espectral e a relação sinal-ruído de pico. O capítulo é encerrado com a descrição do sistema de reconhecimento (identificação) de locutor baseado em QV paramétrica considerado no presente trabalho, sendo descritos os parâmetros acústicos utilizados pelo sistema: os coeficientes LPC, cepestrais, cepestrais ponderados, delta cepestrais e delta cepestrais ponderados.

O Capítulo 3 fornece algumas noções básicas de redes neurais, com destaque para a visão geral das regras de aprendizagem supervisionada e não-supervisionada. São descritos o algoritmo de Kohonen e os algoritmos SOA, SSC e FS-SSC. São apontadas diversas diferenças entre os algoritmos SOA e de Kohonen, sendo discutidos alguns aspectos que justificam a adequação do algoritmo SOA para projeto de dicionários. É apresentada uma abordagem elucidativa do paradigma de vizinhança introduzido pelo algoritmo SOA.

O Capítulo 4 apresenta diversos resultados de avaliação dos algoritmos SOA, SSC, e FS-SSC, sendo realizada uma comparação de desempenho desses algoritmos com o algoritmo LBG. Tanto em codificação de sinais utilizando QV como em identificação de locutor baseada em QV paramétrica, fica evidenciado o bom desempenho dos algoritmos SOA, SSC e FS-SSC.

No Capítulo 5 é apresentada uma avaliação da complexidade computacional dos algoritmos SSC e LBG. São obtidas expressões analíticas para o número de operações executadas por esses algoritmos. Essas expressões estabelecem as condições que devem ser obedecidas para que o algoritmo SSC seja mais eficiente que o algoritmo LBG em termos de número de multiplicações, subtrações, adições e comparações (ressalte-se que o algoritmo SSC realiza apenas uma operação de divisão, sendo mais eficiente que o algoritmo LBG quanto a essa operação). Os resultados apresentados ao final do capítulo, concernentes ao projeto de dicionários para codificação de forma de onda de voz, respaldam as formulações analíticas desenvolvidas no presente trabalho para a avaliação de complexidade.

O Capítulo 6, que trata da análise de componentes principais aplicada ao projeto de dicionários destinados à codificação de forma de onda de voz, procura fornecer o

instrumental matemático necessário para a perfeita compreensão da técnica de projeto proposta. Assim, especial atenção é dada à transformada KLT (*Karhunen-Loève transform*), sendo enfocados os autovetores e os autovalores (componentes principais) correspondentes. O capítulo apresenta uma descrição do algoritmo de projeto proposto, denotado por PCA, sendo realizada uma avaliação comparativa de desempenho desse método com os algoritmos SOA, SSC e LBG. São destacadas a simplicidade, a rapidez e a adequação do algoritmo PCA. No Capítulo 6 a organização estruturada dos dicionários PCA é explorada eficientemente em um novo método de codificação, que leva a uma redução significativa da complexidade computacional do processo de determinação do vizinho mais próximo de cada vetor da fonte a ser codificada. O método de codificação proposto, denominado 1/2PCA em virtude de o codificador utilizar apenas metade do dicionário previamente projetado com o algoritmo PCA, é discutido e avaliado. O Capítulo 6 é finalizado com uma avaliação dos dicionários PCA para uso em transmissão de voz por canal ruidoso. A avaliação dos dicionários PCA quanto à robustez aos erros de canal é levada a efeito utilizando o modelo de canal binário simétrico. Os resultados apresentados mostram que os dicionários PCA apresentam uma menor sensibilidade (um maior robustez) aos erros de canal quando comparados aos dicionários LBG.

No Capítulo 7 são apresentadas as conclusões do trabalho, sendo destacadas as principais contribuições. Por fim são apresentadas propostas de continuação da pesquisa.

Finalmente, o Apêndice A descreve a entropia normalizada dos vetores-código.

Capítulo 2

Quantização Vetorial

O principal objetivo das técnicas de compressão de sinais é reduzir o número de bits necessários para se representar adequadamente os sinais, visando minimizar os requisitos de largura de faixa (em aplicações envolvendo transmissão de sinais) e de memória (em aplicações envolvendo armazenamento de sinais) [1]. Apesar de alguns sistemas não apresentarem grandes limitações de largura de faixa, como é o caso das redes de comunicações por fibra ótica, e embora a evolução tecnológica esteja continuamente contribuindo para o surgimento de memórias com grandes capacidades de armazenamento, a compressão de sinais desempenha um papel importante, devido a uma série de fatores, tais como [2, 3]:

- a grande utilização dos sistemas multimídia tem levado ao aumento da demanda no tocante ao armazenamento de voz, música, imagens, vídeo e dados em forma comprimida;
- um maior número de canais de comunicações pode ser multiplexado em sistemas de faixa larga, por meio do uso de técnicas de compressão para reduzir os requisitos de largura de faixa de cada sinal a ser multiplexado;
- nos sistemas de reconhecimento de fala e de locutor e nos sistemas de resposta vocal, vocabulários maiores podem ser armazenados por meio da redução dos requisitos de memória necessários para cada padrão de voz;
- nas redes digitais de serviços integrados (ISDN, *integrated services digital networks*), as técnicas de compressão permitem uma integração eficiente de sinais e dados;

- em telefonia móvel celular, a largura de faixa é severamente limitada, o que tem motivado muitos estudos em compressão de voz.

Nesse cenário a quantização vetorial [3,4] tem sido utilizada com sucesso, permitindo obter elevadas taxas de compressão em diversos sistemas de codificação de voz, imagem e vídeo.

A quantização vetorial também pode ser utilizada em outras aplicações, como por exemplo [5,6]: realce de imagens, reconhecimento automático de alvos (ao ser utilizado radar), verificação de assinaturas, reconhecimento de fala, reconhecimento de locutor e supressão de ruído em sinais de voz.

2.1 Visão Geral

A quantização vetorial, que pode ser vista como uma extensão da quantização escalar em um espaço multidimensional, encontra-se fundamentada na Teoria da Distorção Versus Taxa [7], formulada por Shannon, segundo a qual um melhor desempenho é obtido codificando-se blocos de amostras (isto é, vetores) ao invés de amostras individuais (isto é, escalares). Em outras palavras, essa teoria ressalta a superioridade da quantização vetorial sobre a quantização escalar [8]. Matematicamente, a quantização vetorial pode ser definida como um mapeamento Q de um vetor de entrada x pertencente ao espaço euclidiano K -dimensional, R^K , em um vetor pertencente a um subconjunto finito W de R^K , ou seja,

$$Q : R^K \rightarrow W. \quad (2.1)$$

O dicionário $W = \{w_i; i = 1, 2, \dots, N\}$ é o conjunto de vetores de reprodução (também denominados vetores-código ou vetores de reconstrução), K é a dimensão do quantizador vetorial e N é o tamanho do dicionário, isto é, o número de vetores-código (ou número de níveis, em analogia com a quantização escalar).

O mapeamento Q introduz um particionamento de R^K em N células (denominadas regiões de Voronoi) $S_i, i = 1, 2, \dots, N$, tais que

$$\bigcup_{i=1}^N S_i = R^K \text{ e } S_i \cap S_j = \emptyset \text{ para } i \neq j, \quad (2.2)$$

em que cada célula S_i é definida da seguinte maneira:

$$S_i = \{x : Q(x) = w_i\}. \quad (2.3)$$

O vetor-código w_i constitui o vetor representativo de todos os vetores de entrada pertencentes à célula S_i , conforme ilustra a Figura 2.1. Como a quantização vetorial realiza um mapeamento de padrões de entrada (vetores de entrada x) semelhantes em padrões de saída (vetores-código w_i) semelhantes, ela pode ser vista como uma forma de reconhecimento de padrões, em que um padrão de entrada é “aproximado” por um padrão de referência, pertencente a um conjunto predeterminado (dicionário) de padrões (vetores-código) de referência [3, 9].

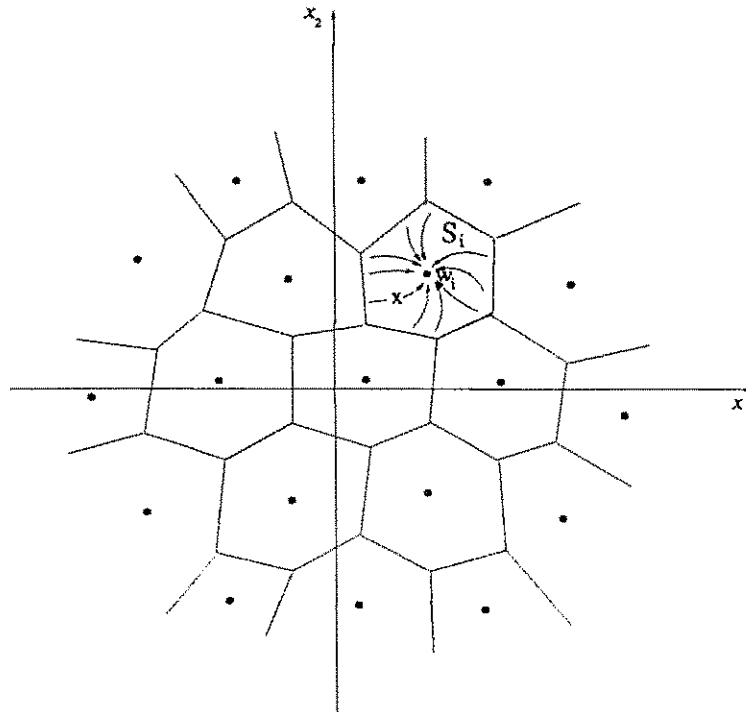


Figura 2.1: Partição do espaço euclidiano bidimensional, R^2 , introduzido pelo mapeamento dos vetores de entrada x nos vetores-código w_i . As coordenadas x_1 e x_2 representam a primeira e a segunda componentes do vetor $x \in R^2$, respectivamente.

Em um sistema de compressão de sinais baseado em quantização vetorial simples, apresentado na Figura 2.2, um quantizador vetorial pode ser visto como a combinação de duas funções: um codificador de fonte e um decodificador de fonte. Dado um vetor $x \in R^K$, da fonte a ser codificada, o codificador calcula a distorção $d(x, w_i)$ entre o vetor de entrada (vetor a ser quantizado) e cada vetor-código w_i , $i = 1, 2, \dots, N$ do dicionário W . A regra ótima para codificação é a regra do vizinho mais próximo [10], na qual uma representação binária do índice I , denotada por b_I , é transmitida ao decodificador de fonte se o vetor-código w_I corresponder à menor distorção [11], isto

é, se w_I for o vetor-código que apresentar a maior similaridade com x dentre todos os vetores-código do dicionário. Em outras palavras, o codificador usa a regra de codificação $C(x) = b_I$ se $d(x, w_I) < d(x, w_i), \forall i \neq I$. Ao receber a representação binária $b_I = (b_1(I), b_2(I), \dots, b_m(I))$ do índice I , o decodificador de fonte, que dispõe de uma cópia do dicionário W , simplesmente procura pelo I -ésimo vetor-código e produz o vetor w_I como a reprodução (versão quantizada) de x . Em outras palavras, é utilizada a seguinte regra de decodificação: $\mathcal{D}(b_I) = w_I$ [12].

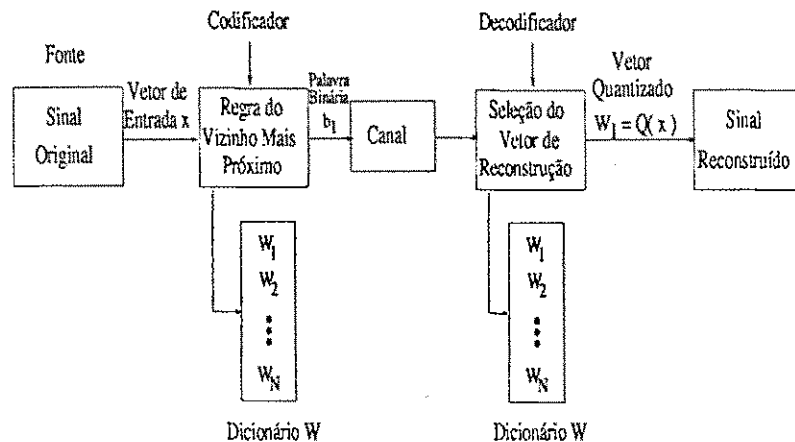


Figura 2.2: Codificação/decodificação em um sistema de codificação baseado em quantização vetorial simples.

A quantização vetorial, portanto, constitui uma técnica de compressão com perdas, visto que o sinal reconstruído é uma versão degradada do sinal original. O erro médio de quantização ao se representar o sinal de entrada por sua versão quantizada é chamado distorção do quantizador. Por outro lado, a taxa de codificação do quantizador vetorial, que mede o número de bits por componente do vetor, é dada por $R = \frac{1}{K} \log_2 N$. Para o caso de serem utilizadas palavras-binárias b_i de comprimento fixo m bits, essa taxa pode ser dada por $R = \frac{m}{K}$ ¹. Em codificação de forma de onda de voz (e. g. [13,14]), R é expressa em bit/amostra. Em se tratando de codificação de imagens (e. g. [11,15,16]), R é expressa em bits por *pixel* (bpp).

¹Ao serem aplicadas técnicas de codificação entrópica, são utilizadas palavras-binárias com comprimentos diferentes, determinados de acordo com considerações estatísticas – o objetivo, nesse caso, é a obtenção de taxas inferiores a $\frac{1}{K} \log_2 N = \frac{m}{K}$. Para tanto, são utilizadas palavras-binárias de comprimento pequeno para os vetores-código mais prováveis de serem os representantes de um dado vetor de entrada e palavras-binárias de comprimento grande para os vetores-código menos prováveis de serem os representantes de um dado vetor de entrada.

Uma das questões principais no projeto de quantizadores vetoriais é o compromisso entre taxa e distorção. O alvo a ser perseguido é a obtenção de um dicionário ótimo, que minimize a distorção média introduzida pela aproximação de cada vetor de entrada por um dos vetores-código. O quantizador ótimo deve satisfazer duas condições necessárias: a condição de vizinho mais próximo para partição e a condição de centróide para os vetores-código [3, 17].

Existem diversas técnicas para o projeto de dicionários para quantização vetorial, dentre as quais podem ser citadas: algoritmo LBG (Linde-Buzo-Gray) [10]; algoritmo auto-organizativo de Kohonen [18–20] e outros algoritmos não-supervisionados de redes neurais [21–26]; *stochastic relaxation* [17]; algoritmos *fuzzy* [27, 28]; algoritmos genéticos [29]; algoritmo PNN (*pair-wise nearest neighbor*) [30].

Por se tratar da técnica mais popular e amplamente utilizada para projeto de quantizadores vetoriais, o algoritmo LBG será descrito a seguir.

Algoritmo LBG

Seja a iteração do algoritmo LBG denotada por n . Dados K , N e um limiar de distorção $\epsilon \geq 0$, o algoritmo LBG [10], também conhecido como GLA (*generalized Lloyd algorithm*) consiste da seguinte seqüência de passos:

- *Passo 1*) inicialização: dado um dicionário inicial W_0 e um conjunto de treino $\mathbf{X} = \{\mathbf{x}_m; m = 1, 2, \dots, M\}$, faça $n = 0$ e $D_{-1} = \infty$;
- *Passo 2*) particionamento: dado W_n (dicionário na n -ésima iteração) aloque cada vetor de treino (vetor de entrada) na respectiva classe (célula de Voronoi) segundo o critério do vetor-código mais próximo; calcule a distorção²

$$D_n = \sum_{i=1}^N \sum_{\mathbf{x}_m \in S_i} d(\mathbf{x}_m, \mathbf{w}_i); \quad (2.4)$$

- *Passo 3*) teste de convergência (critério de parada): se $(D_{n-1} - D_n)/D_n \leq \epsilon$ pare, com W_n representando o dicionário final (dicionário projetado); caso contrário, continue;

²Em algumas abordagens, é calcula a distorção média $\tilde{D}_n = D_n/M$. O cálculo da média, no entanto, requer uma operação de divisão (por M), que pode ser evitada, tendo em vista que qualquer das abordagens (D_n ou \tilde{D}_n) tem o mesmo efeito em termos do teste de convergência (*passo 3* do algoritmo).

- *Passo 4)* atualização do dicionário: compute os novos vetores-código como os centróides das classes de vetores; faça $W_{n+1} = W_n$; faça $n = n + 1$ e vá para *Passo 2*.

Em essência, no algoritmo LBG a função distorção decresce monotonicamente, uma vez que o dicionário é iterativamente atualizado visando satisfazer as condições de centróide e de vizinho mais próximo. Na dinâmica do algoritmo LBG, a distorção introduzida ao se representarem os vetores do conjunto de treinamento pelos correspondentes vetores-código (centróides) é monitorada a cada iteração. A regra de parada (teste de convergência) do algoritmo baseia-se nessa distorção monitorada – o treinamento do dicionário é encerrado quando $(D_{n-1} - D_n)/D_n \leq \epsilon$. Existem, contudo, alguns problemas apresentados pelo algoritmo LBG, comumente relatados [31]: alguns vetores-código podem ser subutilizados e, em casos extremos, até mesmo nunca serem acessados, ou seja, o projeto do dicionário pode resultar em células de Voronoi pequenas ou até mesmo vazias; a velocidade de convergência e o desempenho do dicionário final dependem do dicionário inicial.

A seguir será abordada a complexidade computacional envolvida na fase de codificação da quantização vetorial.

2.2 A Complexidade Computacional da Quantização Vetorial

O problema da determinação do vizinho mais próximo desempenha papel importante em diversas aplicações, como por exemplo reconhecimento de padrões e compressão de sinais envolvendo quantização vetorial. Em virtude da necessidade de comparação direta de um padrão de teste com cada padrão de referência armazenado, a regra do vizinho mais próximo implica elevada complexidade computacional, o que tem despertado o interesse de diversas pesquisas relacionadas à análise e à concepção de métodos de procura eficientes.

Em quantização vetorial, em particular, a complexidade computacional existente na fase de codificação dos vetores a serem quantizados constitui um problema crítico.

A etapa de representação de um vetor de entrada por um dos vetores-código, ou seja, a etapa de determinação do vizinho mais próximo, representa uma das principais limitações da quantização vetorial, isto é: a complexidade computacional na fase de

codificação da QV cresce exponencialmente com a dimensão para uma dada taxa, conforme será demonstrado a seguir.

A relação entre o número de vetores-código N , a dimensão K e a taxa R pode ser expressa como $N = 2^{KR}$. Para codificar um dado vetor de entrada, é necessário encontrar sua distância para cada um dos N vetores-código e depois comparar as distâncias de modo a encontrar o vetor-código mais semelhante, ou seja, o vizinho mais próximo. No algoritmo convencional de busca total, a codificação de cada vetor de entrada requer, portanto, 2^{KR} cálculos de distância (distorção) e $2^{KR} - 1$ comparações. No caso da medida de distorção de erro quadrático, isto é,

$$d(\mathbf{x}, \mathbf{w}_i) = \sum_{j=1}^K (x_j - w_{ij})^2, \quad (2.5)$$

em que w_{ij} é a j -ésima componente do vetor-código \mathbf{w}_i e x_j é a j -ésima componente do vetor de entrada \mathbf{x} , cada cálculo de distância requer K multiplicações, K subtrações e $K - 1$ adições. Portanto, para codificar cada vetor de entrada, $K2^{KR}$ multiplicações, $K2^{KR}$ subtrações, $(K - 1)2^{KR}$ adições e $2^{KR} - 1$ comparações devem ser realizadas. A complexidade de um quantizador vetorial pode ser alternativamente expressa em termos de 2^{KR} multiplicações, 2^{KR} subtrações, $(1 - \frac{1}{K})2^{KR}$ adições e $(2^{KR} - 1)/K$ comparações por amostra.

Em resumo, a complexidade computacional de um quantizador vetorial de dimensão K e taxa R requer um número de operações por amostra da ordem de $N = 2^{KR}$ para cada vetor de entrada se uma busca completa (busca total, busca exaustiva) é realizada ao longo do dicionário.

As técnicas destinadas à aceleração do processo de codificação da QV podem ser classificadas em dois grupos [32]. O primeiro grupo consiste de métodos que não se ocupam com o problema de determinação do vizinho mais próximo propriamente dito – consiste de métodos que buscam uma solução sub-ótima (mas que geralmente é “muito boa”) em termos de erro médio quadrático introduzido ao se representar o sinal por sua versão quantizada. Um desses métodos consiste no uso de quantização vetorial estruturada em árvore (TSVQ, *tree-structured vector quantization*) [3], na qual a busca pelo vizinho mais próximo é realizada em etapas. Dentre os algoritmos pertencentes ao primeiro grupo, pode ser citado o descrito em [33]. O segundo grupo de técnicas procura uma solução exata para o problema de codificação por vizinho mais próximo, que apresente uma menor complexidade computacional comparada à busca exaustiva

do vizinho mais próximo. Dentre os algoritmos pertencentes ao segundo grupo, podem ser citados aqueles apresentados em [34–40]. A seguir serão descritos os algoritmos propostos em [34] e [35], em virtude de serem utilizados em algumas simulações realizadas no presente trabalho.

2.2.1 Algoritmo PDS

O algoritmo de busca por distância parcial, denotado ao longo deste trabalho por PDS (*partial distance search*), proposto em [34], constitui uma das técnicas tradicionais para redução da complexidade computacional envolvida na fase de procura do vizinho mais próximo (fase de codificação). Nesse algoritmo, o codificador toma a decisão antes de completar o cálculo de distância entre o vetor de entrada (vetor a ser codificado) e um vetor-código.

O codificador decide que o vetor-código em questão não representa o vizinho mais próximo se, para algum $j < K$, a distância acumulada (ou seja, a distância parcial) para as primeiras j amostras do vetor de entrada é maior que a menor distância até então encontrada no processo de procura. O codificador, então, interrompe o cálculo de distância para esse vetor-código e inicia esse cálculo para o próximo vetor-código do dicionário. Com essa abordagem, o número de multiplicações por amostra é reduzido drasticamente. Observa-se, também, redução do número de subtrações/adições por amostra. Muito embora o algoritmo PDS introduza um aumento do número de comparações, a complexidade global observada na fase de procura do vizinho mais próximo é reduzida.

A estrutura básica do algoritmo PDS é apresentada na Figura 2.3, em que d denota a distância (distorção), d_{min} representa a distância mínima (ou seja, a menor distância até então encontrada no processo de procura), x_j denota a j -ésima componente do vetor de entrada K -dimensional \mathbf{x} e w_{ij} representa a j -ésima componente do i -ésimo vetor-código K -dimensional \mathbf{w}_i , $i = 1, \dots, N$.

2.2.2 Algoritmo PDS-O

Esse método consiste em ordenar os vetores-código de acordo com o tamanho das células de Voronoi correspondentes, visando aumentar a eficiência do algoritmo PDS. No presente trabalho, essa técnica é denotada por PDS-O, onde O representa o processo de ordenamento dos vetores-código.

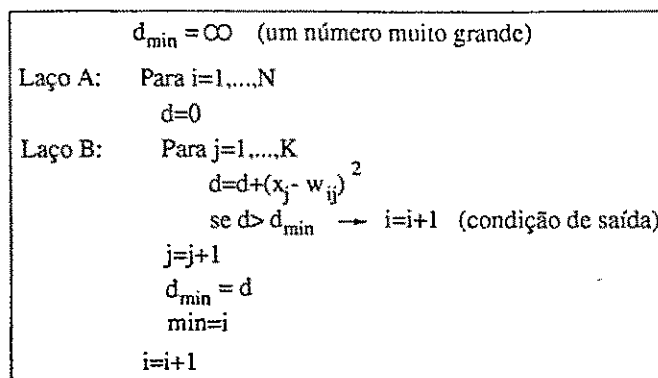


Figura 2.3: Algoritmo PDS.

Observando a estrutura do algoritmo PDS, apresentada na Figura 2.3, verifica-se que, para um dado vetor de entrada \mathbf{x} , a máxima redução de cálculos computacionais é obtida quando a condição de saída é satisfeita o mais cedo possível dentro do Laço B para cada w_i . Em outras palavras, uma saída prematura do Laço B é alcançada com a condição de que o vetor-código que leve ao menor valor possível de d_{\min} para um dado \mathbf{x} seja acessado o mais cedo possível no Laço A.

Conforme [35], como d_{\min} assume o menor valor possível para um dado \mathbf{x} apenas depois que o vizinho (vetor-código) mais próximo de \mathbf{x} tenha sido verificado, é possível obter máxima economia de esforço computacional tendo como primeiro vetor-código o vizinho mais próximo de \mathbf{x} , isto é, $w_1 = Q(\mathbf{x})$. Tal condição não é, obviamente, prática, uma vez que o conhecimento *a priori* de $Q(\mathbf{x})$ elimina a necessidade de procura. Segundo [35], entretanto, uma solução probabilística pode ser dada para diminuir a complexidade computacional, conforme abordado a seguir.

Ordenamento

Seja $W = \{w_i; i = 1, 2, \dots, N\}$ o dicionário de tamanho N , em que $w_i = (w_{ij}, j = 1, \dots, K)$ é um vetor K -dimensional. Seja p_i a probabilidade de que um dado vetor de entrada \mathbf{x} pertença à região de Voronoi S_i , correspondente ao vetor-código w_i . Em outras palavras, p_i corresponde à probabilidade de que um vetor-código w_i seja o vetor-código mais próximo (vizinho mais próximo) para um dado vetor \mathbf{x} .

Seja $S = \{x_s, s = 1, 2, \dots, m\}$ um longo conjunto (isto é, $m \gg N$) representativo de vetores usados para o treinamento do quantizador vetorial. Ao final do treinamento, o conjunto S é particionado em N subconjuntos $S_i, i = 1, 2, \dots, N$, em que $S_i =$

$\{x_s : w_i = Q(x_s)\}$, ou seja, w_i é o vetor-código mais próximo de todos $x_s \in S_i$.

Se m_i é o tamanho da região S_i (isto é, m_i é o número de vetores de treino pertencentes à célula de Voronoi S_i), a estimativa para a probabilidade de que w_i seja o vetor-código mais próximo para qualquer x_s pode ser obtida como

$$p_i = \frac{m_i}{m} \text{ para } m \rightarrow \infty. \quad (2.6)$$

Os p_i 's obtidos com o uso de um longo conjunto representativo de vetores de treino servem como uma boa medida das probabilidades necessárias para o ordenamento do dicionário. Se os vetores-código estão arranjados no dicionário de acordo com a ordem decrescente de p_i , ou seja $p_1 > p_2 > \dots > p_N$, então a probabilidade de se obter o menor valor de d_{min} em estágios iniciais da procura pode aumentar, o que ajuda a diminuir a complexidade computacional do algoritmo PDS.

2.3 Quantização Vetorial no Domínio da Transformada *Wavelet* Discreta

2.3.1 Introdução

A técnica de codificação por sub-bandas desempenha um papel importante em diversos sistemas de codificação de voz, imagem, vídeo e áudio.

A codificação por sub-bandas foi originalmente introduzida no contexto de codificação de voz por Crochiere *et al.* [41]. A extensão da filtragem de sub-bandas de 1-D para 2-D foi apresentada por Vetterli [42], sendo originalmente aplicada à codificação de imagens por Woods *et al.* [43,44].

Em codificação por sub-bandas, um banco de filtros de análise é aplicado ao sinal de entrada, gerando um conjunto de sinais com faixa de frequências mais estreita, cada um representando uma determinada sub-banda do espectro do sinal de entrada. As sub-bandas são codificadas, posteriormente recombinadas e, por meio da aplicação de filtros de síntese, o sinal reconstruído é obtido.

O particionamento do espectro de um sinal pode ser levado a efeito por meio de uma decomposição *wavelet*, a qual tem sido amplamente utilizada em vários sistemas de codificação de imagens, permitindo a obtenção de excelentes resultados.

Nas seções a seguir, são apresentados alguns fundamentos da transformada *wavelet* discreta, uma das ferramentas utilizadas no presente trabalho.

2.3.2 Transformada *Wavelet* Discreta

A transformada *wavelet* discreta (DWT) tem sido utilizada com sucesso em diversas aplicações [45–49], dentre as quais podem ser citadas: supressão de ruído, solução de equações diferenciais parciais, codificação de canal [50, 51] e compressão de sinais de áudio, vídeo, voz e imagem.

Conforme mostra a Figura 2.4, a transformada *wavelet* discreta [48, 52–54] pode ser descrita a partir de um banco de filtros. Os filtros $h_0(n)$ e $h_1(n)$ correspondem aos filtros *wavelet* de análise, enquanto que $g_0(n)$ e $g_1(n)$ correspondem aos filtros *wavelet* de síntese. Um sinal de entrada $x(n)$ é convoluído com o filtro passa-baixa $h_0(n)$ e com o filtro passa-alta $h_1(n)$. O sinal resultante de cada convolução é então submetido a uma decimação (dizimação, sub-amostragem) de ordem 2. Geram-se, assim, no processo de decomposição (ou análise), o sinal de aproximação $a_1(n)$ e o sinal de detalhe $d_1(n)$. Em outras palavras, $a_1(n)$ contém os coeficientes *wavelet* correspondentes à componente de aproximação do sinal, ao passo que $d_1(n)$ contém os coeficientes *wavelet* referentes à componente de detalhe do sinal. No processo de reconstrução (ou síntese), procede-se uma interpolação (super-amostragem) de ordem 2, seguida de uma convolução com os filtros passa-baixa $g_0(n)$ e passa-alta $g_1(n)$. Satisfeitas as condições de reconstrução perfeita, o sinal $x(n)$ é reconstruído de forma exata, como a soma das saídas dos filtros *wavelet* de síntese, ou seja, $y(n) = x(n)$.

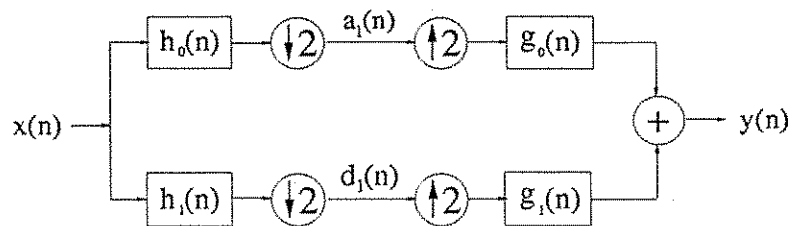


Figura 2.4: Estrutura básica do banco de filtros e das operações de decimação e interpolação envolvidos na DWT.

Em se tratando de uma decomposição *wavelet* multiresolucional (multinível) [52, 55, 56], o próximo nível de decomposição (resolução mais grosseira) é obtido mediante a convolução do sinal $a_1(n)$ com um par idêntico de filtros $h_0(n)$ e $h_1(n)$, seguida da decimação de ordem 2. Geram-se, assim, os sinais $a_2(n)$ e $d_2(n)$, em que o índice 2 diz respeito ao segundo nível de decomposição. Esse processo pode ser repetido iterativamente, e o número de pares de filtros *wavelet* de análise define o número de níveis de

decomposição (ou número de níveis de resolução), denotado por \mathcal{L} , da transformada *wavelet* discreta. Deste modo, são obtidos, ao final de \mathcal{L} níveis de decomposição, um sinal de aproximação $a_{\mathcal{L}}(n)$, com resolução reduzida por um fator de $2^{\mathcal{L}}$ em relação ao sinal de entrada, e os sinais de detalhe $d_{\mathcal{L}}(n)$, $d_{\mathcal{L}-1}(n)$, \dots , $d_1(n)$. Em outras palavras, $d_i(n)$, com $1 \leq i \leq \mathcal{L}$, contém os coeficientes *wavelet* correspondentes às componentes de detalhe no i -ésimo nível de decomposição, enquanto que $a_{\mathcal{L}}(n)$ contém os coeficientes *wavelet* referentes à componente de aproximação no \mathcal{L} -ésimo nível de decomposição. A Figura 2.5(a) ilustra a estrutura piramidal de uma decomposição *wavelet* de 3 níveis.

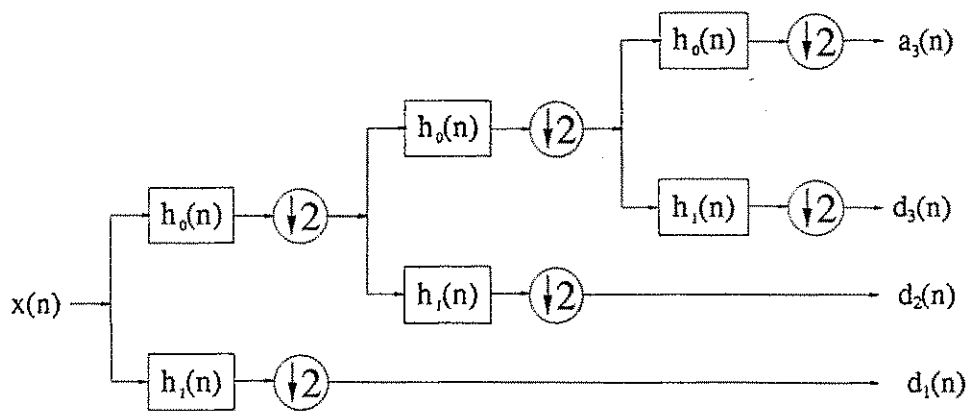
Para se realizar a etapa de síntese, ilustrada na Figura 2.5(b), os sinais $a_i(n)$ e $d_i(n)$, após interpolação de ordem 2, são submetidos, respectivamente, aos filtros *wavelet* de síntese $g_0(n)$ e $g_1(n)$ para permitirem a reconstrução de $a_{i-1}(n)$, que constitui a componente de aproximação no próximo nível de resolução mais alta. Uma vez asseguradas as condições de reconstrução perfeita, $y(n) = x(n)$. O conjunto de operações (convolução e interpolação) que permitem a reconstrução do sinal a partir dos coeficientes *wavelet* $a_{\mathcal{L}}(n)$ e $d_i(n)$, com $1 \leq i \leq \mathcal{L}$, constitui a transformada *wavelet* discreta inversa (IDWT, *inverse discrete wavelet transform*).

2.3.3 Transformada *Wavelet* Discreta Aplicada a Imagens

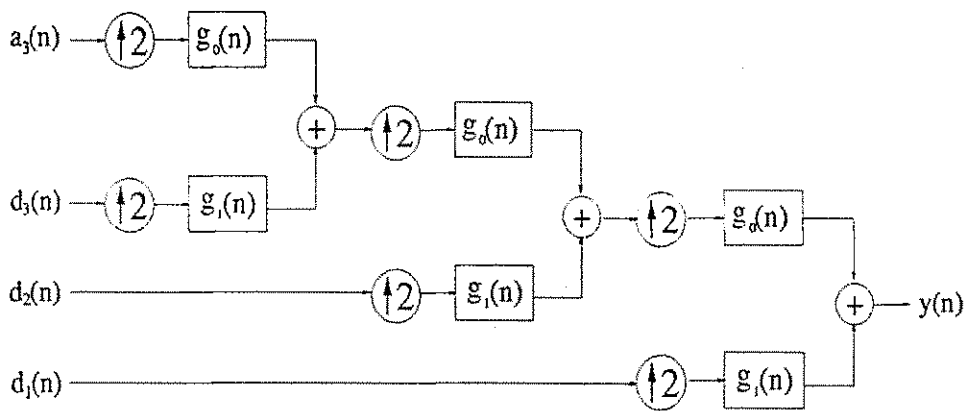
Conforme abordado na seção anterior, o princípio básico da transformada *wavelet* discreta (DWT) é a decomposição de um sinal em um conjunto de componentes de aproximação e de detalhe, mediante uso de bancos de filtros *wavelet*. Cada uma dessas componentes corresponde a uma faixa de frequências (sub-banda).

Nesta seção, procurando obedecer à notação mais comumente utilizada em transformada *wavelet* aplicada à imagem, o filtro *wavelet* passa-baixa de análise $h_0(n)$ será simplesmente denotado por L (*low-pass filter*), enquanto que o filtro *wavelet* passa-alta de análise $h_1(n)$ será simplesmente denotado por H (*high-pass filter*).

A abordagem mais utilizada para a decomposição *wavelet* multiresolucional hierárquica foi introduzida por Mallat [55]. A transformada *wavelet* discreta é implementada por meio da aplicação de um filtro passa-baixa, L , e de um filtro passa-alta, H , nas direções horizontal e vertical da imagem. Em seguida, as saídas dos filtros são sub-amostradas por um fator de 2, gerando as sub-bandas LH_1 , HL_1 , HH_1 e LL_1 no nível de decomposição (nível de resolução) 1, que corresponde à escala mais fina. Para se obterem as próximas sub-bandas (em uma escala mais grosseira), o processo é repe-



(a)



(b)

Figura 2.5: Estrutura piramidal de uma decomposição *wavelet* multiresolucional em 3 níveis: (a) operação de análise ou decomposição; (b) operação de síntese ou reconstrução.

tido na sub-banda passa-baixa LL_1 para gerar o próximo nível (resolução, escala) da decomposição, ou seja, o nível de resolução 2, com as correspondentes sub-bandas LH_2 , HL_2 , HH_2 e LL_2 . Para se obterem as sub-bandas do próximo nível de resolução, o processo é repetido em LL_2 . O processo continua até que o nível de resolução desejado seja obtido. A Figura 2.6 mostra uma decomposição *wavelet* de 3 escalas (3 níveis de resolução, decomposição em 3 níveis) de uma imagem 256×256 .

Ao longo deste trabalho será adotada a notação apresentada na Tabela 2.1, em que S_{ij} denota a sub-banda correspondente à j -ésima direção do i -ésimo nível de decomposição. Deste modo, $j = 1$ identifica a direção horizontal (HL), $j = 2$ identifica a direção vertical (LH) e $j = 3$ identifica a direção diagonal (HH), ao passo que $j = 0$ identifica a componente (sub-banda) de aproximação (LL) no nível de resolução mais baixa. Com a notação adotada neste trabalho, portanto, as componentes de detalhe são identificadas por $j = 1, 2, 3$.

A Figura 2.7 apresenta a disposição das sub-bandas obtidas em uma decomposição *wavelet* de 3 níveis de uma imagem 256×256 .

LL ₃ 32 × 32	HL ₃ 32 × 32	HL ₂ 64 × 64	HL ₁ 128 × 128 Horizontal
LH ₃ 32 × 32	HH ₃ 32 × 32		
LH ₂ 64 × 64		HH ₂ 64 × 64	HH ₁ 128 × 128 Diagonal
LH ₁ 128 × 128 Vertical			

Figura 2.6: Decomposição *wavelet* de 3 níveis de uma imagem 256×256 . A filtragem passa-baixa é denotada por L e a passa-alta por H .

A seguir será apresentada uma breve abordagem de alguns aspectos relacionados à aplicação da DWT em codificação de imagens.

Tabela 2.1: Notação utilizada.

	Sub-banda	Nível	Direção
HL_1	S_{11}	1	1 (Horizontal)
LH_1	S_{12}	1	2 (Vertical)
HH_1	S_{13}	1	3 (Diagonal)
HL_2	S_{21}	2	1 (Horizontal)
LH_2	S_{22}	2	2 (Vertical)
HH_2	S_{23}	2	3 (Diagonal)
HL_3	S_{31}	3	1 (Horizontal)
LH_3	S_{32}	3	2 (Vertical)
HH_3	S_{33}	3	3 (Diagonal)
LL_3	S_{30}	3	0

S_{30} 32×32	S_{31} 32×32	S_{21} 64×64	S_{11} 128×128 Horizontal
S_{32} 32×32	S_{33} 32×32		
S_{22} 64×64	S_{23} 64×64		
S_{12} 128×128 Vertical		S_{13} 128×128 Diagonal	

Figura 2.7: Sub-bandas obtidas em uma decomposição *wavelet* de 3 níveis de uma imagem 256×256 . A sub-banda S_{ij} diz respeito à j -ésima direção do i -ésimo nível de decomposição, $i, j \in \{1, 2, 3\}$. Observe que $j = 0$ diz respeito à sub-banda correspondente a LL .

Codificação de Imagens Utilizando DWT

A transformada *wavelet* discreta (DWT) apresenta características que justificam sua adequação (e, conseqüentemente, sua ampla utilização) em compressão de imagens, tais como [57, 58]:

- a DWT é uma descrição multiresolucional de uma imagem, de modo que é possível proceder uma decodificação seqüencial, de um nível de resolução muito baixo a um nível de resolução mais alto;
- a DWT é bem mais próxima do sistema visual humano que a transformada discreta de cosseno (DCT, *discrete cosine transform*), a qual é utilizada no padrão JPEG (*joint photographic experts group*) [59];
- as imagens reconstruídas por meio da DWT não apresentam a incômoda distorção sob a forma de bloqueamento da imagem, típica de técnicas de codificação envolvendo transformada DCT ou quantização vetorial simples (isto é, no próprio domínio espacial) a elevadas taxas de compressão;
- a representação hierárquica da imagem, em níveis de resolução distintos, sob a forma de componentes de aproximação e de detalhe, contribui para o desenvolvimento de estratégias de quantização mais “sintonizadas” com as características do sistema visual humano;
- em virtude de permitir uma representação compacta da energia da imagem (grande parte da energia se concentra em um pequeno número de coeficientes *wavelet*), a DWT favorece a concepção de quantizadores (escalares ou vetoriais) eficientes, que procuram aproveitar a estatística dos coeficientes em cada sub-banda da decomposição *wavelet*.

Diversos fatores afetam o desempenho dos sistemas de codificação de imagens que utilizam DWT, dentre os quais podem ser citados [60]: a escolha das bases (famílias ou filtros) *wavelet*, o tipo de extensão utilizada ao se aplicar a DWT, a estratégia de quantização/alocação de bits adotada, a utilização de codificadores entrópicos e a concepção/aplicação de medidas de distorção sintonizadas com critérios de percepção visual. Em seguida, cada um desses aspectos será brevemente abordado.

A escolha da base *wavelet* para utilização em codificação de imagens constitui objeto de interesse de diversas pesquisas, sendo abordada, sucinta ou profundamente,

em vários trabalhos [61–65]. Características importantes [48, 52, 53] (que influenciam a qualidade da imagem reconstruída) devem ser observadas, tais como: suavidade, regularidade, tamanho de suporte, seletividade em frequência do filtro, precisão de aproximação e número de momentos de decaimento (*vanishing moments*). Contudo, a melhor combinação desses parâmetros que caracterizam uma base *wavelet* ainda não é conhecida. Apesar de serem relatados, com grande frequência, resultados de análises comparativas de desempenho de diversas famílias *wavelet* – Daubechies (com vários comprimentos de filtro: 2, 6, 8, 10, 12, 16, 18, 20), Battle-Lemarié, Haar, Beylkin’s 6, Spline, Coiflet (com vários comprimentos de filtro: 6, 18, 30), dentre outras – o filtro biortogonal 7-9 constitui a opção mais amplamente recomendada para aplicações envolvendo compressão de imagens.

O tipo de extensão da imagem [48, 66] (necessária para se efetuar a convolução com os filtros *wavelet*) utilizada ao se aplicar a DWT pode afetar a qualidade da imagem reconstruída, obtida após aplicação da IDWT. Diversos tipos de extensão têm sido utilizados: a extensão do tipo *zero-padding* e a extensão circular podem introduzir degradação nas proximidades das bordas da imagem reconstruída; a extensão simétrica, por sua vez, apresenta-se como uma alternativa mais adequada, em virtude de levar a uma maior preservação de continuidade nas proximidades das bordas.

A estratégia de quantização dos coeficientes *wavelet* desempenha papel importantíssimo para o bom desempenho dos sistemas de codificação de imagens que utilizam DWT. Tanto a quantização escalar (utilizada em diversos trabalhos, *e.g.* [58, 67–70]) como a quantização vetorial (usada em vários trabalhos, *e.g.* [64, 65, 71–76]) têm sido aplicadas. Em se tratando de quantização escalar, algumas abordagens podem ser mencionadas, dentre as quais os quantizadores Lloyd-Max e os quantizadores com zona morta (*dead zone*). Esta última utiliza um limiar abaixo do qual um grande número de coeficientes *wavelet* são *zerados*, o que contribui para aumentar a eficiência de uma codificação entrópica subsequente. Convém mencionar que a idéia de utilizar um limiar para definição de uma zona morta também é adotada em sistemas de supressão de ruído baseados em *wavelets* [77–79].

Em se tratando, especificamente, da estratégia de alocação de bits [80, 81], o objetivo é determinar o número de bits b_{ij} alocados para cada sub-banda S_{ij} da decomposição *wavelet*, de modo que a distorção introduzida no processo de quantização seja minimizada, obedecendo à restrição $\sum_i \sum_j b_{ij} \leq b$.

De forma genérica, o problema de alocação de bits pode ser formulado da seguinte maneira. Sejam Y_1, Y_2, \dots, Y_P variáveis aleatórias (correspondentes, por exemplo, aos conjuntos de coeficientes *wavelet* em cada uma das P sub-bandas de uma decomposição multiresolucional). A alocação de bits consiste em minimizar a distorção média total sobre todas as P variáveis aleatórias dada uma restrição na taxa total de bits a ser distribuída dentre todas as variáveis aleatórias. Seja R_p a taxa alocada para a p -ésima variável aleatória e seja $D_p = E[(Y_p - Q(Y_p))^2]$ a distorção média para o p -ésimo quantizador, na qual $Q(Y_p)$ denota a saída do quantizador para Y_p e $E[\cdot]$ denota o operador expectativa. Então o objetivo a ser alcançado é minimizar a distorção média total $\sum_p D_p$ sujeita à restrição na taxa total $\sum_p R_p \leq R$. Esse problema pode ser resolvido utilizando um multiplicador de Lagrange – com essa abordagem, o problema consiste em minimizar $\sum_p D_p(R_p) + \lambda \sum_p R_p$ para algum $\lambda \geq 0$. Tomando as derivadas em relação a R_p e igualando a zero, segue que $D'_p(R_p) + \lambda = 0$ para cada p . Esse resultado traduz o método mais comum para alocação de bits: método da mesma inclinação (*“equal-slope” method*) [6], que é aplicado tanto para quantizadores escalares quanto para quantizadores vetoriais. Cada R_p é escolhido de modo a satisfazer $D'_p(R_p) = -\lambda$.

A utilização de codificadores entrópicos (tais como o código de Huffman, o código do tipo *run-length* e o código aritmético) contribui para redução da taxa de codificação, em virtude de explorar adequadamente a estatística dos símbolos utilizados na representação resultante da quantização.

A concepção/aplicação de medidas de distorção sintonizadas com critérios de percepção visual constitui importante área de estudo em codificação de imagens [82–84]. De fato, apesar de existirem diversas medidas de distorção objetivas [85–88] utilizadas para avaliar a qualidade de imagens comprimidas (reconstruídas), costumam ser relatados problemas apresentados por essas medidas, sendo freqüentemente apontada a incapacidade de modelarem de forma eficiente as características do sistema visual humano (HVS, *human visual system*) [82, 89–91], ou seja, a incapacidade de apresentarem elevada correlação com resultados de avaliação subjetiva da qualidade de imagens. Deste modo, apesar de as técnicas de codificação serem levadas a efeito no sentido de minimizar uma medida de distorção objetiva, não necessariamente se observa como resultado a obtenção de uma imagem reconstruída com elevada qualidade subjetiva. É importante mencionar, contudo, que algumas abordagens [92–94] têm sido utilizadas

para introduzir aspectos do HVS em sistemas de codificação de imagens.

Ainda com respeito às medidas de distorção utilizadas para avaliar a qualidade de imagens, cumpre registrar que, recentemente, uma medida de qualidade objetiva denominada PQS (*picture quality scale*), foi apresentada por Miyahara *et al.* em [95]. Levando em consideração tanto distorções globais como degradações locais da imagem, PQS parece apresentar-se como uma alternativa promissora para avaliação de algoritmos de compressão de imagens. Segundo Miyahara *et al.*, PQS apresenta-se fortemente correlacionada com resultados de avaliações subjetivas, sendo relatado em [95] um coeficiente de correlação de 0,92 entre PQS e a medida de qualidade subjetiva MOS (*mean opinion score*)³.

2.3.4 Codificação de Imagens Usando QV dos Coeficientes *Wavelet*

A Figura 2.8 mostra um sistema de compressão de imagens que utiliza uma decomposição *wavelet* no primeiro estágio e algum tipo de QV no segundo estágio, como por exemplo [6]: quantização vetorial com restrição de entropia (ECVQ, *entropy-constrained vector quantization*), quantização vetorial estruturada em árvore (TSVQ, *tree-structured vector quantization*), quantização vetorial multi-estágio (MSVQ, *multistage vector quantization*), quantização vetorial preditiva (PVQ, *predictive vector quantization*), quantização vetorial classificada (CVQ, *classified vector quantization*), quantização vetorial codificada em treliça (TCVQ, *trellis-coded vector quantization*), quantização vetorial hierárquica (HVQ, *hierarchical vector quantization*), *lattice vector quantization* e quantização vetorial com busca total e sem estrutura (*full-search unstructured vector quantization*). O terceiro estágio, opcional, contempla a utilização de técnicas de codificação sem perdas (codificação entrópica, codificação inversível), tais como os códigos aritmético, Lempel-Ziv e de Huffman.

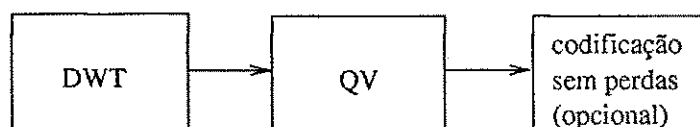


Figura 2.8: Sistema de compressão de imagens que utiliza DWT e QV.

³O escore médio de opinião (MOS) é descrito na Seção 2.5.1.

De um modo geral, os algoritmos que utilizam QV dos coeficientes de uma decomposição *wavelet* podem ser classificados de acordo com o modo de operação da QV nas sub-bandas *wavelet*, isto é, se a QV opera no modo *intra-banda* ou se ela utiliza algum tipo de informação *inter-banda*. Segundo Cosman *et al.* [6], é possível questionar o emprego do termo *intra-banda*, visto que não existe uma sistemática de quantização puramente separada das sub-bandas *wavelet* – no mínimo se utiliza alguma decisão de como alocar os bits ao longo das diversas bandas; assim, o processo de quantização não pode ser visto como completamente separado, tendo em vista que as sub-bandas compartilham um determinado número total (fixo) de bits. A essa situação, contudo, Cosman *et al.* [6] utiliza a terminologia QV *intra-banda*⁴. Os termos QV *inter-banda* e QV em bandas cruzadas (*crossband vector quantization*) são utilizados para indicar os algoritmos que procuram explorar explicitamente a correlação entre as bandas – incluem, portanto, os sistemas que quantizam os coeficientes de diferentes bandas em vetores comuns ou que utilizam explicitamente uma informação de uma banda para afetar a codificação de outra banda.

Em se tratando de codificação de imagens utilizando a decomposição *wavelet* “*octave-band*”, a QV *inter-banda* pode ser classificada em duas categorias [6]: *inter-banda* do tipo *mesma orientação* ou *inter-banda* do tipo *mesmo nível*. No primeiro caso, os vetores são formados por coeficientes *wavelet* pertencentes a sub-bandas que possuem a mesma orientação (vertical, horizontal ou diagonal). No segundo, os vetores são formados por coeficientes *wavelet* de sub-bandas pertencentes a um mesmo nível de resolução.

Em QV *inter-banda* do tipo *mesma orientação*, o vetor pode ser formado por um coeficiente *wavelet* e seus descendentes, conforme ilustra a Figura 2.9, na qual a dimensão do vetor (formado por coeficientes pertencentes a sub-bandas com orientação horizontal) é 21.

A Figura 2.10 ilustra a QV *inter-banda* do tipo *mesmo nível*. Observa-se que os coeficientes do nível de resolução 2 são agrupados em vetores de dimensão 3 e os coeficientes do nível de resolução 1 (nível de resolução mais fina) são agrupados em vetores de dimensão 12.

Tanto em QV *inter-banda* do tipo *mesma orientação* como do tipo *mesmo nível*, a maioria das técnicas utilizadas codificam separadamente a sub-banda, correspondente

⁴O sistema de codificação baseado em DWT e QV considerado no presente trabalho é do tipo *intra-banda*.

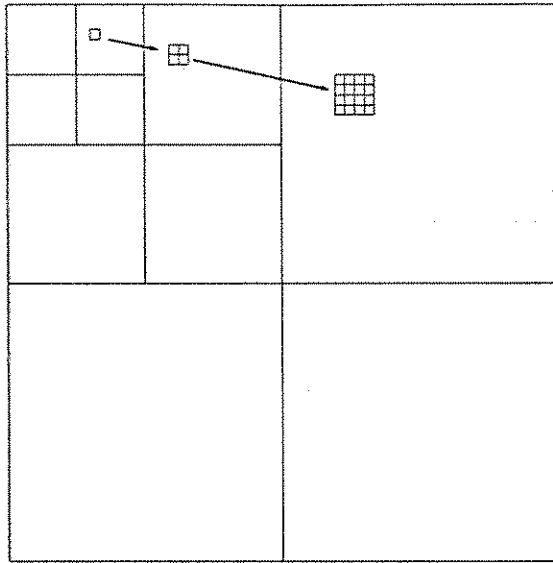


Figura 2.9: Exemplo de QV inter-banda do tipo mesma orientação.

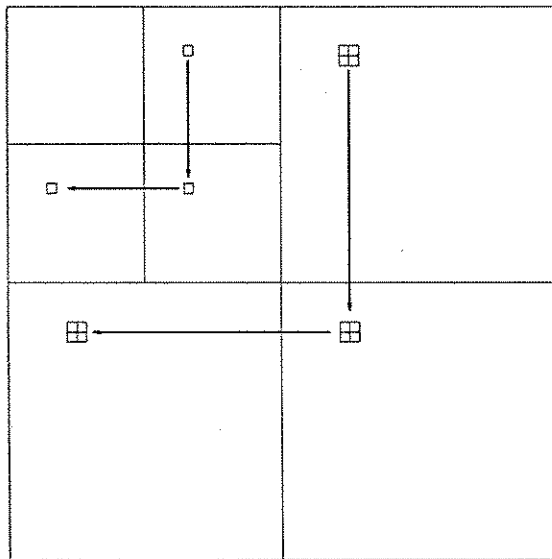


Figura 2.10: Exemplo de QV inter-banda do tipo mesmo nível.

a filtragens passa-baixa, do nível de resolução mais grosseira, em virtude de essa sub-banda apresentar características bastante distintas das demais sub-bandas *wavelet*, tanto sob o ponto de vista estatístico quanto com respeito a aspectos de percepção visual, ou seja, com relação à contribuição para a qualidade visual (avaliação subjetiva) da imagem reconstruída.

2.4 Quantização Vetorial em Canais Ruidosos

A quantização vetorial tem sido utilizada em diversos sistemas de codificação de sinais, permitindo elevadas taxas de compressão. No entanto, quando o sistema de comunicação envolve a utilização de um canal ruidoso, o desempenho do quantizador vetorial pode ser seriamente prejudicado. O ruído introduzido pelo canal, caso não seja tratado de forma adequada, pode levar a uma queda significativa de desempenho do sistema de comunicações que usa QV.

Em se tratando da transmissão de imagens, submetidas à quantização vetorial, através de canais ruidosos, quando uma palavra-binária correspondente a um vetor-código é corrompida por ruído, um bloco de $K = K_1 \times K_2$ *pixels* é afetado. Conseqüentemente, em QV de imagem através de canais ruidosos, bloqueamentos espúrios típicos, incômodos, de $K_1 \times K_2$ *pixels* são introduzidos na imagem reconstruída [96,97]⁵. No tocante à transmissão de voz baseada em QV, os erros de canal podem levar a degradações muito incômodas nos sinais reconstruídos – em [99] são relatadas sessões de escuta informais que apontaram a ocorrências de “estalos” bastante desagradáveis nos sinais de voz reconstruídos após transmissão em canal com desvanecimento.

O problema da quantização vetorial (QV) através de canais ruidosos tem sido objeto de estudo de diversas pesquisas. De forma geral, as abordagens dedicadas à minimização dos efeitos dos erros de canal no desempenho dos sistemas de comunicação baseados em QV podem ser classificadas em duas categorias. Na primeira, denominada quantização vetorial robusta (QVR), o dicionário é treinado (projetado) admitindo-se um canal sem erro. Posteriormente, por meio de um algoritmo de atribuição de índices (AI), faz-se com que o dicionário de QV torne-se robusto a erros de canal [100–102]. É importante mencionar que AI é um processo por meio do qual os

⁵Em [96] e [97] são avaliados o impacto de uma canal com desvanecimento e a influência da dimensão do quantizador vetorial na qualidade da imagem reconstruída. Além disso, é avaliado o desempenho do código espaço-temporal [98] diante das degradações introduzidas pelo canal.

vetores de reconstrução são adequadamente rotulados (indexados) de modo a reduzir o impacto dos erros de canal na qualidade do sinal reconstruído. Na segunda categoria, denominada quantização vetorial otimizada para canal, o quantizador vetorial é treinado para um canal específico, ou seja, levando-se em consideração a distorção de canal [100, 103, 104].

Quantização Vetorial Robusta

Considere o sistema de comunicação apresentado na Figura 2.11. O objetivo do sistema é transmitir uma seqüência de vetores $\mathbf{X} = \{\mathbf{x}\} \subset R^K$ através de um canal ruidoso, utilizando, para tanto, quantização vetorial dos vetores de entrada (vetores da fonte) \mathbf{x} . Seja $\mathbf{b}_i \in \{0, 1\}^m$ a palavra binária de m bits enviada pelo codificador. Devido a erros de canal, considere que o decodificador recebe a palavra binária \mathbf{b}_j , produzindo como saída o vetor reconstruído $\hat{\mathbf{x}} = \mathbf{w}_j$.



Figura 2.11: Modelo do sistema de comunicação.

Seja $p_{\mathbf{x}}$ a função densidade de probabilidades K -dimensional da fonte \mathbf{X} . Seja p_i a probabilidade *a priori* de o vetor \mathbf{w}_i ser selecionado como a reconstrução de \mathbf{x} e $p_{j|i}$ a probabilidade de o decodificador receber \mathbf{b}_j dado que \mathbf{b}_i foi enviado pelo codificador. Admitindo-se que a medida de distorção erro médio quadrático é adotada (ou seja, $d(\mathbf{x}, \mathbf{w}_i) = \|\mathbf{x} - \mathbf{w}_i\|^2$) e que a condição de centróide é satisfeita, a distorção total introduzida ao se transmitir a fonte \mathbf{X} através do canal ruidoso pode ser expressa como [100, 101]

$$\begin{aligned}
 D = D_Q + D_C &= \sum_{i=1}^N \int_{S_i} p_{\mathbf{x}} \|\mathbf{x} - \mathbf{w}_i\|^2 d\mathbf{x} \\
 &+ \sum_{i=1}^N \sum_{j=1}^N p_i p_{j|i} \|\mathbf{w}_i - \mathbf{w}_j\|^2,
 \end{aligned} \tag{2.7}$$

em que D_Q e D_C representam, respectivamente, a distorção de quantização e a distorção devida aos erros de canal.

A distorção total D pode ser reduzida através da minimização de [105]

$$I_{\text{dis}}(s) = \sum_{i=1}^N \sum_{j \in H^1(i)} \|w_i - w_j\|^2, \quad (2.8)$$

em que $j \in H^1(i)$ denota o conjunto de palavras b_j tais que a distância de Hamming para b_i é 1, s denota um arranjo (ordenamento) de dicionário e o termo $I_{\text{dis}}(s)$ é denominado índice de desordem do dicionário com ordenamento s .

Seja Π_N o conjunto de todas funções $\pi : \{0, 1\}^m \rightarrow \{0, 1\}^m$. Cada uma das $N!$ bijeções $\pi \in \Pi_N$ é denominada função de atribuição de índices do dicionário W [100]. Uma permutação π mapeia (unicamente) cada palavra $b_i \in \{0, 1\}^m$ em outra palavra $b_i' \in \{0, 1\}^m$, em que $b_i' = \pi(b_i)$. Assim, uma permutação pode ser vista como um rearranjo (um reordenamento, uma reorganização) do dicionário. Deste modo, associada a uma permutação π existe uma única organização $\pi(s)$ do dicionário. Considere, por exemplo, o conjunto de índices $s = (1, 2, 3, 4, 5, 6, 7, 8)$, correspondente ao conjunto de palavras binárias $B = (000, 001, 010, 011, 100, 101, 110, 111)$. Um novo arranjo $s' = \pi(s)$ pode ser obtido por meio de uma permutação π tal que o segundo dígito binário de cada palavra-código seja invertido. Procedendo dessa maneira, obtém-se um novo conjunto de palavras binárias $B' = (010, 011, 000, 001, 110, 111, 100, 101)$ correspondente, portanto, a $s' = (3, 4, 1, 2, 7, 8, 5, 6)$. Nesse exemplo, com a utilização da permutação π seria obtido, portanto, um “novo” dicionário (um dicionário reorganizado) W' que apresenta os mesmos vetores de reconstrução de W , contudo, em uma ordem (arranjo, organização) diferente. Em se tratando de QV em canais ruidosos, o alvo é obter uma configuração $s' = \pi(s)$ tal que $I_{\text{dis}}(s') < I_{\text{dis}}(s)$, de modo que o dicionário com ordenamento s' seja mais robusto (menos sensível) aos erros de canal que o dicionário inicial (original) com ordenamento s .

As técnicas de AI têm como objetivo organizar o dicionário de tal maneira que os erros de canal (que introduzem degradação nas palavras b_i enviadas pelo codificador) façam com que os vetores de reconstrução decodificados incorretamente (correspondentes a palavras binárias erroneamente recebidas como b_j) se aproximem, em média, dos vetores de reconstrução concernentes a uma transmissão sem erro. A determinação da configuração de dicionário ótima envolve uma elevada complexidade computacional, em virtude de existirem $N!$ configurações a serem consideradas, correspondentes a $N!$ possíveis permutações π . Por exemplo, um dicionário com $N = 16$ vetores de reconstrução tem um total de $N! = 2,09 \times 10^{13}$ permutações a serem avaliadas. O problema

de atribuição de índices para QVR, portanto, pode ser classificado como pertencente à classe dos problemas NP-completos. Desta forma, técnicas para determinação de configurações localmente ótimas devem ser consideradas.

Diversos problemas de otimização, de difícil solução, têm sido satisfatoriamente abordados com o algoritmo *simulated annealing* [106,107]. Por se tratar de um problema de otimização NP-completo, a atribuição de índices (AI) de um dicionário (atribuição de palavras binárias para os vetores de reconstrução), visando quantização vetorial robusta, pode ser obtida por meio do algoritmo *simulated annealing* (SA). Em [105] simulações envolvendo transmissão de imagens em canal com desvanecimento mostraram que o algoritmo SA é uma alternativa adequada para o problema de AI – em [105] mostrou-se também que um ganho de desempenho pode ser obtido inicializando-se o algoritmo SA com a indexação proposta em [108]⁶. Em [109] são apresentados resultados de simulações referentes à transmissão de imagens por um canal binário simétrico – para probabilidades de erro de bit iguais a 0,01 e 0,05 são relatados, respectivamente, ganhos de qualidade de 0,86 dB e 1,75 dB, em termos de relação sinal-ruído de pico⁷ da imagem Mandrill 256×256 reconstruída, ao ser utilizada a indexação obtida com SA em substituição a uma indexação arbitrária.

2.5 Avaliação de Desempenho de Quantizadores Vetoriais Aplicados à Compressão de Sinais

Um dos grandes desafios em codificação digital de sinais é a concepção e o desenvolvimento de metodologias de avaliação de qualidade de sinais reconstruídos (obtidos com a aplicação de técnicas de compressão). De forma geral, as medidas utilizadas para avaliação da qualidade de sinais enquadram-se em duas classes: medidas de qualidade subjetivas e medidas de qualidade objetivas. As primeiras baseiam-se em comparações (realizadas por meio de testes de escuta ou de visualização), entre o sinal original e o sinal processado, realizadas por um grupo de pessoas, que subjetivamente classificam a qualidade do sinal processado segundo uma escala pré-determinada. As medidas ob-

⁶O algoritmo de AI apresentado em [108] utiliza uma circunferência hipotética para ordenar (organizar) o dicionário de tal maneira que vetores-código com pequenas/grandes distância euclidianas são indexados com palavras-binárias com pequenas/grandes distâncias de Hamming.

⁷A relação sinal-ruído de pico, bastante utilizada como medida objetiva para avaliação da qualidade de imagens reconstruídas, é descrita na Seção 2.5.6.

jetivas, por sua vez, baseiam-se numa comparação matemática direta entre os sinais original e processado [110].

Para serem úteis, as medidas de qualidade objetivas devem apresentar no mínimo duas características. Primeiramente, devem ter significado subjetivo, no sentido de que pequenas e grandes variações das medidas objetivas devem significar pequenas e grandes variações da qualidade subjetiva das imagens reconstruídas, respectivamente – devem apresentar, portanto, uma correlação forte, positiva ou negativa, com resultados de avaliação subjetiva. Segundo, devem ser matematicamente tratáveis e facilmente implementáveis [111].

Medidas de qualidade subjetivas são utilizadas para avaliar de forma plena, definitiva, a qualidade de algoritmos/técnicas de codificação de sinais. Os testes subjetivos, contudo, são de difícil realização, uma vez que requerem a disponibilidade de um grande número de avaliadores (entre leigos, especialistas e possíveis usuários do sistema de codificação de sinais), envolvem grande volume de sinal processado e necessitam da disponibilidade de laboratórios com condições adequadas ao desenvolvimento das avaliações subjetivas, enfim, são bastante dispendiosos de tempo, implicando custo elevado de realização. Devido a esses problemas, as medidas de qualidade objetivas, por serem pouco dispendiosas de tempo, desempenham um papel importante no processo de avaliação de qualidade de sinais reconstruídos e constituem ferramenta valiosa no processo de ajuste de parâmetros de algoritmos/técnicas de compressão.

A avaliação de qualidade de sinais de voz e imagem tem sido objeto de estudo de diversas pesquisas, tendo sido abordada suscinta ou profundamente em diversos trabalhos (*e.g.* [83, 85, 111–118]).

A seguir serão abordadas algumas medidas de qualidade utilizadas para avaliação de desempenho de quantizadores vetoriais aplicados à compressão de sinais de voz e imagem. Serão descritos o escore médio de opinião (MOS, *mean opinion score*) e os testes de preferência, por se tratarem das avaliações subjetivas consideradas no presente trabalho. Também serão apresentadas a relação sinal-ruído (SNR, *signal-to-noise ratio*), a relação sinal-ruído segmental (SNRseg, *segmental signal-to-noise ratio*) e a distorção espectral (SD, *spectral distortion*), por serem as medidas de qualidade objetivas utilizadas neste trabalho para avaliação de qualidade dos sinais de voz. Por fim, será apresentada a relação sinal-ruído de pico (PSNR, *peak signal-to-noise ratio*), que constitui uma das medidas objetivas mais utilizadas para avaliação da qualidade de imagens reconstruídas.

2.5.1 Escore Médio de Opinião

Uma medida subjetiva bastante utilizada para avaliação de desempenho de sistemas de compressão de voz e imagem denomina-se escore médio de opinião (MOS). No teste MOS, cada avaliador atribui um escore de qualidade ao sinal reconstruído, segundo a escala graduada apresentada na Tabela 2.2 [2,110]. É calculada a média aritmética dos escores obtidos e determinado o valor final da avaliação, ou seja,

$$MOS = \frac{1}{L} \sum_{l=1}^L s_l, \quad (2.9)$$

em que L é o número de avaliadores utilizados no teste e s_l é o escore atribuído pelo l -ésimo avaliador.

Tabela 2.2: Escala para o teste MOS.

Escore (s)	Qualidade	Nível de Distorção
5	Excelente	Imperceptível
4	Boa	Apenas perceptível mas não incômodo
3	Razoável	Perceptível e pouco incômodo
2	Pobre	Incômodo
1	Ruim	Muito incômodo

2.5.2 Testes de Preferência

Os testes de preferência são realizados por comparação entre pares de sinais.

Uma forma de realização desse tipo de avaliação subjetiva consiste em conceder um conceito de acordo com três possíveis resultados de comparação, ou seja [119]:

- Conceito A - A qualidade do primeiro sinal é melhor do que a do segundo;
- Conceito B - A qualidade do segundo sinal é melhor do que a do primeiro;
- Conceito C - A qualidade de ambos sinais não se distingue.

Outra forma de realização de testes de preferência, muito comum para a avaliação de imagens, consiste em proceder a comparação com relação a uma imagem de referência,

utilizando-se uma escala de graduação com valores que variam de um a cinco, na qual cada valor correspondente a um conceito obtido do processo de comparação [119]:

- Conceito 5 - A imagem sob teste tem qualidade muito superior à apresentada pela imagem de referência;
- Conceito 4 - A imagem sob teste tem qualidade um pouco superior à apresentada pela imagem de referência;
- Conceito 3 - A imagem sob teste tem a mesma qualidade da apresentada pela imagem de referência.
- Conceito 2 - A imagem sob teste tem qualidade um pouco inferior à apresentada pela imagem de referência;
- Conceito 1 - A imagem sob teste tem qualidade muito inferior à apresentada pela imagem de referência.

2.5.3 Relação Sinal-Ruído (SNR)

Sejam $x(n)$ o sinal original, $y(n)$ o sinal processado e $e(n) = x(n) - y(n)$ o sinal erro no instante de tempo n .

A energia contida no sinal original é

$$E_x = \sum_n x^2(n). \quad (2.10)$$

A energia contida no sinal erro é

$$E_e = \sum_n e^2(n) = \sum_n [x(n) - y(n)]^2. \quad (2.11)$$

A medida SNR resultante, expressa em dB, é dada por

$$\text{SNR} = 10 \log_{10} \frac{E_x}{E_e} = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n [x(n) - y(n)]^2}. \quad (2.12)$$

2.5.4 Relação Sinal-Ruído Segmental (SNRseg)

Apesar da simplicidade matemática, a medida SNR apresenta uma incômoda limitação: a igual ponderação de todos os erros no domínio do tempo. Por essa razão, um indesejável alto valor de SNR pode ser obtido se uma seqüência de fala apresentar alta concentração de segmentos vocais, sonoros, de alta energia, uma vez que o efeito do ruído é maior nos segmentos de baixa energia, como por exemplo os sons fricativos surdos.

Uma medida de qualidade mais refinada pode ser obtida se for tomada a média da relação sinal-ruído medida em curtos intervalos de tempo. É definida, então, a relação sinal-ruído segmental (ou razão sinal-ruído segmentar, seguindo a terminologia adotada em [120]):

$$\text{SNRseg} = E[\text{SNR}(j)], \quad (2.13)$$

em que $\text{SNR}(j)$ denota a relação sinal-ruído (SNR) convencional⁸ para o j -ésimo segmento (janela de tempo) do sinal.

A medida SNRseg é formulada da seguinte forma:

$$\text{SNRseg} = \frac{1}{J} \sum_{j=0}^{J-1} 10 \log_{10} \left[\frac{\sum_{n=m_j-N_A-1}^{m_j} x^2(n)}{\sum_{n=m_j-N_A-1}^{m_j} [x(n) - y(n)]^2} \right], \quad (2.14)$$

em que m_0, m_1, \dots, m_{J-1} são os instantes finais para as J janelas de tempo, de N_A amostras, de comprimento típico de 15 a 25 ms [110, 119].

Segundo Voran [116], as medidas SNR e SNRseg podem fornecer uma indicação de qualidade subjetiva em alguns codificadores de voz que têm por objetivo a representação da forma de onda. Contudo, quando utilizadas em sistemas de codificação e transmissão mais gerais, SNR e SNRseg apresentam pouca correlação com resultados de avaliação subjetiva da qualidade de voz. Voran [116] atribui a popularidade de SNR e SNRseg a razões históricas (tratam-se de medidas clássicas, utilizadas há bastante tempo), à simplicidade dessas medidas, e ao fato de inexistirem medidas objetivas que tenham sido amplamente testadas e aceitas, de tal maneira que possam substituir SNR e SNRseg.

⁸Ao longo do presente trabalho, a relação sinal-ruído convencional (SNR) também é denotada por SNRtot (relação sinal-ruído total).

2.5.5 Distorção Espectral

Para aplicações que requisitam codificação de voz a baixas taxas (como é o caso de sistemas de comunicações móveis), é imprescindível quantizar precisa e adequadamente os parâmetros LSF, utilizando o menor número de bits possível. O desenvolvimento de métodos de codificação LSF que utilizam quantização vetorial tem sido objeto de interesse de muitos pesquisadores [121–132].

A qualidade da quantização dos parâmetros LSF é avaliada por meio da distorção espectral (SD), definida como

$$SD = \left[\frac{1}{F_s} \int_0^{F_s} [10 \log_{10} S(f) - 10 \log_{10} \hat{S}(f)]^2 df \right]^{1/2}, \quad (2.15)$$

em que $S(f)$ e $\hat{S}(f)$ denotam, respectivamente, a envoltória espectral original e quantizada.

2.5.6 Relação Sinal-Ruído de Pico (PSNR)

Dentre as diversas medidas objetivas utilizadas para avaliação de qualidade de imagens [85–88] a relação sinal-ruído de pico (ou razão pico-ruído, seguindo a terminologia adotada em [133]) apresenta-se como a opção mais utilizada, apesar de serem freqüentemente registradas críticas, como relata por exemplo [134], no que diz respeito à correlação de PSNR com resultados de avaliações subjetivas.

A relação sinal-ruído de pico é definida como 10 vezes o logaritmo na base 10 da razão entre o quadrado do valor de pico da amplitude de entrada, (v_p^2) , e o erro médio quadrático (MSE, *mean square error*):

$$PSNR = 10 \log_{10} \left[\frac{(v_p^2)}{MSE} \right]. \quad (2.16)$$

Para o caso de uma imagem original codificada a 8,0 bpp,

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right], \quad (2.17)$$

em que o erro médio quadrático entre as imagens original e reconstruída é definido como

$$MSE = \frac{1}{256^2} \sum_{l=1}^{256} \sum_{c=1}^{256} [F(l, c) - \hat{F}(l, c)]^2, \quad (2.18)$$

em que $F(l, c)$ e $\hat{F}(l, c)$ representam os valores de *pixels* das imagens original e reconstruída, l designa a l -ésima linha e c denota a c -ésima coluna de uma imagem (matriz) 256×256 .

2.6 Quantização Vetorial Aplicada ao Reconhecimento de Locutor

2.6.1 Preliminares

A comunicação vocal homem-máquina divide-se em três áreas principais [135]: sistemas de resposta vocal, sistemas de reconhecimento de fala e sistemas de reconhecimento de locutor.

Os sistemas de resposta vocal são projetados para responder a algum pedido de informação utilizando mensagens faladas. A comunicação da voz, portanto, é realizada em uma única direção: da máquina para o usuário. Os sistemas de resposta vocal podem ser utilizados em várias aplicações, tais como: sistemas automáticos de informação de preços, de vôos, de divulgação de produtos em supermercados ou lojas especializadas e sistemas bancários [119].

Os sistemas de reconhecimento de fala têm como objetivo reconhecer uma determinada elocução de uma sentença ou “entender” a expressão falada, ou seja, responder corretamente ao que se está falando [135, 136]. Esses sistemas desempenham papel importante quando uma ou mais das seguintes condições se aplicam [136]: as mãos do usuário estão ocupadas; os olhos do operador estão fixos em um *display*, um instrumento ótico ou um objeto a ser rastreado; é inconveniente o uso de teclado em um ambiente. Deste modo, os sistemas de reconhecimento de fala, por não requererem nem os olhos nem as mãos do usuário para funcionarem, podem ter diversas aplicações [137], tais como: controle de tráfego aéreo, controle de dados cartográficos e barométricos e auxílio a deficientes físicos.

Os sistemas de reconhecimento de locutor podem ser classificados em sistemas de identificação de locutor e sistemas de verificação de locutor [135, 138–140]. No primeiro caso, o objetivo é identificar a pessoa mais provável de ser o locutor dentro de um conjunto pré-estabelecido de possíveis locutores. Em se tratando de verificação de locutor, o objetivo do sistema é verificar se o locutor é quem ele alega ser [141]. A verificação de locutor, portanto, é realizada por meio de uma comparação com um único padrão pré-estabelecido, enquanto que a identificação de locutor realiza comparação com todos os padrões pré-estabelecidos. Os sistemas de reconhecimento de locutor [139, 140, 142–144] são bastante úteis em aplicações de segurança, como controle de acesso a ambientes restritos (utilização da voz para abrir e fechar portas) e controle de acesso

de dados em computador. Em criminalística, os sistemas de reconhecimento de locutor podem ser utilizados com o mesmo propósito que é dado às impressões digitais [119].

2.6.2 Técnicas para Reconhecimento de Locutor

Os parâmetros acústicos utilizados nas técnicas de reconhecimento de locutor podem ser obtidos diretamente dos métodos de análise e codificação de voz. Dentre os parâmetros comumente utilizados, podem ser citados [145-148]: coeficientes LPC, coeficientes cepestrais, coeficientes cepestrais ponderados, coeficientes delta cepestrais e coeficientes delta cepestrais ponderados. Esses parâmetros visam capturar informação espectral suficiente para permitir a tarefa de reconhecimento [139].

Os métodos conhecidos para reconhecimento de locutor diferenciam-se na forma como os parâmetros acústicos extraídos do sinal de voz são utilizados. Podem ser identificados dois grandes grupos: métodos paramétricos e métodos estatísticos [149].

Nos métodos paramétricos, após a detecção de fim de palavra, é realizada uma redução de dados explícita, após a qual é obtido um padrão de referência que continua ainda na forma paramétrica. A regra de decisão no processo de comparação de padrões de referência e de teste baseia-se em medidas de distância [141].

Nos métodos estatísticos, a construção de padrões é obtida por meio de modelos estatísticos, tais como os modelos de Markov escondidos [150, 151]. Os parâmetros acústicos extraídos do sinal de voz são, com o auxílio da teoria das probabilidades, representados por modelos estocásticos nos quais está presente uma redução implícita de dados. Nesses métodos não é feita uma comparação direta de padrões; a decisão é tomada através do cálculo de probabilidades associadas aos modelos [141].

Diversas técnicas têm sido utilizadas para reconhecimento de locutor, dentre as quais destacam-se: alinhamento dinâmico no tempo (DTW, *dynamic time warping*) [110, 140], quantização vetorial [138, 148, 152-156], modelos de Markov escondidos (HMMs, *hidden Markov models*) [141, 157, 158] e redes neurais [159-161].

Em sistemas de reconhecimento de locutor, a quantização vetorial (QV) permite uma eficiente taxa de redução de dados quando aplicada aos vetores acústicos obtidos a partir do processo de parametrização da voz. Outro mérito da quantização vetorial aplicada a reconhecimento de locutor reside no fato de que essa técnica dispensa a necessidade de alinhamento temporal [144, 153], no sentido de que permite ao sistema de reconhecimento flexibilidade quanto ao tamanho das sentenças proferidas pelos lo-

cutores. Em diversos sistemas de reconhecimento de locutor, a quantização vetorial é responsável pela geração e utilização de dicionários para determinação da similaridade de padrões vocais. Nesse contexto, a tarefa de reconhecimento de locutor utiliza duas fases: treinamento e reconhecimento. Realizada *off-line*, a fase de treinamento tem como objetivo produzir uma memória de referência, ou seja, um dicionário de padrões acústicos de referência. Na fase de reconhecimento, geralmente realizada em tempo real, são obtidos padrões de teste que são comparados com os padrões de referência previamente armazenados e então, utilizando-se uma regra de decisão, para cada padrão de teste é determinado o padrão de referência mais semelhante. No sistema de identificação de locutor, portanto, o locutor é identificado como aquele que proporcionar o melhor “casamento” entre os padrões de teste e os padrões de referência. Esse tipo de identificação pode ocorrer de duas formas [139,149]: conjunto-aberto (o locutor pode não fazer parte da população) e conjunto-fechado (sabe-se *a priori* que o locutor é um membro da população). Além disso, a identificação pode ser dependente do texto (o locutor pronuncia uma senha ou sentença pré-definida) ou independente do texto (não se exige que os locutores profiram uma elocução pré-estabelecida) [138,144,146].

A seguir é apresentada uma breve descrição do sistema de reconhecimento de locutor, dependente do texto, baseado em quantização vetorial paramétrica, considerado neste trabalho.

2.6.3 Descrição do Sistema de Reconhecimento de Locutor

O sistema de reconhecimento automático da identidade vocal, baseado em quantização vetorial paramétrica, desenvolvido no presente trabalho é composto das seguintes etapas:

1. Processamento do sinal;
2. Extração de características;
3. Quantização vetorial;
4. Regra de decisão.

Conforme ilustra a Figura 2.12, de um sinal de voz amostrado, após realização de uma etapa de processamento (que consiste de pré-ênfase, segmentação e janelamento do

signal), é extraída uma seqüência de vetores de parâmetros acústicos, que é então submetida à quantização vetorial, após a qual o reconhecimento (identificação do locutor) é realizado segundo uma regra de decisão.

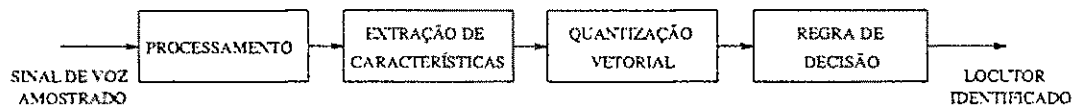


Figura 2.12: Etapas do processo de reconhecimento de locutor baseado em quantização vetorial paramétrica.

Como toda tarefa de reconhecimento de padrões, o sistema se divide em duas fases: treinamento e reconhecimento (identificação), descritas a seguir.

Fase de Treinamento

A fase de treinamento do sistema de identificação de locutor baseado em quantização vetorial paramétrica, ilustrada na Figura 2.13, tem como objetivo gerar, para cada locutor cadastrado, um conjunto de padrões acústicos de referência – nessa fase é gerado um dicionário de padrões acústicos para cada locutor cadastrado. Nesse sentido, o algoritmo de projeto de dicionários desempenha um papel importante: cada dicionário projetado deve representar, de forma eficiente, as características vocais do locutor por ele representado. A fase de treinamento consiste das seguintes etapas:

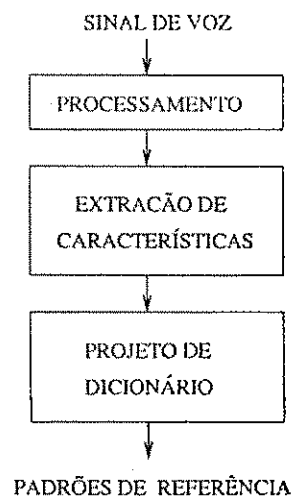


Figura 2.13: Diagrama de blocos da fase de treinamento do sistema de reconhecimento (identificação) de locutor.

1. Elocução da sentença de treinamento – cada locutor cadastrado pelo sistema profere uma sentença; a cada locutor cadastrado pelo sistema está associado um sinal de voz, que será adequadamente submetido a um processamento (na etapa 2) e do qual será extraída uma seqüência de parâmetros acústicos (na etapa 3) para treinamento do dicionário (na etapa 4);
2. Processamento do sinal – corresponde à realização de pré-ênfase [110,135], seguida de segmentação em blocos e janelamento [110,135];
3. Extração de características – para cada locutor, a partir da elocução da sentença de treinamento, adequadamente processada, é obtida uma seqüência de treinamento do dicionário, constituída de parâmetros acústicos. Em geral, esses parâmetros são obtidos por meio de análise por predição linear, devendo ser capazes de representar eficientemente a identidade vocal. Dentre os parâmetros acústicos mais utilizados, destacam-se os coeficientes LPC, cepestrais, cepestrais ponderados, delta cepestrais e delta cepestrais ponderados;
4. Projeto de dicionário – para cada um dos M_L locutores cadastrados pelo sistema, é projetado um dicionário de padrões acústicos de referência. Assim, se o sistema de reconhecimento de locutor utilizar como parâmetros acústicos os coeficientes LPC, será projetado um dicionário de coeficientes LPC de referência para cada locutor. Por outro lado, se o sistema de reconhecimento de locutor utilizar como parâmetros acústicos os coeficientes cepestrais, será projetado um dicionário de coeficientes cepestrais de referência para cada locutor.

Em resumo, a fase de treinamento do sistema de identificação de locutor baseado em quantização vetorial paramétrica tem como objetivo definir (projetar) e armazenar, para cada um dos M_L locutores cadastrados, um conjunto de padrões acústicos de referência, isto é, um dicionário de padrões acústicos de referência.

Fase de Identificação

A fase de identificação, cujo diagrama de blocos é apresentado na Figura 2.14, compreende as seguintes etapas:

1. Elocução da sentença de teste – o locutor que solicita identificação ao sistema de reconhecimento profere uma sentença; do sinal de voz desse locutor, após

processamento adequado (na etapa 2), é extraída uma seqüência de parâmetros acústicos de teste (na etapa 3);

2. Processamento do sinal – corresponde à realização de pré-ênfase, seguida de segmentação em blocos e janelamento;
3. Extração de características – a partir da sentença de teste, devidamente processada, é obtida uma seqüência de parâmetros acústicos de teste (padrões de teste), que podem ser, por exemplo, coeficientes LPC, cepestrais, cepestrais ponderados, delta cepestrais ou delta cepestrais ponderados;
4. Comparação dos padrões acústicos de teste com o conjunto de padrões acústicos de referência (previamente definidos e armazenados durante a fase de treinamento do sistema de reconhecimento) dos M_L locutores cadastrados pelo sistema;
5. Utilização de uma regra de decisão para identificação do locutor.

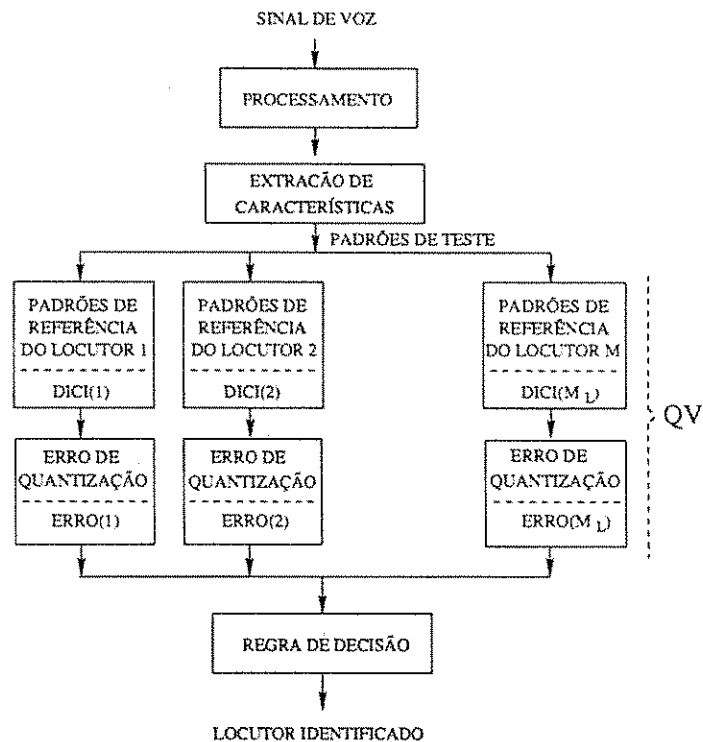


Figura 2.14: Diagrama de blocos da fase de identificação do sistema de reconhecimento de locutor.

É importante observar que a seqüência de parâmetros acústicos de teste é representada individualmente por meio de cada dicionário (DICI(1), DICI(2),...,DICI(M_L)) de padrões acústicos de referência. O erro de representação da seqüência de parâmetros acústicos de teste por sua correspondente versão quantizada (ou seja, o erro de quantização) é determinado para cada um dos M_L dicionários de padrões acústicos de referência. Existem duas formas de decisão:

1. Sem a utilização de um limiar de rejeição. Neste caso, o locutor correspondente ao menor erro médio de quantização⁹ é apontado como o locutor reconhecido pelo sistema. Esta forma de decisão caracteriza os sistemas de reconhecimento que operam com um conjunto-fechado de locutores – apenas os locutores cadastrados¹⁰ solicitam reconhecimento;
2. Com a utilização de um limiar de rejeição. Neste caso, o locutor correspondente ao menor erro médio de quantização é apontado como o locutor reconhecido se tal erro médio for inferior a um determinado limiar. Esta forma de decisão caracteriza os sistemas de reconhecimento que operam com um conjunto-aberto de locutores. A definição do limiar de rejeição procura descartar a possibilidade de reconhecimento de locutores não cadastrados¹¹ pelo sistema.

A seção a seguir apresenta uma breve abordagem da análise por predição linear ou análise LPC (*linear prediction coding*), a partir da qual são obtidos os coeficientes LPC. Por meio de técnicas derivadas da análise LPC, são obtidos outros parâmetros acústicos utilizados em sistemas de reconhecimento de locutor – coeficientes cepestrais, delta cepestrais e delta cepestrais ponderados, dentre outros.

2.6.4 Análise por Predição Linear

A análise por predição linear tem se destacado como uma poderosa técnica de análise de sinais de voz, constituindo a metodologia predominante para a estimação dos

⁹Em virtude de utilizar o erro médio de quantização, o sistema de reconhecimento de locutor considerado no presente trabalho dispensa a necessidade de alinhamento temporal. O sistema, portanto, tem flexibilidade quanto ao tamanho da sentença preferida pelos locutores.

¹⁰Para os quais foram previamente projetados dicionários de parâmetro acústicos na etapa 4 da fase de treinamento.

¹¹Para os quais não foram previamente projetados dicionários de parâmetro acústicos na etapa 4 da fase de treinamento.

parâmetros básicos da fala, como *pitch*, formantes e funções do trato vocal. Em codificação de voz, essa técnica pode ser utilizada para representar de forma eficiente a fala, visando transmissão e armazenamento a baixas taxas – nesse cenário estão definidos, dentre outros, os codificadores CELP (*code-excited linear prediction*) e VSELP (*vector sum excited linear prediction*) [110]. Convém salientar que a importância da análise preditiva linear está ligada à sua capacidade de fornecer estimativas extremamente eficientes dos parâmetros da voz e à sua alta velocidade de cálculo [135, 136].

O processo de predição linear consiste em estimar o valor de uma amostra atual do sinal $s(n)$ a partir de uma combinação linear dos valores das amostras passadas $s(n - k)$; $k = 1, 2, \dots, K$:

$$\hat{s}(n) = s(n) * h(n), \quad (2.19)$$

ou seja,

$$\hat{s}(n) = \sum_{k=1}^K c_k s(n - k), \quad (2.20)$$

em que $s(n)$ representa o sinal de voz discreto no tempo, $\hat{s}(n)$ representa o sinal de voz estimado por meio da predição linear e c_k , $k = 1, 2, \dots, K$, representa os coeficientes do preditor de ordem K .

A resposta ao impulso, $h(n)$, do preditor é definida como

$$h(n) = \sum_{k=1}^K c_k \delta(n - k), \quad (2.21)$$

e a correspondente função de transferência é

$$H_K(z) = \sum_{k=1}^K c_k z^{-k}. \quad (2.22)$$

A Figura 2.15 ilustra a estrutura geral do processo de predição linear. A resposta ao impulso do preditor deve ser tal que forneça a melhor estimaco $\hat{s}(n)$, ou seja, o preditor é otimizado de forma a fornecer o menor erro de estimaco $e(n)$, dado por

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^K c_k s(n - k). \quad (2.23)$$

O critrio de otimizao do preditor consiste em minimizar a varincia do erro de estimaco, ou seja,

$$\sigma_e^2 = E\{[s(n) - \hat{s}(n)]^2\} \doteq \min. \quad (2.24)$$

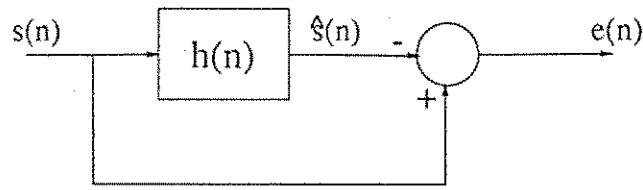


Figura 2.15: Estrutura geral do processo de predição linear: $h(n)$ representa a resposta ao impulso do preditor, $s(n)$ representa o sinal de voz discreto no tempo, $\hat{s}(n)$ denota o sinal de voz estimado por meio da predição linear e $e(n)$ denota o erro de estimação.

Os coeficientes do preditor são otimizados substituindo-se o valor do sinal estimado, conforme a Equação (2.20), na Equação (2.24) e minimizando-se o erro de estimação, $e(n)$, em relação a cada um dos coeficientes c_k , ou seja,

$$\frac{\partial \sigma_e^2}{\partial c_k} = 0. \quad (2.25)$$

Obtém-se, então,

$$R_{ss}(i) = \sum_{k=1}^K c_k R_{ss}(i-k); i = 1, 2, \dots, K, \quad (2.26)$$

em que R_{ss} denota a função de autocorrelação do sinal $s(n)$.

Para o cálculo dos coeficientes ótimos do preditor, portanto, se faz necessária apenas a determinação dos coeficientes da função de autocorrelação do sinal [119].

Os coeficientes do preditor são calculados a partir do sistema de equações descrito pela Expressão (2.26), conhecida como Equação de Wiener-Hopf, a qual pode ser apresentada segundo a forma matricial

$$\begin{bmatrix} R_{ss}(0) & R_{ss}(1) & R_{ss}(2) & \cdots & R_{ss}(K-1) \\ R_{ss}(1) & R_{ss}(0) & R_{ss}(1) & \cdots & R_{ss}(K-2) \\ R_{ss}(2) & R_{ss}(1) & R_{ss}(0) & \cdots & R_{ss}(K-3) \\ \vdots & \vdots & \vdots & & \vdots \\ R_{ss}(K-1) & R_{ss}(K-2) & R_{ss}(K-3) & \cdots & R_{ss}(0) \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_K \end{bmatrix} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ R_{ss}(3) \\ \vdots \\ R_{ss}(K) \end{bmatrix}. \quad (2.27)$$

O sistema pode ser resolvido de forma recursiva utilizando o algoritmo de Levinson-Durbin, que aproveita a estrutura Töeplitz da matriz de coeficientes de autocorrelação para cálculo rápido dos coeficientes do preditor [119].

Cumpra ressaltar que, devido à natureza variante no tempo do sinal de voz, os coeficientes do preditor devem ser estimados em segmentos (blocos de amostras) de curtos intervalos de tempo.

Nesta seção foi abordado o método da autocorrelação para cálculo dos coeficientes LPC, denotados por c_k . Em [135], são apresentados outros métodos para determinação dos coeficientes LPC, dentre os quais: método da covariância, formulação do filtro inverso, formulação da estimação espectral, formulação da máxima verossimilhança e formulação do produto interno.

A seguir são apresentados os coeficientes cepestrais, delta cepestrais e delta cepestrais ponderados.

Coefficientes Cepestrais

Os coeficientes cepestrais vêm sendo bastante utilizados como parâmetros acústicos destinados à identificação de locutores, em virtude de serem apropriados para modelagem do trato vocal, a qual é fundamental para distinção de locutores [146]. Esses coeficientes, denotados por ce_i , são usados para descrever a envoltória espectral do sinal de voz a curtos intervalos de tempo. Podem ser calculados recursivamente a partir dos coeficientes LPC, c_i , por meio da expressão [145, 146, 162, 163]

$$ce_i(n) = c_i(n) + \sum_{j=1}^{n-1} \binom{j}{n} ce_i(j) c_i(n-j), \quad 1 \leq n \leq K, \quad (2.28)$$

em que n denota o índice do coeficiente e i representa o índice do bloco de amostras no qual os coeficientes LPC foram previamente determinados.

Coefficientes Delta Cepestrais

Os coeficientes cepestrais representam as propriedades espectrais de um determinado bloco de amostras de voz. No entanto, não caracterizam a informação temporal ou de transição de uma seqüência de blocos de amostras de voz. Para aplicações relacionadas a texto, tais como reconhecimento de fala e reconhecimento de locutor dependente de texto, um melhor desempenho pode ser obtido com a introdução da derivada cepestral no espaço de características, em virtude de essa derivada ser capaz de capturar a informação de transição da voz [146]. Nesse cenário, são definidos os coeficientes delta cepestrais, que podem ser obtidos a partir de uma versão simplificada da equação que define a primeira derivada do cepstrum, também conhecida como delta cepstrum.

Os coeficientes delta cepestrais, denotados por Δce_i , podem ser obtidos da seguinte forma [146,162]:

$$\Delta ce_i(n) = \left[\sum_{q=-Q}^Q q ce_{i-q}(n) \right] G, \quad 1 \leq n \leq K, \quad (2.29)$$

em que:

G - termo de ganho;

K - número de coeficientes delta cepestrais calculados em um bloco de amostras;

$2Q + 1$ - número de blocos de amostras sobre os quais o cálculo é realizado;

n - índice do coeficiente;

i - índice do bloco de amostras no qual os coeficientes cepestrais foram previamente determinados.

Coefficientes Delta Cepstrais Ponderados

Um dos principais objetivos da introdução de uma operação de ponderação nos coeficientes delta cepestrais é minimizar a influência do ruído nos coeficientes de alta ordem. Essa operação também pode ter como propósito reduzir a influência da envoltória espectral nos coeficientes de baixa ordem.

A ponderação consiste em multiplicar $\Delta ce_i(n)$ por uma janela $jp(n)$. A operação de ponderação é também conhecida como filtragem ou suavização (*liftering*).

As técnicas de ponderação diferem de acordo com o tipo de janela utilizada. A mais simples é a janela retangular, dada por [146]

$$jp(n) = \begin{cases} 1, & n = 1, 2, \dots, K \\ 0, & \text{caso contrário} \end{cases}, \quad (2.30)$$

sendo K o tamanho da janela.

Outras formas de janelamento incluem a ponderação linear (*quefreny liftering*), em que [146]

$$jp(n) = \begin{cases} n, & n = 1, 2, \dots, K \\ 0, & \text{caso contrário} \end{cases} \quad (2.31)$$

e a filtragem (ou suavização) passa-faixa (BPL - *bandpass liftering*), na qual [146]

$$jp(n) = \begin{cases} 1 + \frac{K}{2} \text{sen}\left(\frac{n\pi}{K}\right), & n = 1, 2, \dots, K \\ 0, & \text{caso contrário} \end{cases}. \quad (2.32)$$

Por meio da ponderação dos coeficientes delta ceestrais por uma dessas janelas, obtém-se, portanto, um conjunto de coeficientes delta ceestrais ponderados, expressos por [146, 162]

$$\Delta cp_i(n) = \Delta ce_i(n) \cdot jp(n). \quad (2.33)$$

Coeficientes Ceestrais Ponderados

Os coeficientes ceestrais ponderados, $cp_i(n)$, são obtidos aplicando uma operação de ponderação nos coeficientes ceestrais. São determinados a partir da expressão

$$cp_i(n) = ce_i(n) \cdot jp(n). \quad (2.34)$$

Capítulo 3

Algoritmos Não-Supervisionados Aplicados à Quantização Vetorial

3.1 Introdução

Uma rede neural é projetada para modelar a maneira como o cérebro desempenha uma tarefa particular ou uma função de interesse. Assemelha-se ao cérebro em dois aspectos:

1. Conhecimento é adquirido pela rede por meio de um processo de aprendizagem;
2. Conexões interneuronais conhecidas como pesos sinápticos são utilizadas para armazenar o conhecimento.

As redes neurais são compostas de muitos elementos simples (neurônios), interconectados por meio de ligações (pesos), de acordo com uma arquitetura. Alguns desses elementos recebem informações externas e as distribuem para a rede, na qual as informações são processadas. O comportamento da rede depende de sua arquitetura e dos valores de seus pesos. A idéia de aprendizagem em uma rede neural é implementada por meio da modificação dos pesos de tal forma que, à medida que a informação é transmitida à rede, o processamento desejado é obtido [23].

Como principais aplicações de redes neurais, podem ser citadas [164, 165]: controle de processos, controle de motores elétricos, controle de robôs, controle de veículos, diagnóstico médico, processamento de sinais médicos, reconhecimento de padrões (*e.g.* reconhecimento de caracteres [166, 167] e reconhecimento de fala), aplicações financeiras, linguagem natural, sensoriamento remoto, processamento de sinais de voz e

imagem, processamento de sinais sísmicos, inspeção automática, realidade virtual e comunicação digital.

Dentre as regras gerais para representação do conhecimento em redes neurais, destacam-se três [20]:

- *Regra 1:* entradas similares devem produzir na rede neural representações internas semelhantes e devem, portanto, ser classificadas como pertencentes à mesma classe;
- *Regra 2:* entradas pertencentes a classes distintas de padrões devem resultar em representações internas bastante diferentes em uma rede neural;
- *Regra 3:* se uma determinada característica é importante, então um grande número de neurônios deve ser utilizado para a representação de tal característica na rede neural.

Quanto à forma de aprendizagem, os algoritmos utilizados durante a fase de treinamento de redes neurais [9, 20, 107, 168–170] podem ser classificados em duas categorias: algoritmos supervisionados e algoritmos não-supervisionados.

As redes que utilizam algoritmo de treinamento supervisionado recebem informação externa sobre a que classe pertence cada padrão de entrada utilizado na fase de treinamento. De posse dessa informação, os pesos das unidades de processamento são modificados de forma a se obter o desempenho desejado, ou seja, uma classificação satisfatória. Um dos algoritmos de treinamento supervisionado mais utilizados é o algoritmo *backpropagation* ou algoritmo de propagação retroativa de erros, que descreve a metodologia de modificação dos pesos na rede neural denominada perceptron multicamadas [20, 168, 169].

Uma característica marcante da aprendizagem supervisionada é a presença de um supervisor externo, conforme ilustra a Figura 3.1. O supervisor é simplesmente o elemento que detém o conhecimento do problema a ser solucionado pela rede, ou seja, ele conhece o mapeamento entrada-saída. O supervisor fornece à rede um conjunto de exemplos entrada-saída. Em outras palavras, a *resposta desejada* (resposta correta) é fornecida à rede pelo supervisor. Essa resposta desejada representa a ação ótima a ser executada pela rede. Nesse contexto, os parâmetros da rede são ajustados de forma a minimizar o *signal erro*, definido como a diferença entre a *resposta real*, apresentada pela rede, e a *resposta desejada*, fornecida pelo supervisor.

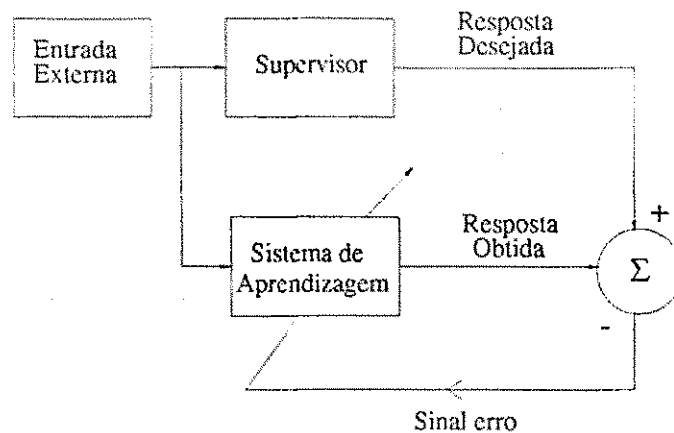


Figura 3.1: Aprendizagem supervisionada.

Na aprendizagem não-supervisionada ou auto-organizativa [20,168], não existe um supervisor externo para orientar o processo de aprendizagem. Nos algoritmos de treinamento não-supervisionado, portanto, nenhuma informação externa referente às classes a que pertencem os dados de treinamento é disponível à rede. As modificações efetuadas nos pesos são realizadas com base apenas na observação de erros internos. O principal objetivo do algoritmo de treinamento não-supervisionado é capturar as regularidades estatísticas dos dados de entrada, ou seja, desenvolver a habilidade de promover representações internas eficientes das características dos padrões de entrada. Os algoritmos de aprendizagem não-supervisionada podem ser hebbianos, quando mais de uma saída é ativada para cada entrada, ou competitivos, quando apenas uma saída é acionada para cada entrada.

As redes com treinamento supervisionado são geralmente usadas como classificadores em sistemas de reconhecimento de padrões. Por outro lado, redes com treinamento não-supervisionado são geralmente utilizadas em análise de agrupamento ou como quantizadores vetoriais.

A seguir serão abordados alguns algoritmos não-supervisionados aplicados ao projeto de dicionários (projeto de quantizadores vetoriais). Serão apresentados o algoritmo de Kohonen e os algoritmos SOA (*self-organizing algorithm*), SSC (*synaptic space competitive*) e FS-SSC (*frequency sensitive SSC*). O primeiro representa o algoritmo clássico para treinamento de mapas auto-organizativos, constituindo a metodologia de aprendizagem não-supervisionada mais amplamente utilizada. O algoritmo SOA, apesar de inspirado na regra de treinamento de Kohonen, apresenta um paradigma de vizinhança diferente do conceito de vizinhança topológica definido por Kohonen [18]. O algoritmo

SSC, por sua vez, utiliza aprendizagem competitiva simples para treinamento dos dicionários. O Algoritmo FS-SSC é resultante da introdução do princípio de consciência de Grossberg [171] no algoritmo SSC.

3.2 Algoritmo de Kohonen

Em um mapa de Kohonen [18–20], os neurônios são posicionados nos nós de um arranjo, geralmente unidimensional ou bidimensional (grade). Mapas com maior dimensão também existem, contudo são pouco comuns. Ao longo do processo de treinamento ou aprendizagem, os neurônios são seletivamente “sintonizados” a vários padrões de entrada (vetores de treinamento). Um mapa auto-organizativo é caracterizado pela formação de um mapeamento topológico dos padrões de entrada de tal maneira que neurônios com localizações espaciais (coordenadas) próximas na grade correspondem a características estatísticas próximas nos padrões de entrada a que se referem [172–174]. O principal propósito dos mapas auto-organizativos é mapear padrões de entrada de dimensão arbitrária em um arranjo discreto e realizar esse mapeamento obedecendo um ordenamento topológico dos neurônios na grade. Nesse sentido, neurônios com posições adjacentes na grade tendem a apresentar pesos sinápticos semelhantes. Uma breve descrição do algoritmo de Kohonen é dada a seguir.

Seja $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_K] \in R^K$ um vetor de entrada (vetor de treino, padrão de entrada, padrão de treino). Seja $\mathbf{w}_i = [w_{i1} \ w_{i2} \ \cdots \ w_{iK}] \in R^K$ o vetor de pesos sinápticos do i -ésimo neurônio, localizado no i -ésimo nó da grade de Kohonen. Ao se utilizar o algoritmo de Kohonen para projeto de quantizadores vetoriais, \mathbf{w}_i corresponde ao i -ésimo vetor-código (vetor de reconstrução) e os pesos sinápticos w_{ij} , com $1 \leq i \leq N$ e $1 \leq j \leq K$, correspondem às componentes do i -ésimo vetor-código K -dimensional do dicionário de tamanho N .

Os neurônios \mathbf{w}_i , depois de inicializados, são iterativamente atualizados de acordo com a utilização de um conjunto de treino. Para cada vetor de treino \mathbf{x} , o neurônio vencedor \mathbf{w}_{i^*} (isto é, o neurônio mais “semelhante” a \mathbf{x} , segundo um critério de distorção) é determinado. É determinada uma vizinhança $\mathcal{N}_{\mathbf{w}_{i^*}} = \{\mathbf{w}_i \mid d_g(\mathbf{w}_i, \mathbf{w}_{i^*}) \leq r_g(n)\}$ em torno do nó correspondente ao neurônio vencedor, em que $r_g(n)$ denota um “raio de vizinhança”, medido no arranjo topológico de nós, e $d_g(\cdot)$ é uma distância medida no arranjo topológico de nós. Os neurônios pertencentes a $\mathcal{N}_{\mathbf{w}_{i^*}}$ são atualizados na

direção do vetor de treino \mathbf{x} , de acordo com a regra de aprendizagem

$$\Delta w_{ij}(n+1) = \eta(n) \cdot \mathcal{O}_i(i^*, n) \cdot (x_j - w_{ij}(n)), \quad (3.1)$$

em que n denota o passo do algoritmo, Δw_{ij} denota a modificação introduzida na j -ésima componente (peso sináptico) do neurônio w_i , η é a taxa de aprendizagem ou ganho de adaptação, com $0 < \eta < 1$, x_j representa j -ésima componente do vetor de treino \mathbf{x} e w_{ij} denota a j -ésima componente do neurônio w_i , $1 \leq i \leq N$ e $1 \leq j \leq K$. A cada passo n , a função $\mathcal{O}_i(i^*, n)$ é usada para alterar a taxa de aprendizagem do i -ésimo vetor-código (neurônio). Trata-se de uma função da “distância física” do nó (na grade ou mapa de Kohonen), correspondente ao i -ésimo neurônio, ao nó i^* , correspondente ao neurônio vencedor [175]. Tipicamente, $\mathcal{O}_i(i^*, n)$ é não-nula para nós próximos a i^* na grade, sendo nula para nós fora dessa vizinhança.

Assim, dado o vetor de pesos sinápticos $w_i(n)$ associado ao i -ésimo nó no passo n , a versão atualizada $w_i(n+1)$ no passo $n+1$ é definida como

$$w_i(n+1) = w_i(n) + \eta(n) \cdot \mathcal{O}_i(i^*, n) \cdot (\mathbf{x} - w_i(n)). \quad (3.2)$$

Na regra de aprendizagem de Kohonen, as funções taxa de aprendizagem e raio de vizinhança decrescem com o passo n .

Podem ser identificadas duas fases no processo de aprendizagem que define o algoritmo não-supervisionado de Kohonen: fase de auto-organização ou ordenamento e fase de convergência. Na primeira, é obtido um ordenamento topológico dos neurônios na grade de nós; para tanto, a função raio de vizinhança deve incluir, no início do processo de treinamento, quase todos os neurônios cujos nós (na grade) estejam (até mesmo relativamente) próximos do nó que identifica o neurônio vencedor; a função raio de vizinhança é então gradualmente reduzida de modo a incluir poucos neurônios ou, eventualmente, apenas o neurônio vencedor. A segunda fase é necessária para proporcionar uma melhor “sintonia” dos neurônios com a distribuição estatística do conjunto de treinamento. Nessa fase, tanto o raio de vizinhança como a taxa de aprendizagem são mantidos pequenos.

3.3 Algoritmo SOA

O algoritmo SOA, embora inspirado no processo de treinamento proposto por Kohonen, utiliza uma abordagem diferente para a definição da vizinhança de atualização dos

neurônios. Em virtude de os pesos sinápticos poderem ser vistos como as coordenadas dos neurônios (vetores-código, nesse contexto) no espaço R^K , a vizinhança pode ser adequadamente definida no próprio espaço de padrões R^K , como uma hiper-esfera centrada no vetor de treino \mathbf{x} . No algoritmo SOA a vizinhança é definida como

$$\mathcal{N}_{\mathbf{x}} = \{\mathbf{w}_i \mid d_e(\mathbf{x}, \mathbf{w}_i) \leq r_e(n)\}, \quad (3.3)$$

em que $r_e(n)$ denota o raio de vizinhança (precisamente, o raio de uma hiper-esfera) na n -ésima iteração e $d_e(\cdot)$ denota a distância euclidiana, medida no espaço R^K .

São definidas duas fases no algoritmo SOA. Na primeira, todos os neurônios pertencentes à vizinhança $\mathcal{N}_{\mathbf{x}}$ têm seus pesos atualizados. Na segunda, apenas o neurônio vencedor tem seus pesos ajustados.

Podem ser apontadas, portanto, algumas diferenças entre o algoritmo SOA e o algoritmo de Kohonen: no primeiro, a vizinhança é definida no próprio espaço R^K (precisamente, como uma hiper-esfera), enquanto que no segundo é definida em um arranjo topológico de nós (em geral, uma grade bidimensional, que pode ser retangular, hexagonal, etc.); no algoritmo SOA, a vizinhança é centrada no vetor de treino, enquanto que no algoritmo de Kohonen é centrada em torno do nó (em uma grade) do neurônio vencedor. Por esse motivo, em sua primeira fase, ao contrário do algoritmo de Kohonen, o algoritmo SOA não determina o neurônio vencedor. Além disso, a segunda fase do algoritmo de Kohonen contempla a possibilidade de serem atualizados os neurônios pertencentes a uma pequena vizinhança; no algoritmo SOA, por sua vez, não se utiliza vizinhança na segunda fase de treinamento: apenas o neurônio vencedor é atualizado.

As Figuras 3.2 e 3.3 mostram as diferenças entre os algoritmos SOA e de Kohonen no tocante ao paradigma de vizinhança. Na Figura 3.2, é ilustrado o caso de uma hiper-esfera de raio $r_e(n)$ definida em R^3 : as coordenadas dos vetores são identificadas por x_1 , x_2 e x_3 . Observa-se que os vetores \mathbf{w}_i e \mathbf{w}_j , pertencentes à vizinhança $\mathcal{N}_{\mathbf{x}}$, são atualizados na direção do vetor de treino \mathbf{x} . A Figura 3.3, por sua vez, mostra que no algoritmo de Kohonen a vizinhança é definida em um arranjo topológico de nós e não no espaço de padrões.

Na seção a seguir o algoritmo SOA é apresentado com maior detalhamento.

3.3.1 Descrição do Algoritmo

Após inicializar os pesos sinápticos dos neurônios (isto é, as componentes/coordenadas dos N vetores-código K -dimensionais), o algoritmo SOA consiste dos seguintes passos:

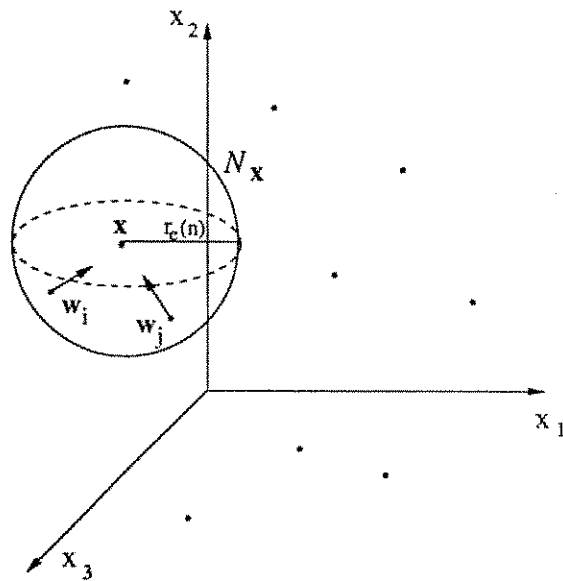


Figura 3.2: Vizinhança \mathcal{N}_x centrada no vetor de treino x . A vizinhança no algoritmo SOA é uma hiper-esfera definida no próprio espaço de padrões, neste caso o espaço R^3 .

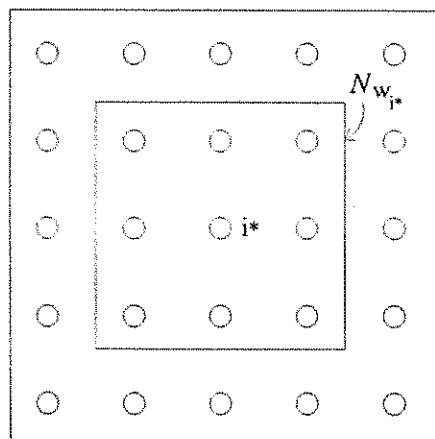


Figura 3.3: Uma vizinhança quadrada $\mathcal{N}_{w_{i^*}}$ em torno do nó i^* que corresponde ao neurônio vencedor w_{i^*} . A vizinhança do algoritmo de Kohonen é definida em um arranjo topológico (neste caso, uma grade bidimensional) de nós.

1. Apresente vetor de treino \mathbf{x} ;
2. Atualize os neurônios (vetores-código) pertencentes a uma vizinhança $\mathcal{N}_{\mathbf{x}}$ na direção de \mathbf{x} .

Os neurônios \mathbf{w}_i são atualizados de acordo com a regra de aprendizagem

$$\Delta w_{ij} = \eta(n) \cdot \mathcal{O}_{\mathbf{x}}(d_e(\mathbf{x}, \mathbf{w}_i), n) \cdot (x_j - w_{ij}), \quad (3.4)$$

em que n denota a iteração do algoritmo, Δw_{ij} denota a modificação introduzida na j -ésima componente (peso sináptico) do neurônio \mathbf{w}_i , η é a taxa de aprendizagem ou ganho de adaptação, com $0 < \eta < 1$, $\mathcal{O}_{\mathbf{x}}$ é a função que define a vizinhança em torno do vetor de treino \mathbf{x} , x_j representa j -ésima componente do vetor de treino \mathbf{x} e w_{ij} denota a j -ésima componente do neurônio \mathbf{w}_i , $1 \leq i \leq N$ e $1 \leq j \leq K$.

A cada iteração (dentro da qual os passos 1 e 2 são repetidos M vezes, em que M denota o número de vetores do conjunto de treino), a função de vizinhança $\mathcal{O}_{\mathbf{x}}$ determina os neurônios que terão seus pesos atualizados. Trata-se de uma função da distância euclidiana $d_e(\mathbf{x}, \mathbf{w}_i)$. Precisamente, $\mathcal{O}_{\mathbf{x}}$ é definida do seguinte modo:

$$\mathcal{O}_{\mathbf{x}}(d_e(\mathbf{x}, \mathbf{w}_i), n) = \begin{cases} 1, & \mathbf{w}_i \in \mathcal{N}_{\mathbf{x}} = \{\mathbf{w}_i \mid d_e(\mathbf{x}, \mathbf{w}_i) \leq r_e(n)\} \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

As funções taxa de aprendizagem e raio de vizinhança decrescem com n , conforme mostram as Figuras 3.4 e 3.5. No que diz respeito à Figura 3.5, observa-se que $r_e(n) = 0$ para $n_f \leq n \leq n_{tot}$, em que n_{tot} denota o número total de iterações (número total de passagens do conjunto de treino) do algoritmo SOA. O parâmetro n_f , portanto, define a iteração a partir da qual o raio de vizinhança é zero. A partir da iteração $n = n_f$, o algoritmo SOA transita para sua segunda fase, na qual apenas o neurônio vencedor (ou seja, o vetor-código mais próximo do vetor de treino) é atualizado. Muito embora o presente trabalho tenha utilizado a taxa de aprendizagem e o raio de vizinhança como funções linearmente decrescentes com a iteração, outras opções podem ser utilizadas, tais como a função gaussiana e a função chapéu mexicano.

É importante salientar que, no algoritmo de Kohonen, n denota o passo – cada passo corresponde a um vetor de treino. No algoritmo SOA, n denota a iteração, isto é, uma passagem completa do conjunto de vetores de treino – no algoritmo SOA, portanto, uma iteração tem como correspondência vários passos do algoritmo de Kohonen. Por essa razão, aponta-se mais uma diferença entre os algoritmos SOA e de Kohonen.

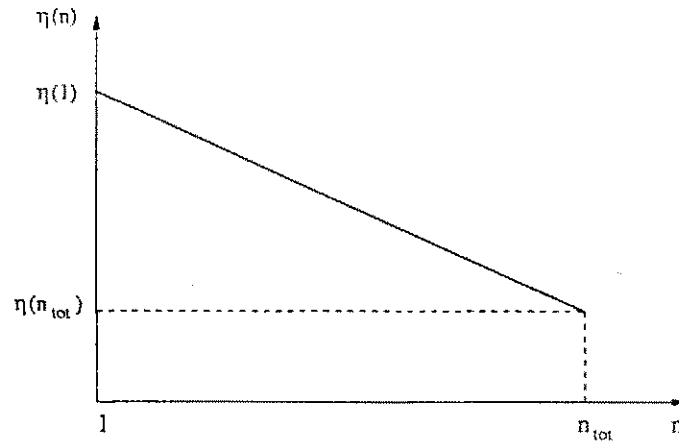


Figura 3.4: Função taxa de aprendizagem $\eta(n)$ no algoritmo SOA.

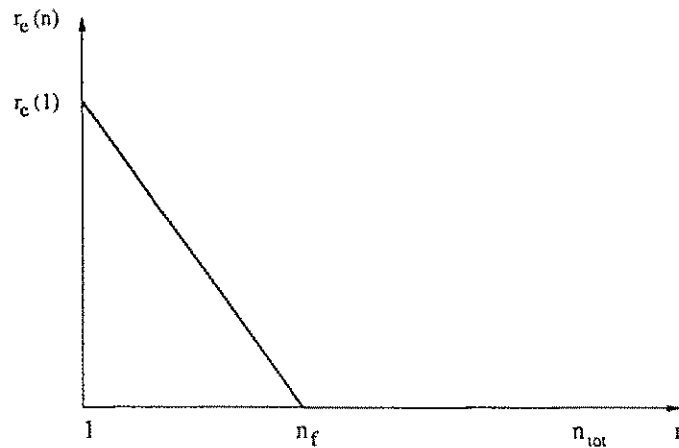


Figura 3.5: Função raio de vizinhança $r_e(n)$ no algoritmo SOA.

No primeiro, as funções taxa de aprendizagem e raio de vizinhança decrescem a cada iteração, permanecendo constantes ao longo de uma determinada iteração¹, isto é, no decurso da apresentação de todos os vetores de treino. No segundo, essas funções decrescem a cada passo do algoritmo².

O algoritmo SOA oferece 7 parâmetros ajustáveis:

1. Dimensão do dicionário (K);
2. Número de vetores-código (N);

¹Isto equivaleria a manter as funções raio de vizinhança e taxa de aprendizagem constantes em vários passos do algoritmo de Kohonen.

²O que equivaleria a fazer com que essas funções variassem (decrecessem) dentro de uma iteração do algoritmo SOA.

3. Número total de iterações (n_{tot});
4. Taxa de aprendizagem inicial ($\eta(1)$);
5. Taxa de aprendizagem final ($\eta(n_{tot})$);
6. Raio de vizinhança inicial ($r_e(1)$);
7. Número de iterações da primeira fase de treinamento (n_f).

A seguir é apresentada uma abordagem do paradigma de vizinhança do algoritmo SOA, que se distingue sobremaneira do paradigma de vizinhança do algoritmo de Kohonen.

3.3.2 O Paradigma de Vizinhança

O paradigma de vizinhança do algoritmo SOA procura ser mais adequado que o paradigma de vizinhança do algoritmo de Kohonen. A Figura 3.6 ilustra uma situação em que a vizinhança SOA seria mais adequada que a vizinhança de Kohonen para o propósito de projeto de dicionários. Na Figura 3.6(a), na grade de nós bidimensional hexagonal 4×7 sobre a qual é definida a regra de atualização dos pesos, o nó correspondente ao neurônio vencedor é denotado por c e uma vizinhança hexagonal de raio $r_g = 1$ é definida em torno de c . Assim, na regra de atualização dos pesos, o algoritmo de Kohonen ajustaria apenas os pesos do neurônio vencedor, correspondente ao nó c , e dos neurônios 1, 2, 3, 4, 5 e 6, cujos nós pertencem à vizinhança hexagonal. Na Figura 3.6(b), que representa o vetor de treino \mathbf{x} e os vetores c , 4 e 7 da Figura 3.6(a) no espaço de padrões (espaço R^K), observa-se que uma abordagem mais justa consistiria em atualizar o vetor 7 e não o vetor 4, tendo em vista que o vetor 7 é mais semelhante ao vetor de treino \mathbf{x} (segundo um critério de distância medida no espaço R^K) que o vetor 4: na Figura as distâncias dos vetores c , 4 e 7 para o vetor de treino \mathbf{x} são denotadas, respectivamente, por d_c , d_4 e d_7 , com $d_c < d_7 < d_4$. Utilizando a definição de vizinhança topológica do algoritmo de Kohonen, o neurônio (vetor) 4 seria atualizado (isto é, teria seus pesos sinápticos ajustados de acordo com a Equação (3.1)) na direção de \mathbf{x} pelo simples fato de seu correspondente nó pertencer à vizinhança hexagonal definida na grade bidimensional de nós. No algoritmo de Kohonen, o vetor 7, apesar de mais semelhante (segundo um critério de distância definida em R^K , como por exemplo a distância euclidiana) a \mathbf{x} que o vetor 4, não seria atualizado por não corresponder a

um nó pertencente à vizinhança hexagonal em torno do nó que identifica o neurônio (vetor) vencedor c . Para contornar esse problema, uma boa alternativa consistiria em definir a vizinhança no próprio espaço de padrões (espaço de interesse, espaço em que a quantização vetorial é levada a efeito – espaço R^3 , no caso da Figura 3.6(b)) como uma hiper-esfera centrada no vetor de treino x . Este paradigma de vizinhança, utilizado pelo algoritmo SOA e consubstanciado na Equação (3.5), procura apresentar-se como uma opção adequada para projeto de dicionários.

É importante mencionar que em virtude de a QV requerer apenas a representação dos vetores de entrada pelos correspondentes vetores-código, relações de vizinhança topológica não são importantes para a qualidade do sinal reconstruído. Assim, por não lidar com um paradigma de vizinhança topológica de nós, o algoritmo SOA se ocupa em procurar produzir uma influência mais adequada (em comparação ao algoritmo de Kohonen) dos vetores de treino nos vetores-código do dicionário.

O seguinte aspecto deve ser destacado: a suscetibilidade do algoritmo de Kohonen a um cenário semelhante ao apresentado na Figura 3.6 pode tender a diminuir ao longo do treinamento, devido à propriedade de auto-organização.

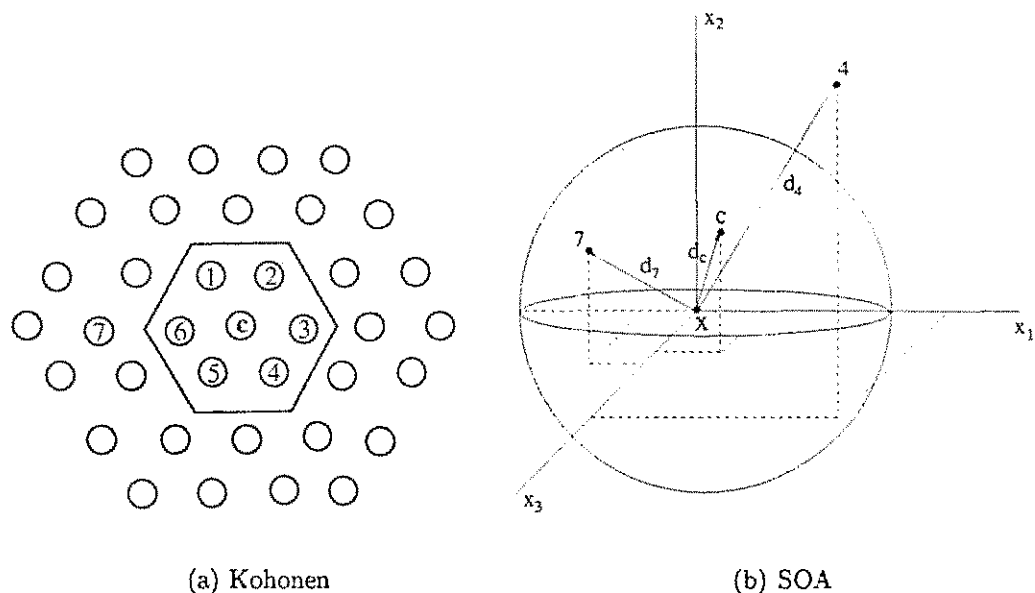


Figura 3.6: Definição de vizinhança no algoritmo de Kohonen e no algoritmo SOA.

A definição de vizinhança do algoritmo SOA representa uma evolução da definição de vizinhança proposta por França e Aguiar Neto em [22, 23]. No algoritmo SSN (*sy-*

naptic space neighborhood) [22, 23], a definição de vizinhança consiste em centrar a hiper-esfera em torno do neurônio (vetor) vencedor c , conforme ilustra a Figura 3.7, precisamente a hiper-esfera mais à esquerda. Nota-se que a definição de vizinhança (hiper-esfera centrada no vetor de treino) utilizada no algoritmo SOA trata-se de uma abordagem mais adequada. Observa-se que a vizinhança centrada no neurônio vencedor c (vetor-código mais próximo de x em termos de distância medida no espaço R^K) não inclui o neurônio 1. No entanto, como a distância d_1 é menor que a distância d_2 , decorre que o vetor (neurônio) 1 é mais semelhante a x que o vetor 2. A atualização dos pesos, portanto, deve ser efetuada no neurônio 1 (o que é contemplado pelo algoritmo SOA) e não no neurônio 2 (este, por sua vez, é “inadequadamente” contemplado pelo algoritmo SSN).

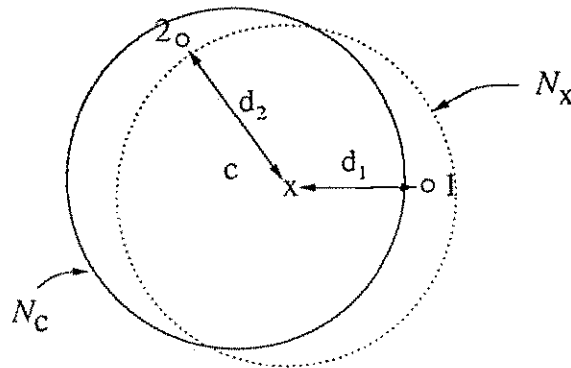


Figura 3.7: Definição de vizinhança no algoritmo de SSN e no algoritmo SOA. A vizinhança (hiper-esfera) centrada no vetor de treino é denotada por \mathcal{N}_x , enquanto que a hiper-esfera centrada no neurônio vencedor é denotada por \mathcal{N}_c .

3.3.3 Considerações Adicionais

O algoritmo SOA se distingue do algoritmo SSN-TV (*Kohonen's algorithm with spherical neighborhood around training vector*) [25, 26, 176], definido por França e Aguiar Neto, por apresentar um parâmetro a mais: a taxa de aprendizagem final³. A introdução desse parâmetro adicional teve como objetivo aumentar a flexibilidade (o

³Nesse contexto, cumpre registrar que, muito embora não explicitamente relatado em [88], [177] e [178], os resultados apresentados foram obtidos com a utilização do parâmetro taxa de aprendizagem final. Assim, rigorosamente, o algoritmo utilizado em [88], [177] e [178] não se trata do SSN-TV mas sim do SOA.

número de graus de liberdade) do algoritmo, de tal forma que o ajuste adequado de um maior número de parâmetros possa contribuir para a obtenção de bons dicionários. Além disso, foi motivada pela constatação de que o algoritmo SSN-TV⁴ utiliza taxa de aprendizagem zero na última iteração (conforme mostra a Figura 9 de [25]) – com isso, o algoritmo SSN-TV simplesmente “desperdiça” a última iteração (passagem) do conjunto de treino, uma vez que não há aprendizado, isto é, não há atualização de pesos para o caso de taxa de aprendizagem igual a zero.

É oportuno mencionar que as simulações relatadas em [25], [26], [179] e [176] foram realizadas tendo como objetivo a aplicação do algoritmo SSN-TV ao projeto de dicionários destinados à quantização vetorial de forma de onda de voz. O presente trabalho apresenta algumas contribuições com respeito aos trabalhos de França e Aguiar Neto, tais como: avaliação da influência do dicionário inicial no desempenho do algoritmo SOA; avaliação da qualidade inerente dos dicionários SOA sob o ponto de vista da complexidade computacional envolvida na fase de codificação da quantização vetorial; avaliação de desempenho do algoritmo SOA quando aplicado ao projeto de dicionários destinados à codificação de imagens e ao reconhecimento de locutor.

Finalmente, cumpre registrar que o algoritmo SOA é denotado por KMTAU (Kohonen modificado com taxa de aprendizagem uniforme) em [86], [87] [156] e [180], por MKOH (*modified Kohonen's algorithm*) em [181] e por KMVVT (Kohonen modificado com vizinhança centrada no vetor de treino) em [149], [182], [183] e [184].

3.4 Algoritmo SSC

No algoritmo SSC, apenas o vencedor tem seus pesos atualizados. Após uma inicialização do dicionário (isto é, dos pesos sinápticos dos neurônios, ou seja, das componentes ou coordenadas dos N vetores-código K -dimensionais), o algoritmo SSC, a cada apresentação de vetor de treino, atualiza o vencedor (vetor-código mais próximo do vetor de treino) correspondente.

Seja n_{SSC} o número total de iterações (número total de passagens do conjunto de treino) do algoritmo SSC e M o número total de vetores de treino. O algoritmo SSC é descrito a seguir.

⁴O algoritmo SSN-TV é denotado por VES-VT (algoritmo de Kohonen com vizinhança no vetor de treino) em [179]. Com a concepção do algoritmo SSN-TV, o algoritmo SSN passou a ser denotado por SSN-SV em [25, 26], em que o acrônimo SV é uma alusão a *synaptic winning vector*.

Algoritmo SSC:

Para $1 \leq n \leq n_{\text{SSC}}$

Para $1 \leq m \leq M$

Determine o vencedor $w_{i^*}(n, m)$:

$$i^* = \arg \min_i d[\mathbf{x}(m), w_i(n, m)].$$

Atualize o vencedor de acordo com

$$\tilde{w}_{i^*j}(n, m) = w_{i^*j}(n, m) + \Delta w_{i^*j}(n, m), \quad (3.6)$$

em que

$$\Delta w_{i^*j}(n, m) = \eta(n)[x_j(m) - w_{i^*j}(n, m)]. \quad (3.7)$$

Na descrição acima, $\mathbf{x}(m)$ é o m -ésimo vetor do conjunto de treino⁵, enquanto $w_i(n, m)$ e $w_{i^*}(n, m)$ denotam, respectivamente, o i -ésimo vetor-código e o vencedor quando da apresentação do m -ésimo vetor de treino na n -ésima iteração. Por sua vez,

$$d[\mathbf{x}(m), w_i(n, m)] = \sum_{j=1}^K [x_j(m) - w_{ij}(n, m)]^2 \quad (3.8)$$

denota a distância euclidiana entre os vetores $\mathbf{x}(m)$ e $w_i(n, m)$, em que $x_j(m)$ é a j -ésima componente do vetor $\mathbf{x}(m)$ e $w_{ij}(n, m)$ é a j -ésima componente do vetor $w_i(n, m)$. Na expressão que descreve a atualização do vencedor, Δw_{i^*j} é a modificação introduzida na j -ésima componente do vencedor, $\eta(n)$ é a taxa de aprendizagem ou ganho de adaptação na n -ésima iteração (com $0 < \eta(n) < 1$), w_{i^*j} é a j -ésima componente do vencedor e \tilde{w}_{i^*j} é a versão atualizada da j -ésima componente do vencedor⁶.

No algoritmo SSC, a exemplo do que ocorre no algoritmo SOA, a taxa de aprendizagem decresce linearmente com a iteração n , mantendo-se constante ao longo de toda iteração, isto é, durante cada passagem completa dos M vetores de treino. É expressa por

$$\eta(n) = \eta(1) + (n - 1) \frac{\eta(n_{\text{SSC}}) - \eta(1)}{n_{\text{SSC}} - 1}, \quad (3.9)$$

⁵É importante observar que, por questões de conveniência de notação, utilizou-se $\mathbf{x}(m)$ na descrição do algoritmo SSC e \mathbf{x}_m na descrição do algoritmo LBG.

⁶Note que se omitiram n e m de $\Delta w_{i^*j}(n, n)$, de $w_{i^*j}(n, n)$ e de $\tilde{w}_{i^*j}(n, m)$ para simplificar a notação.

em que $\eta(1)$ e $\eta(n_{SSC})$ denotam a taxa de aprendizagem inicial e a taxa de aprendizagem final, respectivamente.

O algoritmo SSC oferece 5 parâmetros ajustáveis:

1. Dimensão do dicionário (K);
2. Número de vetores-código (N);
3. Número total de iterações (n_{SSC});
4. Taxa de aprendizagem inicial ($\eta(1)$);
5. Taxa de aprendizagem final ($\eta(n_{SSC})$).

Nesse sentido, apesar da manutenção do acrônimo SSC, o algoritmo apresentado nesta seção e com o qual foram obtidos diversos resultados apresentados no presente trabalho difere do algoritmo SSC apresentado por França e Aguiar Neto em [24], [25] e [26], o qual apresenta 4 parâmetros ajustáveis. A introdução do parâmetro *taxa de aprendizagem final* na versão original do algoritmo SSC teve como objetivo aumentar a flexibilidade (o número de graus de liberdade) do algoritmo, de tal forma que o ajuste adequado de parâmetros possa contribuir para a consecução de bons dicionários. Também foi motivada pela constatação de que a versão original de SSC apresenta o mesmo problema do algoritmo SSN-TV com respeito ao desperdício da última iteração do conjunto de treino, isto é, utilização de taxa de aprendizagem final igual a zero.

É oportuno mencionar que as simulações relatadas em [24], [25], [26], [21] e [179]⁷ foram realizadas tendo como objetivo a aplicação do algoritmo SSC no projeto de dicionários destinados à quantização vetorial de forma de onda de voz. O presente trabalho apresenta algumas contribuições com respeito aos trabalhos de França e Aguiar Neto, tais como: avaliação da influência do dicionário inicial no desempenho do algoritmo SSC; avaliação da qualidade inerente dos dicionários SSC sob o ponto de vista da complexidade computacional envolvida na fase de codificação da quantização vetorial; avaliação de desempenho do algoritmo SSC quando aplicado ao projeto de dicionários destinados à codificação de imagens, à codificação dos parâmetros LSF e ao reconhecimento de locutor.

A seção a seguir apresenta uma descrição do algoritmo FS-SSC, resultante da introdução do princípio de *consciência* de Grossberg [171] no algoritmo SSC.

⁷No relatório técnico [21], bem como em [179], o algoritmo SSC é registrado como CES (competitivo no espaço sináptico).

3.5 Algoritmo FS-SSC

O principal problema da aprendizagem competitiva simples é que alguns neurônios (vetores-código, no contexto de QV) podem ter pouca ou nenhuma chance de ganhar a competição, o que pode resultar em um dicionário que contém vetores-código que não tenham sido suficientemente treinados (vetores-código subutilizados), podendo resultar, em casos extremos, em um dicionário que contenha alguns vetores-código não treinados (o que seria um problema equivalente ao da existência de células de Voronoi vazias no algoritmo LBG).

Uma das maneiras de contornar esse problema foi apresentada por Krishnamurthy *et al.* em [185]. A técnica proposta, denominada FSCL (*frequency sensitive competitive learning*), pode ser utilizada para melhorar o desempenho de redes neurais competitivas. Na técnica FSCL a distorção é também uma função da frequência com que os vetores-código ganham a competição (número de vezes em que os vetores-código são eleitos vencedores). A abordagem FSCL procura treinar igualmente todos os vetores-código, isto é, procura fazer com que todos os vetores-código sejam treinados (tenham sua componentes ajustadas) aproximadamente o mesmo número de vezes. De acordo com [185], o algoritmo FSCL constitui uma implementação do princípio de consciência de Grossberg [171].

No presente trabalho, a idéia proposta na técnica FSCL é introduzida no contexto do algoritmo SSC, dando origem ao algoritmo FS-SSC (*frequency sensitive SSC*).

No algoritmo FS-SSC a frequência com que cada vetor-código ganha a competição é monitorada. Esta informação é usada durante o treinamento para assegurar que todos os vetores-código tenham aproximadamente a mesma oportunidade de serem atualizados. Precisamente, o algoritmo FS-SSC utiliza uma medida de distorção (distância) modificada, apresentada em [185], que incorpora a frequência com que cada vetor-código é escolhido vencedor.

Seja $d[\mathbf{x}(m), \mathbf{w}_i(n, m)]$ a medida de distorção utilizada pelo algoritmo SSC. A medida de distorção modificada utilizada pelo algoritmo FS-SSC é dada por

$$\hat{d}[\mathbf{x}(m), \mathbf{w}_i(n, m)] = f_i \times d[\mathbf{x}(m), \mathbf{w}_i(n, m)], \quad (3.10)$$

em que f_i denota o número de vezes em que o i -ésimo vetor-código foi até então

escolhido como vencedor. A Equação (3.10), portanto, pode ser expressa como

$$\hat{d}[\mathbf{x}(m), \mathbf{w}_i(n, m)] = f_i \times \sum_{j=1}^K [x_j(m) - w_{ij}(n, m)]^2. \quad (3.11)$$

A Equação (3.11) mostra que se um vetor-código é freqüentemente escolhido vencedor, sua distorção \hat{d} aumentará. Conseqüentemente, sua chance de vencer a(s) próxima(s) competição(ões) diminuirá, dando, portanto, aos outros vetores-código com pequeno valor de f_i a oportunidade de serem vencedores (de serem atualizados) na(s) próxima(s) apresentação(ões) de vetores de treino. Essa natureza adaptativa do algoritmo FS-SSC possibilita que os vetores-código tendam a ser atualizados aproximadamente o mesmo número de vezes durante o treinamento.

3.6 Considerações Finais

Existe uma diferença marcante entre os algoritmos não-supervisionados (SOA, SSC e FS-SSC) apresentados neste trabalho e o algoritmo LBG: nos três primeiros, o dicionário é atualizado a cada apresentação de vetor de treino, enquanto que no algoritmo LBG o dicionário é atualizado ao final de cada iteração (isto é, ao final de cada passagem completa do conjunto de treino).

Além disso, nos algoritmos SOA, SSC e FS-SSC, o número de iterações é especificado *a priori*. No algoritmo LBG, o número de iterações depende do dicionário inicial e do limiar de distorção utilizado como critério de parada do algoritmo. No algoritmo LBG, portanto, o número de iterações não é estabelecido *a priori*.

Um aspecto deve ser destacado: nos algoritmos SOA, SSC e FS-SSC, a taxa de aprendizagem decresce a cada iteração, permanecendo constante ao longo de toda uma iteração (durante cada passagem completa de vetores de treino). Esse aspecto diferencia os algoritmos SOA, SSC e FS-SSC de outros algoritmos não supervisionados [19,185,186], nos quais a taxa de aprendizagem varia a cada apresentação de vetor de treino.

Capítulo 4

Avaliação dos Dicionários SOA, SSC e FS-SSC

Este capítulo apresenta diversos resultados obtidos com os algoritmos SOA, SSC e FS-SSC. São apresentadas avaliações concernentes ao projeto de dicionários destinados à codificação de voz e imagem e ao reconhecimento de locutor.

4.1 Quantização Vetorial de Forma de Onda de Voz – Domínio Temporal

Os resultados apresentados nesta seção foram obtidos com dicionários projetados utilizando um conjunto de treino curto, constituído de 7120 amostras (0,89 s), correspondente à palavra “*aplausos*”. Optou-se pela escolha de um conjunto de treino curto depois de se observar que os resultados foram compatíveis com os obtidos ao serem utilizados longos conjuntos de treino. A utilização de conjuntos de treino curtos implica projeto de dicionários com maior eficiência com relação ao tempo de processamento. A escolha da palavra “*aplausos*” foi motivada pelo fato de que ela apresenta uma boa variedade de sons: explosivos (fonema $|p|$), sonoros (fonema $|a|$) e fricativos surdos (fonema $|s|$) [180]. Para avaliação dos dicionários, utilizou-se um conjunto de teste relativamente longo, constituído de 29120 amostras (3,64 s), correspondente ao conjunto de sentenças “*O sol ilumina a fachada de tarde. Trabalhou mais do que podia*”.

A aquisição (resolução 8,0 bit/amostra e taxa de amostragem 8 kHz) dos sinais de voz utilizados nas simulações foi realizada usando uma estação de trabalho Sun[®],

instalada com utilitários de processamento de áudio. A qualidade dos sinais de voz reconstruídos foi avaliada utilizando-se como medida de desempenho a relação sinal-ruído segmental (SNRseg).

4.1.1 Sensibilidade ao Dicionário Inicial

A sensibilidade dos algoritmos SOA, SSC e LBG ao dicionário inicial pode ser avaliada observando-se as Tabelas 4.1, 4.2 e 4.3. São apresentados resultados de SNRseg do sinal de voz reconstruído com dicionários projetados utilizando três dicionários iniciais diferentes e o correspondente valor de coeficiente de variação, um parâmetro bastante utilizado para indicar o grau de homogeneidade de dados. O coeficiente de variação (CV), em valor percentual, é expresso como [187, 188]

$$CV = \frac{\sigma_X}{\bar{X}} 100\%, \quad (4.1)$$

em que \bar{X} e σ_X denotam, respectivamente, a média e o desvio-padrão de um conjunto de observações (X_1, \dots, X_n) , expressos por

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i; \quad (4.2)$$

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}. \quad (4.3)$$

Para cada par de valores K e N , a maior sensibilidade corresponde ao maior valor de CV. Por outro lado, a menor sensibilidade corresponde ao menor valor de CV. Assim, por exemplo, para $K = 4$ e $N = 64$, a menor sensibilidade ao dicionário inicial está associada ao algoritmo SOA, por apresentar o menor valor de CV (igual a 0,51) dentre os três algoritmos considerados, enquanto que a maior sensibilidade está associada ao algoritmo SSC em virtude de esse algoritmo apresentar o maior CV (igual a 3,66) dentre os algoritmos avaliados.

As Tabelas 4.1, 4.2 e 4.3 mostram que o algoritmo SOA apresenta-se como a alternativa menos sensível ao dicionário inicial. O algoritmo SSC, por sua vez, é o que apresenta maior sensibilidade ao dicionário inicial. De fato, dentre as 15 combinações de K e N avaliadas, o algoritmo SOA apresenta uma menor sensibilidade (menor CV) em 8 combinações, enquanto que o algoritmo SSC apresenta maior sensibilidade em 7 combinações.

Tabela 4.1: Sensibilidade do algoritmo SOA a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.

K	N	SNRseg (dB)			CV (%)
		DI	DII	DIII	
2	8	6,33	6,35	6,35	0,18
2	16	8,70	8,64	8,73	0,52
2	32	11,12	10,73	10,94	1,78
2	64	13,70	13,83	13,59	0,87
2	128	15,93	16,00	16,05	0,37
4	8	4,42	4,02	4,07	5,22
4	16	6,27	6,63	6,48	2,79
4	32	7,81	8,05	7,37	4,45
4	64	9,82	9,78	9,72	0,51
4	128	11,13	11,20	11,30	0,76
8	8	2,90	3,12	3,12	4,16
8	16	4,30	4,32	2,46	28,92
8	32	5,54	5,55	4,09	16,60
8	64	6,82	6,58	6,19	4,86
8	128	7,83	7,32	7,67	3,42

Tabela 4.2: Sensibilidade do algoritmo SSC a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.

K	N	SNRseg (dB)			CV (%)
		DI	DII	DIII	
2	8	6,01	6,34	6,23	2,71
2	16	8,60	8,97	8,36	3,55
2	32	10,89	11,47	11,07	2,66
2	64	13,59	13,60	13,61	0,07
2	128	15,07	15,19	15,20	0,47
4	8	3,41	4,21	3,40	12,65
4	16	5,98	6,36	6,58	4,81
4	32	7,39	8,02	7,77	4,10
4	64	9,09	9,66	9,72	3,66
4	128	9,92	11,05	10,46	5,39
8	8	2,39	2,75	2,16	12,22
8	16	4,12	4,31	4,05	3,23
8	32	5,13	5,69	5,51	5,25
8	64	5,63	6,60	6,45	8,38
8	128	5,97	7,43	7,45	12,21

Tabela 4.3: Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de SNRseg (dB). São apresentados os valores de CV referentes aos resultados de SNRseg obtidos.

K	N	SNRseg (dB)			CV (%)
		DI	DII	DIII	
2	8	4,28	4,10	4,11	2,42
2	16	7,12	7,34	6,57	5,65
2	32	10,19	10,15	10,15	0,22
2	64	13,39	13,30	13,25	0,53
2	128	15,76	15,99	15,98	0,81
4	8	2,78	2,76	2,92	3,09
4	16	4,83	4,46	4,16	7,48
4	32	7,27	7,01	7,31	2,26
4	64	9,49	10,01	9,92	2,83
4	128	10,68	11,26	11,31	3,15
8	8	1,33	0,91	0,86	24,98
8	16	2,50	2,56	2,94	8,94
8	32	4,63	4,79	4,65	1,85
8	64	6,51	6,83	6,81	2,66
8	128	6,73	7,75	7,63	7,56

4.1.2 SNRseg versus Taxa de Codificação

As Figuras 4.1, 4.2 e 4.3 apresentam o desempenho dos algoritmos SOA, SSC e LBG em termos de relação sinal-ruído segmental (SNRseg) do sinal de voz reconstruído, para diversas taxas de um quantizador vetorial com dimensão 2, 4 e 8, respectivamente.

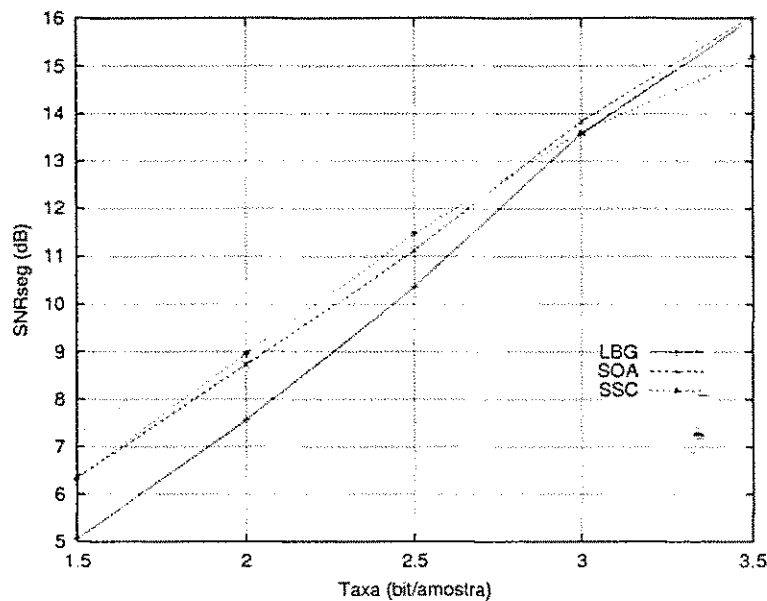


Figura 4.1: Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 2$.

Conforme mostra a Figura 4.1, fixada a dimensão $K = 2$, os algoritmos SOA e SSC apresentam desempenho superior ao apresentado pelo algoritmo LBG para as taxas de codificação na faixa de 1,5 bit/amostra a 3,0 bit/amostra. Para a taxa de 1,5 bit/amostra, o algoritmo SOA supera o algoritmo em LBG em 1,3 dB; para a taxa de 2,0 bit/amostra, o algoritmo SSC supera o algoritmo LBG em 1,4 dB. Além disso, observa-se que os algoritmos SOA e SSC apresentam um desempenho semelhante para as taxas de codificação inferiores a 3,0 bit/amostra. É possível observar, entretanto, que para a taxa de 3,5 bit/amostra, o algoritmo SSC apresenta um desempenho inferior, em cerca de 0,8 dB, ao apresentado pelos algoritmos SOA e LBG.

Em se tratando de quantização vetorial com dimensão $K = 4$, a Figura 4.2 mostra que, para a faixa de taxas compreendida entre 0,75 bit/amostra e 1,25 bit/amostra, os algoritmos SOA e SSC apresentam um desempenho superior ao apresentado pelo algoritmo LBG, levando a maiores valores de SNRseg do sinal reconstruído: para as

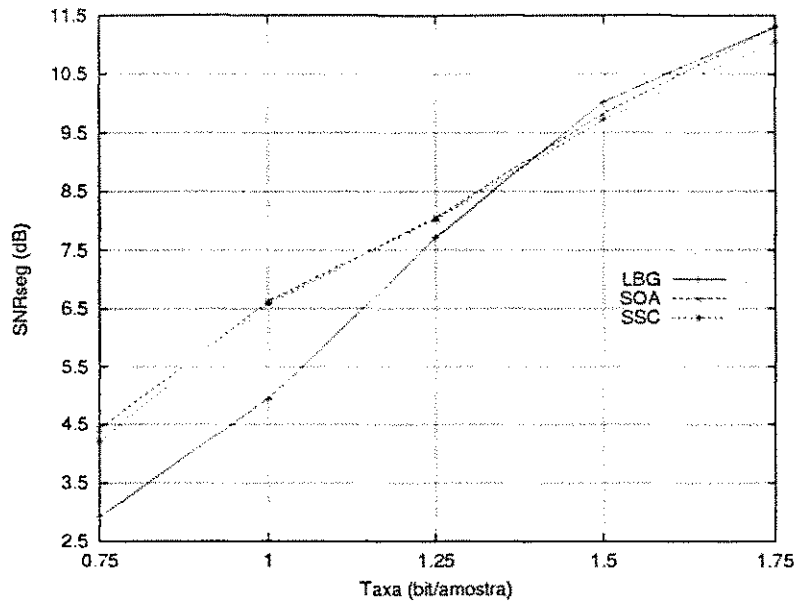


Figura 4.2: Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 4$.

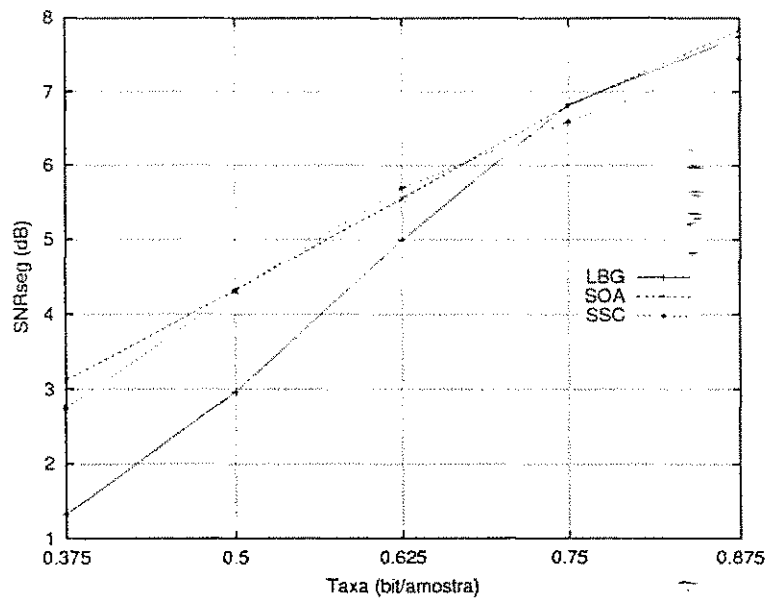


Figura 4.3: Desempenho dos algoritmos SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 8$.

taxas de 0,75 bit/amostra e 1,0 bit/amostra, por exemplo, o algoritmo SOA supera o algoritmo LBG em cerca de 1,5 dB e 1,7 dB, respectivamente. Para taxas superiores a 1,25 bit/amostra, o desempenho dos três algoritmos considerados é bastante próximo.

No tocante à quantização vetorial de forma de onda de voz (domínio temporal) com dimensão $K = 8$, a Figura 4.3 mostra que os dicionários projetados com os algoritmos SOA e SSC levam a sinais reconstruídos com maiores valores de SNRseg que os obtidos com dicionários LBG, para taxas inferiores a 0,7 bit/amostra. A superioridade dos algoritmos SOA e SSC sobre o algoritmo LBG acentua-se à medida que a taxa diminui: o algoritmo SOA supera o algoritmo LBG em cerca de 1,8 dB e 1,3 dB para as taxas de 0,375 bit/amostra e 0,5 bit/amostra, respectivamente. A Figura 4.3 também mostra que os algoritmos SOA e SSC apresentam desempenho semelhante, notadamente nas taxas compreendidas entre 0,5 bit/amostra e 0,75 bit/amostra.

4.2 Quantização Vetorial de Forma de Onda de Voz – Domínio *Wavelet*

Nesta seção é apresentada uma avaliação da importância das componentes de aproximação e de detalhe de uma decomposição *wavelet* multiresolucional na qualidade do sinal de voz reconstruído. São apresentados, também, resultados de codificação. O desempenho do sistema de codificação de forma de onda de voz utilizando DWT e QV é comparado ao desempenho obtido com uso de QV simples (isto é, QV convencional, sem uso de DWT). Para avaliação objetiva da qualidade do sinal de voz reconstruído são utilizadas a relação sinal-ruído total (SNRtot) e a relação sinal-ruído segmental (SNRseg). No tocante à avaliação subjetiva de qualidade, são relatados resultados de testes de preferência que contaram com a participação de 20 avaliadores.

4.2.1 A Importância das Componentes de Aproximação e de Detalhe

Em todas as simulações realizadas, utilizou-se, na DWT, a família Daubechies 6. Os resultados apresentados nesta seção foram obtidos com o uso de uma decomposição *wavelet* multiresolucional de 3 níveis.

O primeiro conjunto de simulações realizadas com sinais de voz consistiu em avaliar a importância das componentes de aproximação e de detalhe, isto é, dos sinais $a_3(n)$,

$d_3(n)$, $d_2(n)$ e $d_1(n)$, na qualidade do sinal de voz reconstruído. Para tanto, foi avaliada a qualidade do sinal obtido mediante aplicação da IDWT com a exclusão de uma ou mais componentes. A Tabela 4.4 mostra que $a_3(n)$ corresponde à componente mais importante para a qualidade do sinal reconstruído. De fato, o menor valor de SNRseg da Tabela 4.4 corresponde ao sinal reconstruído após a exclusão de $a_3(n)$. Por outro lado, $d_1(n)$ corresponde ao conjunto de coeficientes *wavelet* menos importante para a qualidade do sinal reconstruído – observa-se, na Tabela 4.4, que o maior valor de SNRseg diz respeito à exclusão de $d_1(n)$. Dentre as componentes de detalhe, é possível verificar que $d_3(n)$ é a componente mais importante para a qualidade do sinal reconstruído.

Vale salientar que testes subjetivos informais realizados com diversos sinais reconstruídos indicaram que a exclusão de $d_1(n)$ implica queda praticamente imperceptível de qualidade do sinal reconstruído. É oportuno ressaltar que $d_1(n)$ contém 50% dos coeficientes *wavelet* obtidos com a DWT, de modo que a taxa de codificação de $d_1(n)$ desempenha um papel importante na definição da taxa de codificação total do sistema de codificação de voz baseado em DWT e QV considerado no presente trabalho.

Tabela 4.4: Importância das componentes de aproximação e de detalhe (de uma decomposição *wavelet* multiresolucional de 3 níveis) na qualidade do sinal de voz reconstruído. *Wavelet* utilizada: Daubechies 6 (db6).

Tipo de reconstrução	SNRseg (dB)
Excluindo-se $a_3(n)$	1,95
Excluindo-se $d_3(n)$	10,21
Excluindo-se $d_2(n)$	16,54
Excluindo-se $d_1(n)$	20,67
Excluindo-se $d_1(n)$ e $d_2(n)$	14,20
Excluindo-se $d_1(n)$ e $d_3(n)$	9,20
Excluindo-se $d_2(n)$ e $d_3(n)$	8,85
Excluindo-se $d_1(n)$, $d_2(n)$ e $d_3(n)$	8,13

4.2.2 Resultados de Codificação

Uma visão geral do sistema de codificação de voz baseado em wavelets e quantização vetorial é apresentada na Figura 4.4. Após aplicação da transformada *wavelet* discreta (DWT), ou seja, de uma decomposição *wavelet* multiresolucional (\mathcal{L} níveis de decomposição), as componentes de detalhe, $d_i(n)$, com $1 \leq i \leq \mathcal{L}$, e a componente de aproximação, $a_{\mathcal{L}}(n)$, são quantizadas mediante utilização de um banco de dicionários multiresolucionais. Em outras palavras, em cada nível de resolução, cada componente da decomposição *wavelet* é quantizada com uso de um dicionário específico. O sinal reconstruído é obtido aplicando-se a transformada *wavelet* discreta inversa (IDWT) nas componentes de aproximação e de detalhe reconstruídas (quantizadas vetorialmente).

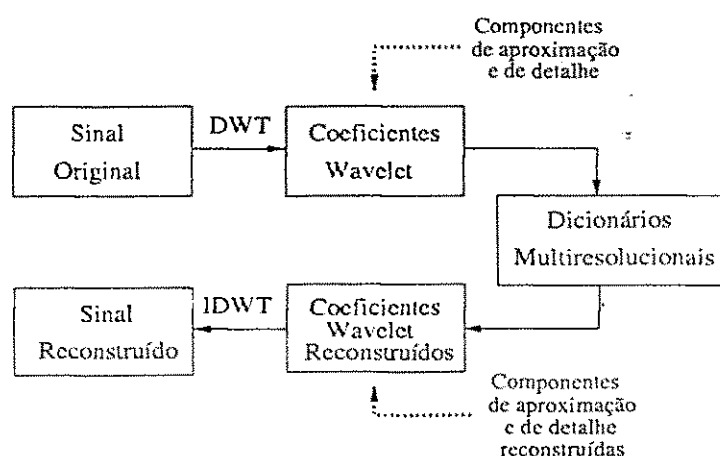


Figura 4.4: Diagrama de blocos do sistema de codificação de voz baseado em *wavelets* e quantização vetorial.

Considere uma decomposição *wavelet* multiresolucional de um sinal $x(n)$, constituído de M amostras. O número de coeficientes *wavelet* em $d_i(n)$, com $1 \leq i \leq \mathcal{L}$, é igual a $\frac{M}{2^i}$, e o número de coeficientes *wavelet* em $a_{\mathcal{L}}(n)$ é igual a $\frac{M}{2^{\mathcal{L}}}$. Seja R_{d_i} a taxa (em bit/amostra, ou mais precisamente em bit/coeficiente) do quantizador vetorial alocada para o sinal $d_i(n)$ e seja $R_{a_{\mathcal{L}}}$ a taxa (em bit/amostra) do quantizador vetorial alocada para o sinal $a_{\mathcal{L}}(n)$. A taxa de codificação total R_T (em bit/amostra) do sistema de codificação de voz utilizando DWT (com \mathcal{L} níveis de decomposição) e QV é dada por [189]

$$R_T = \frac{1}{M} \left(\frac{M}{2^{\mathcal{L}}} \cdot R_{a_{\mathcal{L}}} + \sum_{i=1}^{\mathcal{L}} \frac{M}{2^i} \cdot R_{d_i} \right), \quad (4.4)$$

isto é,

$$R_T = \frac{R_{a_\mathcal{L}}}{2^\mathcal{L}} + \sum_{i=1}^{\mathcal{L}} \frac{R_{d_i}}{2^i}. \quad (4.5)$$

A Tabela 4.5 exemplifica um esquema de alocação de taxa para as componentes $a_3(n)$, $d_3(n)$, $d_2(n)$ e $d_1(n)$, no sistema de codificação de voz usando DWT (de 3 níveis de decomposição) e QV. Observe-se que os valores de número de níveis N e dimensão K do quantizador vetorial definem a taxa para cada componente. De acordo com a Equação (4.5), a taxa de codificação total para a estratégia de alocação em questão é

$$R_T = \frac{R_{a_3}}{8} + \frac{R_{d_1}}{2} + \frac{R_{d_2}}{4} + \frac{R_{d_3}}{8}, \quad (4.6)$$

ou seja,

$$R_T = \frac{2}{8} + \frac{0,75}{2} + \frac{1}{4} + \frac{1,5}{8} = 1,0625 \text{ bit/amostra}. \quad (4.7)$$

Tabela 4.5: Um exemplo de alocação de taxa para as componentes $a_3(n)$, $d_3(n)$, $d_2(n)$ e $d_1(n)$. Os valores de N e K do quantizador vetorial definem a taxa para cada componente.

Componente	N	K	Taxa (bit/amostra)
$a_3(n)$	256	4	$R_{a_3} = 2,0$
$d_3(n)$	64	4	$R_{d_3} = 1,5$
$d_2(n)$	256	8	$R_{d_2} = 1,0$
$d_1(n)$	64	8	$R_{d_1} = 0,75$

Os sinais de voz¹ utilizados nas simulações correspondem a sentenças extraídas de um conjunto de frases foneticamente balanceadas da língua portuguesa [190]. Os dicionários multiresolucionais utilizados foram projetados com o tradicional algoritmo LBG. Os resultados de codificação apresentados nesta seção, referentes ao uso da quantização vetorial no domínio da DWT, dizem respeito à aplicação de uma decomposição *wavelet* em 3 níveis de decomposição, ou seja, $\mathcal{L} = 3$.

¹Obedecendo as condições da Seção 4.1, a aquisição (resolução 8,0 bit/amostra e taxa de amostragem 8 kHz) dos sinais de voz utilizados nas simulações foi realizada usando uma estação de trabalho Sun[®], instalada com utilitários de processamento de áudio.

Os resultados de codificação foram obtidos com um conjunto de simulações que teve como objetivo avaliar diferentes esquemas de alocação de bits para os sinais (componentes de aproximação e de detalhe) resultantes da decomposição *wavelet* multiresolucional. Para tanto, foram projetados diversos dicionários multiresolucionais, variando-se a dimensão K e o número de níveis N dos quantizadores vetoriais – a taxa correspondente $R = \frac{1}{K} \log_2 N$ é expressa em bit/amostra ou, mais precisamente, em bit/coeficiente. A Tabela 4.6 apresenta algumas estratégias de alocação avaliadas e os respectivos valores de SNRseg e SNRtot obtidos para os sinais reconstruídos. Conforme mencionado anteriormente, a exclusão de $d_1(n)$ implica queda praticamente imperceptível na qualidade subjetiva do sinal reconstruído. Deste modo, adotou-se taxa de codificação de 0,0 bit/amostra para a componente $d_1(n)$, ou seja, exclusão de $d_1(n)$ na realização da IDWT.

A Tabela 4.7 apresenta valores de SNRseg e SNRtot para diversas taxas de um sistema de codificação utilizando quantização vetorial simples, ou seja, no domínio original (a própria forma de onda, sem uso de DWT) do sinal de voz.

As Tabelas 4.6 e 4.7 mostram que, para todas as taxas de codificação consideradas (com exceção da taxa de 0,5 bit/amostra), a QV simples leva a valores de SNRseg superiores aos obtidos com uso de DWT + QV. Com exceção da taxa de 2,0 bit/amostra, a superioridade de QV simples sobre DWT + QV também se manifesta em termos de SNRtot². Entretanto, os testes subjetivos indicaram que, para as taxas de codificação discriminadas na Tabela 4.7, os sinais reconstruídos com uso de DWT + QV apresentam qualidade superior à apresentada pelos sinais reconstruídos mediante utilização de QV simples. De fato, a Tabela 4.8 mostra o resultado de testes de preferência realizados com 20 avaliadores. Utilizou-se a seguinte escala de pontuação: A – a qualidade do sinal obtido com DWT + QV é superior à obtida com QV simples; B – a qualidade do sinal obtido com QV simples é superior à obtida com DWT + QV; C – não se distingue a qualidade dos sinais obtidos com DWT + QV e QV simples. Para a taxa de 2,0 bit/amostra, 95% dos avaliadores consideraram a qualidade do sinal obtido com DWT + QV superior à qualidade do sinal obtido com QV simples; para a taxa de 1,0 bit/amostra, 85% dos avaliadores manifestaram sua preferência pelo sinal ob-

²Em se tratando da avaliação de desempenho dentro do conjunto de treino, constatou-se uma superioridade de DWT + QV sobre QV simples tanto em termos de qualidade objetiva (avaliação dos resultados de SNRseg e SNRtot) como de qualidade subjetiva (realização de testes de escuta) do sinal reconstruído.

tido com uso de DWT + QV; para a taxa de 0,875 bit/amostra, 65% dos avaliadores apontaram a melhor qualidade do sinal obtido com uso de DWT + QV, 10% registraram preferência pelo sinal obtido com QV simples e 25% registraram inexistência de distinção de qualidade. É importante destacar, conforme apresentam as duas últimas linhas da Tabela 4.8, que os avaliadores também se manifestaram favoráveis ao esquema DWT + QV até mesmo para o caso de esse sistema estar operando a uma taxa inferior à apresentada pelo sistema QV simples – 60% dos avaliadores consideraram a qualidade do sinal reconstruído com uso de DWT + QV à taxa de 0,875 bit/amostra superior à qualidade do sinal reconstruído utilizando QV simples à taxa de 1,0 bit/amostra. Ao se comparar DWT + QV à taxa de 0,6875 bit/amostra com QV simples à taxa de 1,0 bit/amostra, 45% dos avaliadores manifestaram preferência pelo sinal reconstruído com uso de DWT + QV, 40% apontaram uma melhor qualidade do sinal obtido com QV simples e 15% registraram inexistência de distinção de qualidade.

Nesse contexto, portanto, é válido observar que, apesar de SNRseg (bem como SNRtot) ser muito utilizada para avaliar o desempenho de codificadores de forma de onda de voz, essa medida não é uma alternativa adequada para análise comparativa de desempenho entre a codificação que utiliza DWT + QV (em que a quantização é levada a efeito no domínio da transformada *wavelet* discreta, ou seja, quantização dos coeficientes *wavelet*) e a codificação que usa QV simples (em que a quantização é realizada diretamente na forma de onda, ou seja, nas próprias amostras do sinal).

A seção a seguir apresenta os resultados obtidos com o uso de dicionários multiresolucionais projetados com o algoritmo SSC.

Resultados Obtidos com o Algoritmo SSC

Comparando as Tabelas 4.6 e 4.9, observa-se que, para todas as taxas de codificação avaliadas, os dicionários multiresolucionais projetados com o algoritmo LBG levam a sinais reconstruídos com maior valor de SNRseg que quando comparados aos sinais reconstruídos obtidos com a utilização de dicionários multiresolucionais SSC³. No entanto, apesar desta superioridade em termos de SNRseg, testes subjetivos informais mostraram que os dicionários SSC e LBG levam a sinais reconstruídos com praticamente a mesma qualidade. Neste contexto, observa-se mais uma vez que a relação sinal-ruído

³Em se tratando de avaliação de desempenho dentro do conjunto de treino, os dicionários multiresolucionais projetados com o algoritmo competitivo SSC em geral evaram a resultados de SNRseg e SNRtot superiores aos obtidos com uso de dicinários multiresolucionais LBG.

segmental (SNRseg) e a relação sinal-ruído total (SNRtot) não constituem medidas objetivas adequadas para análise comparativa de desempenho de sistemas/algoritmos de codificação de forma de onda de voz baseados em wavelets e quantização vetorial.

4.3 Quantização Vetorial de Imagem – Domínio Espacial

Esta seção tem como objetivo apresentar uma avaliação dos algoritmos SOA, SSC e LBG para projeto de dicionários destinados à quantização vetorial de imagens, realizada no domínio espacial. São apresentados resultados referentes à investigação da influência do dicionário inicial no desempenho desses algoritmos. Além disso, são mostradas curvas de relação sinal-ruído de pico (PSNR) versus taxa de codificação para imagens reconstruídas. A seção encerra com a apresentação dos resultados concernentes à avaliação da *qualidade inerente* dos dicionários SOA, SSC e LBG sob o ponto de vista da complexidade computacional envolvida na fase de codificação dos vetores, ou seja, na fase de procura do vizinho mais próximo. Nesse contexto, cumpre salientar que, em se tratando de compressão de sinais baseada em quantização vetorial, os algoritmos SOA, SSC e LBG visam gerar dicionários de qualidade, no sentido de que levem a sinais reconstruídos que apresentem a menor distorção para uma determinada taxa de codificação. A dinâmica dos algoritmos SOA, SSC e LBG não leva em consideração a questão da complexidade na fase de codificação dos vetores. Daí utilizar-se, com efeito, o termo *qualidade inerente*, ao serem abordados aspectos de complexidade computacional. Precisamente, é apresentada uma avaliação de complexidade, em termos de número de multiplicações por amostra, ao serem utilizados os algoritmos de otimização de procura propostos em [34] e [35].

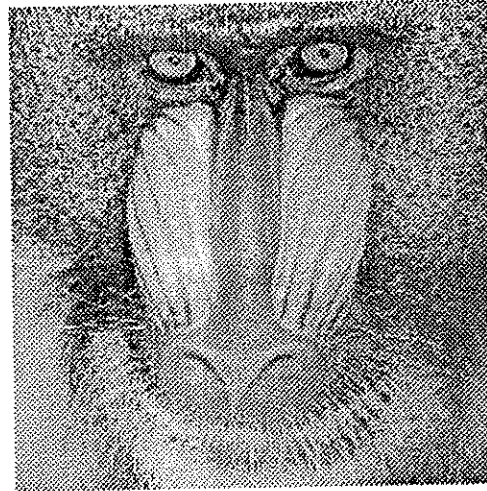
Os resultados apresentados nesta seção dizem respeito à quantização vetorial com dimensão $K = 16$, isto é, a quantização é realizada em blocos de 4×4 pixels. A Figura 4.5 mostra algumas imagens 256×256 , originalmente codificadas a 8,0 bpp, utilizadas no presente trabalho.

4.3.1 Sensibilidade ao Dicionário Inicial

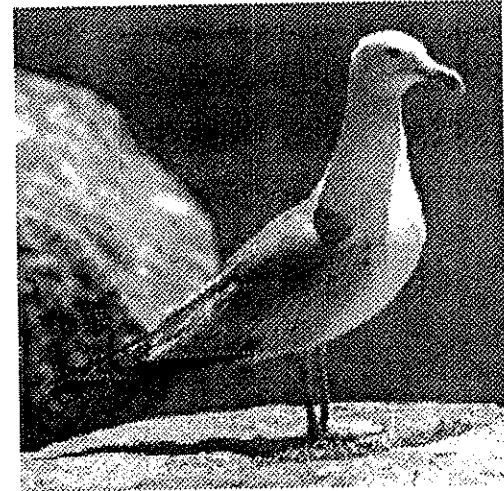
A sensibilidade dos algoritmos SOA, SSC e LBG ao dicionário inicial, utilizando como conjunto de treino a imagem Lena, pode ser avaliada observando-se as Tabe-



(a) Lena



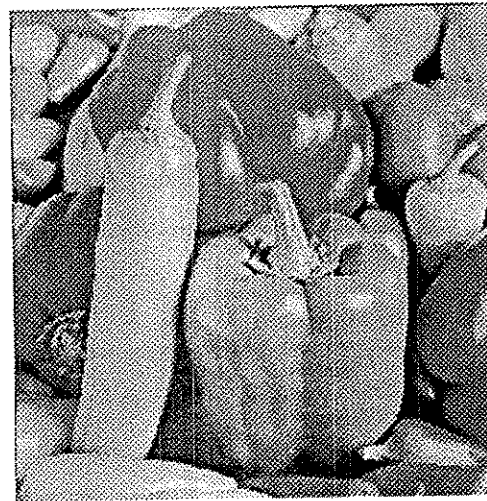
(b) Mandrill



(c) Gull



(d) Frog



(e) Peppers



(f) Goldhill

Figura 4.5: Imagens originais 256×256 utilizadas nas simulações.

las 4.10, 4.11 e 4.12. São apresentados resultados em termos de PSNR da imagem reconstruída com dicionários projetados utilizando três dicionários iniciais diferentes e os correspondentes valores de coeficiente de variação (CV).

Para cada par de valores K e N , a maior sensibilidade corresponde ao maior valor de CV. Por outro lado, a menor sensibilidade corresponde ao menor valor de CV. Assim, considerando $K = 16$ e $N = 128$, a menor sensibilidade ao dicionário inicial está associada ao algoritmo SOA, por apresentar o menor valor de CV (igual a 0,13) dentre os três algoritmos considerados, enquanto que a maior sensibilidade está associada ao algoritmo SSC em virtude de esse algoritmo apresentar o maior CV (igual a 5,23) dentre os algoritmos avaliados.

As Tabelas 4.10, 4.11 e 4.12 mostram que o algoritmo SOA apresenta-se como a alternativa menos sensível ao dicionário inicial. O algoritmo SSC, por sua vez, é o que apresenta maior sensibilidade ao dicionário inicial. O algoritmo LBG, apesar de menos sensível ao dicionário inicial que o algoritmo SSC, não apresenta coeficiente de variação muito menor que o apresentado pelo algoritmo SSC; o algoritmo SOA, por sua vez, apresenta coeficiente de variação muito menor que os apresentados pelos algoritmo SSC e LBG. Isto significa que, em termos gerais, o algoritmo LBG é apenas ligeiramente menos sensível à influência do dicionário inicial que o algoritmo SSC, enquanto que o algoritmo SOA é muito menos sensível à inicialização que os algoritmos LBG e SSC.

Tabela 4.6: Desempenho, fora do conjunto de treino, para diversas estratégias de alocação de bits – valores de SNRseg e SNRtot do sinal reconstruído para diversas taxas de codificação total, R_T (em bit/amostra), com uso de dicionários LBG. Foi realizada DWT com 3 níveis de decomposição. Família *wavelet* utilizada: Daubechies 6 (db6).

	$d_2(n)$	$d_3(n)$	$a_3(n)$	R_T	SNRseg (dB)	SNRtot (dB)
N	256	256	256			
K	2	2	2	2,0	13,20	14,46
R	4,0	4,0	4,0			
N	256	256	256			
K	4	4	4	1,0	6,72	7,04
R	2,0	2,0	2,0			
N	64	256	256			
K	4	4	4	0,875	6,47	6,91
R	1,5	2,0	2,0			
N	64	64	256			
K	4	4	4	0,8125	5,95	6,22
R	1,5	1,5	2,0			
N	256	64	256			
K	8	4	4	0,6875	5,91	6,08
R	1,0	1,5	2,0			
N	256	128	128			
K	8	4	4	0,6875	5,70	6,05
R	1,0	1,75	1,75			
N	-	-	256			
K	-	-	2	0,5	6,12	2,70
R	0,0	0,0	4,0			
N	-	256	256			
K	-	4	4	0,5	5,84	5,98
R	0,0	2,0	2,0			

Tabela 4.7: Desempenho, fora do conjunto de treino, em termos de SNRseg e SNRtot do sinal de voz reconstruído para diversas taxas de codificação total R_T (em bit/amostra). Algoritmo utilizado para projeto de dicionários destinados à quantização vetorial simples: LBG.

K	N	R_T	SNRseg (dB)	SNRtot (dB)
4	256	2,0	13,47	13,75
8	256	1,0	9,42	9,20
8	128	0,875	8,54	9,08
10	32	0,5	4,45	5,83
12	64	0,5	5,48	6,25

Tabela 4.8: Desempenho fora do conjunto de treino - resultados, em termos de percentagens, de testes de preferência realizados com 20 avaliadores. São consideradas várias taxas de codificação R . Algoritmo utilizado para projeto de dicionários: LBG.

Taxa R em DWT + QV	Taxa R em QV simples	Pontuação		
		A	B	C
2,0 bit/amostra	2,0 bit/amostra	95%	5%	0%
1,0 bit/amostra	1,0 bit/amostra	85%	15%	0%
0,875 bit/amostra	0,875 bit/amostra	65%	10%	25%
0,875 bit/amostra	1,0 bit/amostra	60%	20%	20%
0,6875 bit/amostra	1,0 bit/amostra	45%	40%	15%

Tabela 4.9: Desempenho, fora do conjunto de treino, para diversas estratégias de alocação de bits - valores de SNRseg e SNRtot do sinal reconstruído para diversas taxas de codificação total, R_T (em bit/amostra), com uso de dicionários SSC. Foi realizada DWT com 3 níveis de decomposição. Família *wavelet* utilizada: Daubechies 6 (db6).

	$d_2(n)$	$d_3(n)$	$a_3(n)$	R_T	SNRseg (dB)	SNRtot (dB)
N	256	256	256			
K	2	2	2	2,0	13,02	14,76
R	4,0	4,0	4,0			
N	256	256	256			
K	4	4	4	1,0	6,07	6,78
R	2,0	2,0	2,0			
N	64	256	256			
K	4	4	4	0,875	5,93	6,57
R	1,5	2,0	2,0			
N	64	64	256			
K	4	4	4	0,8125	5,37	6,70
R	1,5	1,5	2,0			
N	256	64	256			
K	8	4	4	0,6875	5,23	5,67
R	1,0	1,5	2,0			
N	256	128	128			
K	8	4	4	0,6875	4,76	6,41
R	1,0	1,75	1,75			
N	-	-	256			
K	-	-	2	0,5	3,41	2,10
R	0,0	0,0	4,0			
N	-	256	256			
K	-	4	4	0,5	5,26	5,69
R	0,0	2,0	2,0			

Tabela 4.10: Sensibilidade do algoritmo SOA a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.

K	N	PSNR (dB)			CV (%)
		DI	DII	DIII	
16	32	26,78	26,68	26,75	0,19
16	64	27,80	27,80	27,80	0,00
16	128	29,07	29,02	28,99	0,13
16	256	30,53	30,53	30,56	0,05

Tabela 4.11: Sensibilidade do algoritmo SSC a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.

K	N	PSNR (dB)			CV (%)
		DI	DII	DIII	
16	32	26,52	26,79	24,90	3,92
16	64	27,35	27,64	25,43	4,48
16	128	28,69	28,87	26,25	5,23
16	256	30,00	30,30	27,23	5,80

Tabela 4.12: Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de PSNR (dB). São apresentados os valores de CV referentes aos resultados de PSNR. Conjunto de treino: Lena. Imagem reconstruída: Lena.

K	N	PSNR (dB)			CV (%)
		DI	DII	DIII	
16	32	26,70	26,96	25,81	2,27
16	64	27,60	27,57	25,90	3,60
16	128	28,22	28,38	25,98	4,87
16	256	28,60	29,04	26,45	4,94

4.3.2 PSNR versus Taxa de Codificação

As Figuras 4.6, 4.7 e 4.8 apresentam o desempenho dos algoritmos SOA, SSC e LBG em termos de PSNR para diversas taxas de codificação ao serem reconstruídas as imagens Mandrill, Gull e Frog, respectivamente. Os resultados apresentados foram obtidos com dicionários projetados utilizando-se como conjunto de treino a imagem Lena.

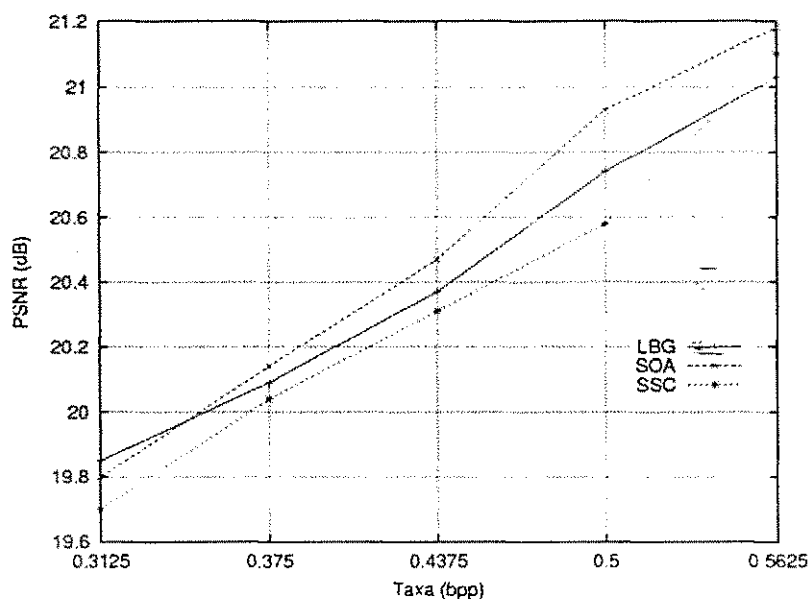


Figura 4.6: Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Mandrill (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$.

No que diz respeito à reconstrução da imagem Mandrill, a Figura 4.6 mostra que, para quase todas as taxas de codificação avaliadas, o algoritmo LBG apresenta um desempenho superior ao apresentado pelo algoritmo SSC. O algoritmo SOA, por sua vez, apresenta-se como a melhor alternativa, levando aos maiores valores de PSNR. Em geral, a diferença de desempenho dos algoritmos SOA, SSC e LBG, à exceção da taxa de 0,5 bpp, gira em torno de 0,2 dB, o que constitui uma diferença bastante pequena.

Em se tratando da quantização vetorial da imagem Gull, observa-se na Figura 4.7 que, à exceção da taxa 0,3125 bpp, o algoritmo SOA apresenta-se como a técnica de projeto de dicionários que leva a imagens reconstruídas com os maiores valores de PSNR. Para as taxas 0,375 bpp e 0,4375 bpp, por exemplo, o algoritmo SOA supera o algoritmo LBG em cerca de 0,4 dB e 0,5 dB, respectivamente. Para taxas inferiores a

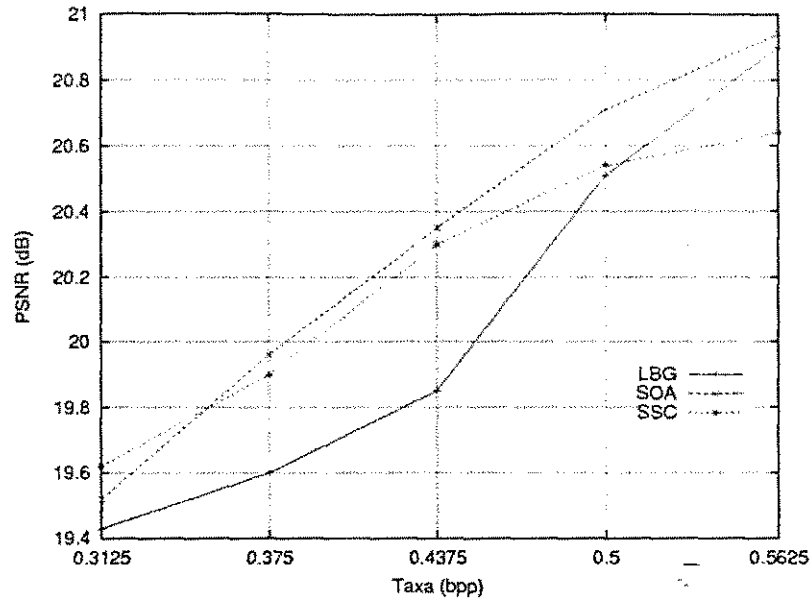


Figura 4.7: Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Gull (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$.

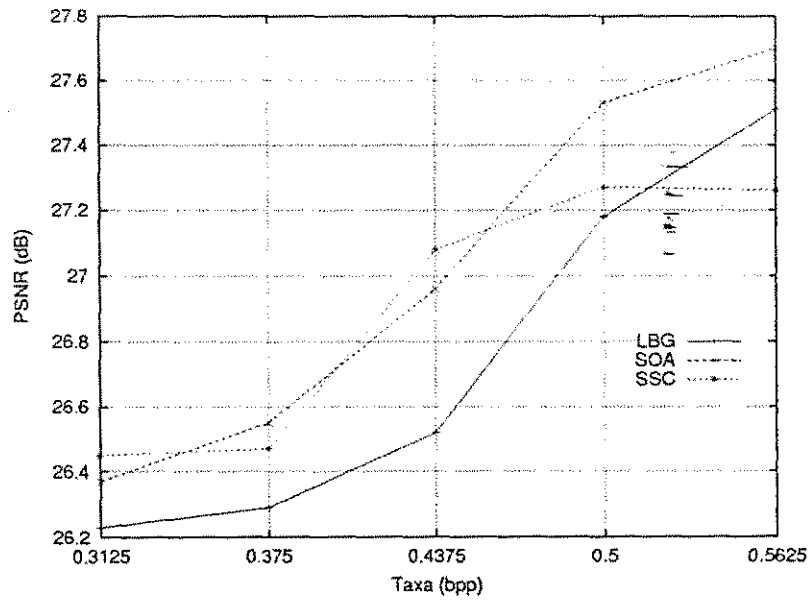


Figura 4.8: Desempenho dos algoritmos SOA, SSC e LBG em QV da imagem Frog (domínio espacial): PSNR da imagem reconstruída versus taxa de codificação para o quantizador vetorial com dimensão $K = 16$.

0,5 bpp, o desempenho do algoritmo SSC é próximo do desempenho do algoritmo SOA, levando a melhores valores de PSNR que os obtidos com o algoritmo LBG. A Figura 4.7 também mostra que, para se obter uma imagem Gull reconstruída com PSNR em torno de 19,6 dB, é necessário utilizar um dicionário LBG à taxa de 0,375 bpp. Para que se obtenha uma imagem Gull reconstruída com a mesma qualidade em termos de PSNR, basta utilizar um dicionário SSC à taxa de 0,3125 bpp.

Quanto à reconstrução da imagem Frog, a Figura 4.8 mostra que, para todas as taxas de codificação na faixa de 0,3125 bpp a 0,5 bpp, os algoritmos SOA e SSC apresentam desempenho superior ao apresentado pelo algoritmo LBG. Ressalte-se que para essa faixa de taxas, os algoritmos SOA e SSC apresentam desempenho próximo. Observa-se, na Figura 4.8, que uma imagem Frog reconstruída com PSNR em torno de 27,5 dB é obtida ao ser utilizado um dicionário SOA à taxa de 0,5 bpp. Para se obter uma imagem Frog reconstruída com aproximadamente essa mesma qualidade em termos de PSNR, faz-se necessário utilizar um dicionário LBG à taxa de 0,5625 bpp. Esse comportamento apresenta-se, também, como um exemplo da adequação do algoritmo SOA para projeto de quantizadores vetoriais de imagem – fixada uma qualidade em termos de PSNR, a quantização vetorial pode ser levada a efeito com menores taxas de codificação ao serem utilizados dicionários SOA em substituição aos dicionários LBG.

Constatou-se que o algoritmo SOA pode apresentar um problema de convergência em sua primeira fase – a escolha de um raio de vizinhança grande pode vir a produzir, ao final da primeira fase, dicionários com todos os vetores-código iguais. Neste contexto, a segunda fase do algoritmo SOA (em que apenas o vencedor é atualizado) pode iniciar-se com um mesmo dicionário, independentemente do dicionário inicial utilizado. Este problema de convergência, entretanto, não compromete a “filosofia” do algoritmo SOA, em que as atualizações ocorrem com base em distâncias medidas no espaço de padrões (espaço em que a QV é levada a efeito), o que se distingue da abordagem de Kohonen, em que as atualizações são realizadas considerando distâncias medidas em um arranjo topológico de nós. A seguinte alternativa para o problema de convergência foi investigada (produzindo dicionários com boa qualidade): promover a atualização dos $P < N$ vetores-código mais próximos do vetor de treino, usando também um fator de alteração da taxa de aprendizagem definido de modo que vetores-código mais próximos/distantes de vetor de treino sejam mais fortemente/fracamente atualizados. Este método (cuja segunda fase atualiza apenas o vencedor) também utiliza distâncias medidas no espaço de padrões para atualizar os vetores-código.

4.3.3 Complexidade Computacional da Etapa de Codificação da QV

Nesta seção é apresentada uma análise comparativa da *qualidade inerente* dos dicionários LBG, SSC e SOA sob o ponto de vista da redução da complexidade computacional envolvida na fase de determinação do vizinho mais próximo (fase de codificação da quantização vetorial), ao serem utilizados os algoritmos de otimização de procura PDS e PDS-O.

Os resultados apresentados dizem respeito à avaliação da redução do número de multiplicações por amostra⁴ envolvido na fase de codificação, em relação ao algoritmo de busca total. Conforme abordado na Seção 2.2, o número de multiplicações por amostra apresentado pelo algoritmo de busca total é igual ao tamanho do dicionário (número de níveis, número de vetores de reconstrução), N .

Em simulações envolvendo quantização vetorial (com dimensão $K = 16$) da imagem Lena, conforme se pode observar na Tabela 4.13, com a utilização do algoritmo PDS, os dicionários projetados com o algoritmo SOA implicam maior redução da complexidade computacional⁵ quando comparados aos dicionários obtidos com os algoritmos LBG e SSC, para todos os valores de R e N avaliados. Para $N = 512$, por exemplo, o dicionário SOA leva a 76,54 multiplicações por amostra, enquanto que os algoritmos SSC e LBG levam, respectivamente, a 94,45 e 135,90 multiplicações por amostra. Observa-se que os dicionários SSC, à exceção de $N = 64$, levam a um menor número de multiplicações por amostra quando comparados aos dicionários LBG.

Um comportamento similar pode ser observado na utilização do algoritmo PDS-O. Na Tabela 4.14, o algoritmo SOA apresenta um desempenho superior ao apresentado pelos algoritmos LBG e SSC, implicando menor complexidade computacional na fase de procura do vizinho mais próximo. Para todos os valores de N considerados, observa-se uma superioridade do algoritmo SSC sobre o algoritmo LBG em termos de redução do número de multiplicações por amostra. Comparando as Tabelas 4.13 e 4.14, observa-se que em todas as simulações o algoritmo PDS-O implica redução de comple-

⁴De acordo com [34], o número de multiplicações por amostra é geralmente utilizado como critério de avaliação de complexidade computacional da fase de codificação por-distância mínima, ou seja, do processo de determinação do vizinho mais próximo.

⁵A redução de complexidade computacional é expressa como $(1 - \frac{N_x}{N}) \cdot 100\%$, em que N_x denota o número de multiplicações por amostra obtido com o algoritmo de otimização de procura do vizinho mais próximo e N denota o número de multiplicações por amostra no algoritmo de busca total

xidade em relação ao algoritmo PDS. É importante observar, também, que a redução de complexidade obtida com os algoritmos PDS e PDS-O tende a aumentar à medida que o número de níveis N aumenta.

Tabela 4.13: Complexidade do algoritmo PDS em termos de número de multiplicações por amostra para a imagem Lena. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.

R (bpp)	N	Número de multiplicações por amostra		
		LBG	SSC	SOA
0,3125	32	11,91 (62,8%)	11,46 (64,2%)	9,95 (68,9%)
0,375	64	19,40 (69,7%)	20,33 (68,2%)	15,95 (75,1%)
0,4375	128	33,10 (74,1%)	32,44 (74,7%)	29,96 (78,9%)
0,5	256	68,60 (73,2%)	55,01 (78,5%)	46,58 (81,8%)
0,5625	512	135,90 (73,5%)	94,45 (81,6%)	76,54 (85,0%)

Tabela 4.14: Complexidade do algoritmo PDS-O em termos de número de multiplicações por amostra para a imagem Lena. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.

R (bpp)	N	Número de multiplicações por amostra		
		LBG	SSC	SOA
0,3125	32	8,58 (73,2%)	8,14 (74,6%)	7,37 (77,0%)
0,375	64	15,13 (76,4%)	13,34 (79,2%)	11,92 (81,4%)
0,4375	128	28,69 (77,6%)	23,81 (81,4%)	20,17 (84,2%)
0,5	256	57,54 (77,5%)	42,95 (83,2%)	35,07 (86,3%)
0,5625	512	114,46 (77,6%)	78,80 (84,6%)	69,05 (86,5%)

O último conjunto de simulações teve como objetivo avaliar a complexidade computacional envolvida ao se codificar uma imagem não pertencente ao conjunto de treinamento dos dicionários. Para tanto, utilizou-se a imagem Mandrill. Considerou-se QV com dimensão $K = 16$. A Tabela 4.15 mostra que, ao se utilizar o algoritmo PDS, os dicionários projetados com o algoritmo SOA, à exceção de $N = 512$, implicam

menor complexidade computacional em relação aos dicionários projetados com os algoritmos LBG e SSC. Observa-se também que o tradicional algoritmo LBG, à exceção de $N = 64$, apresenta-se como a técnica que produz os dicionários que levam à maior complexidade computacional (maior número de multiplicações por amostra) na fase de determinação do vizinho mais próximo. Conforme mostra a Tabela 4.16, com a utilização do algoritmo PDS-O, uma menor complexidade, em termos de número de multiplicações por amostra, é obtida com a utilização dos dicionários SOA para todos os valores de N avaliados, à exceção de $N = 512$. Constata-se que os dicionários LBG implicam maior complexidade computacional dentre todos os dicionários avaliados.

Uma comparação das Tabelas 4.15 e 4.16 mostra que o algoritmo PDS-O, assim como observado nas simulações com a imagem Lena, apresenta desempenho superior ao apresentado por PDS, levando a um menor número de multiplicações por amostra.

Tabela 4.15: Complexidade do algoritmo PDS em termos de número de multiplicações por amostra para a imagem Mandrill. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.

R (bpp)	N	Número de multiplicações por amostra		
		LBG	SSC	SOA
0,3125	32	13,42 (58,1%)	13,15 (58,9%)	12,40 (61,3%)
0,375	64	26,87 (58,0%)	27,47 (57,1%)	23,60 (63,1%)
0,4375	128	51,73 (59,6%)	47,85 (62,6%)	44,01 (65,6%)
0,5	256	108,40 (57,7%)	89,02 (65,2%)	81,00 (68,4%)
0,5625	512	215,94 (57,8%)	166,39 (67,5%)	183,22 (64,2%)

Em resumo, as Tabelas 4.13 e 4.15 mostram que, ao se utilizar o algoritmo PDS para otimização do processo de procura do vizinho mais próximo, o algoritmo SOA apresenta-se como a alternativa mais adequada, produzindo dicionários que levam a uma menor complexidade computacional (menor número de multiplicações por amostra). O algoritmo LBG, por sua vez, é o que produz dicionários que levam ao maior número de multiplicações por amostra na fase de determinação do vizinho mais próximo. De fato, dentre as 10 avaliações (5 valores distintos N para cada uma das duas imagens, Lena e Mandrill, reconstruídas), os dicionários SOA levam a uma menor

Tabela 4.16: Complexidade do algoritmo PDS-O em termos de número de multiplicações por amostra para a imagem Mandrill. Algoritmos utilizados para projeto de dicionários: LBG, SSC e SOA. Entre parênteses está indicada a redução de complexidade em relação ao algoritmo de busca total.

R (bpp)	N	Número de multiplicações por amostra		
		LBG	SSC	SOA
0,3125	32	12,52 (60,9%)	12,08 (62,3%)	11,26 (64,8%)
0,375	64	25,86 (59,6%)	23,10 (63,9%)	19,95 (68,8%)
0,4375	128	49,67 (61,2%)	43,33 (66,2%)	36,57 (71,4%)
0,5	256	103,34 (59,6%)	80,86 (68,4%)	73,55 (71,3%)
0,5625	512	205,27 (59,9%)	155,07 (69,7%)	176,31 (65,6%)

complexidade computacional em 9 casos, enquanto que os dicionários LBG resultam em uma maior complexidade computacional em 8 casos.

No que diz respeito ao algoritmo PDS-O, as Tabelas 4.14 e 4.16 mostram que o menor número de multiplicações por amostra é obtido com uso de dicionários SOA, enquanto o maior número de multiplicações por amostra é obtido com dicionários LBG. Dentre as 10 avaliações (5 valores distintos N para cada uma das duas imagens, Lena e Mandrill, reconstruídas) consideradas, os dicionários SOA levam a uma menor complexidade computacional em 9 casos, enquanto que os dicionários LBG resultam em maior complexidade computacional em 10 casos.

4.4 Quantização Vetorial de Imagem – Domínio *Wavelet*

A seguir é apresentado um conjunto de avaliações relacionadas à aplicação da DWT em imagens. É investigada a importância das sub-bandas da decomposição *wavelet* na qualidade da imagem reconstruída e são apresentados resultados de compressão, em que o desempenho de um sistema de codificação de imagens utilizando DWT + QV é comparado ao desempenho de outros sistemas de codificação.

4.4.1 A Importância do Nível de Decomposição 1

Nesta seção é apresentada a avaliação da importância do nível de decomposição 1 (nível de resolução mais fina) na qualidade da imagem reconstruída. Esse interesse particular no nível 1 reside no fato de que esse nível, que contém 75% dos coeficientes *wavelet*, é de fundamental importância para a definição da taxa de codificação total do sistema de codificação baseado em DWT e QV considerado no presente trabalho. Durante o processo de reconstrução, o erro de quantização dos coeficientes do nível 1 não interfere na qualidade dos demais níveis de decomposição.

A Tabela 4.17 apresenta resultados de relação sinal-ruído de pico (PSNR), em decibéis (dB), para a imagem Lena reconstruída após a exclusão de determinada(s) sub-banda(s) da decomposição *wavelet* multiresolucional. Observa-se, no nível de resolução 1 da decomposição multiresolucional da imagem Lena, que S_{13} (direção diagonal) constitui a sub-banda menos importante para a qualidade da imagem reconstruída, ao passo que S_{11} (direção horizontal) representa a sub-banda mais importante. De fato, o maior valor de PSNR da imagem reconstruída está associado à exclusão de S_{13} , enquanto que o menor valor de PSNR diz respeito à exclusão de S_{11} . Conforme se observa na Tabela 4.17, a exclusão das sub-bandas S_{11} , S_{12} e S_{13} resulta em uma imagem reconstruída com um valor satisfatório de PSNR.

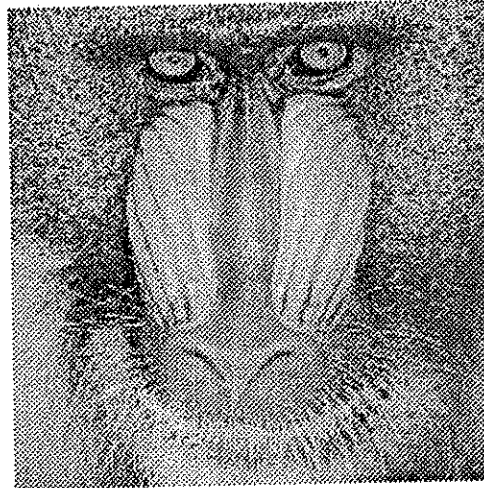
A Figura 4.9 apresenta um conjunto de imagens 256×256 originais (8,0 bpp), bastante usadas para avaliação de algoritmos de codificação de imagens. A Figura 4.10 mostra a qualidade das correspondentes imagens obtidas com aplicação da transformada *wavelet* (utilizou-se a família Daubechies 6) inversa após exclusão das sub-bandas S_{11} , S_{12} e S_{13} . Observa-se que essas imagens apresentam boa qualidade visual.

4.4.2 A Importância das Sub-Bandas do Nível de Decomposição 1

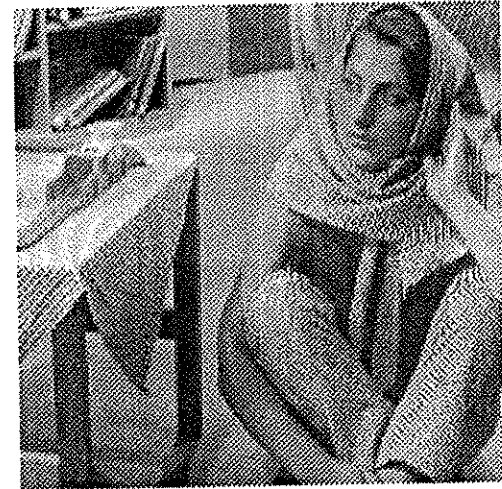
Nesta seção são apresentados resultados e comentários referentes à importância das sub-bandas (ou seja, das direções) do nível de decomposição 1 (nível de resolução mais fina) na qualidade da imagem reconstruída. O interesse em tal avaliação reside no fato de que, em se tratando de uma decomposição *wavelet* multiresolucional, cada uma das sub-bandas, S_{11} , S_{12} e S_{13} , contém 25% dos coeficientes *wavelet*. Desta forma, as estratégias de alocação de bits em tais sub-bandas são de fundamental importância para o compromisso entre a definição da taxa de codificação do sistema de codificação



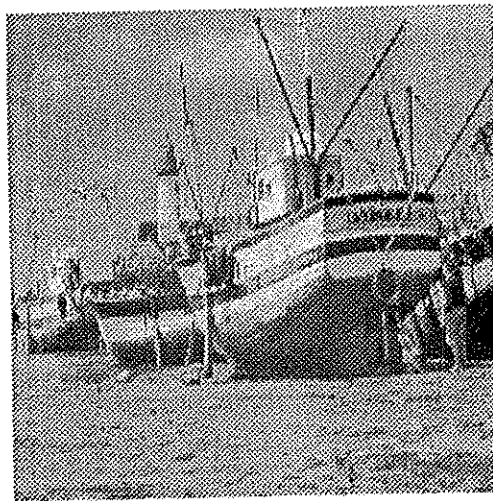
(a) Lena



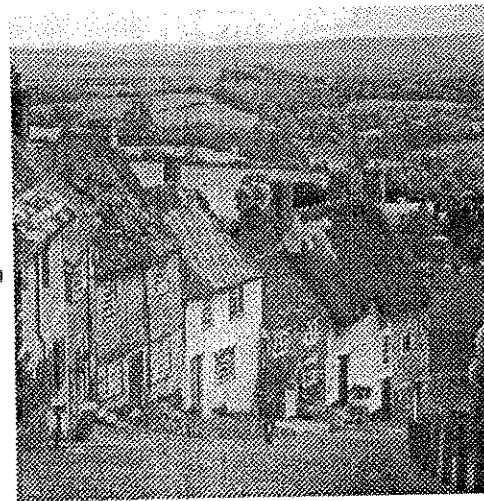
(b) Mandrill



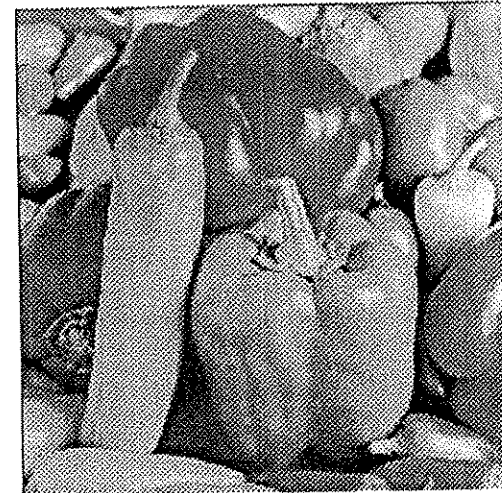
(c) Barbara



(d) Boat



(e) Goldhill

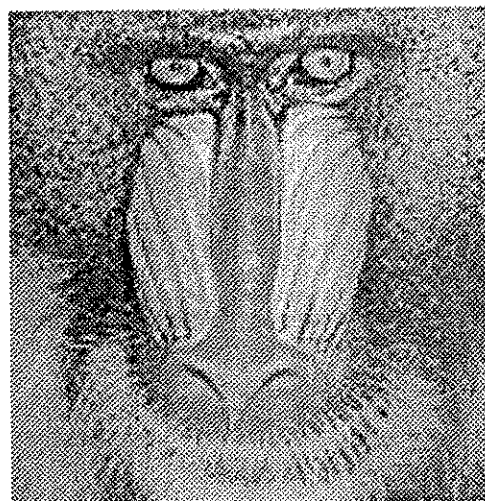


(f) Peppers

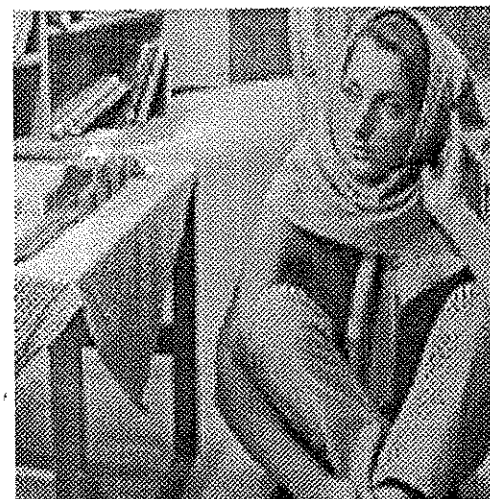
Figura 4.9: Imagens originais 256×256 .



(a) Lena



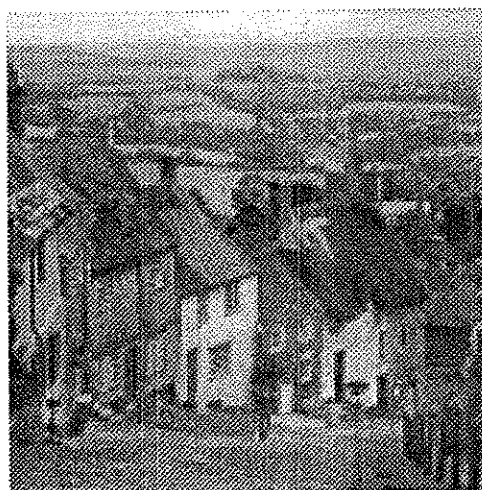
(b) Mandrill



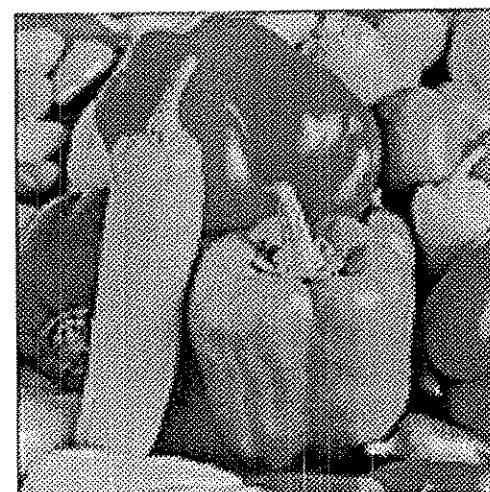
(c) Barbara



(d) Boat



(e) Goldhill



(f) Peppers

Figura 4.10: Imagens reconstruídas após a exclusão das sub-bandas S_{11} , S_{12} e S_{13} .

Tabela 4.17: PSNR da imagem Lena reconstruída após a exclusão de determinada(s) sub-banda(s) da decomposição multiresolucional. *Wavelet* utilizada: Daubechies 6 (db6).

Sub-banda excluída	PSNR (dB)
S_{13}	37,77
S_{12}	36,45
S_{11}	32,95
S_{11} e S_{12}	31,36
S_{11} e S_{13}	31,72
S_{12} e S_{13}	34,06
S_{11} , S_{12} e S_{13}	30,47

baseado em DWT e QV considerado no presente trabalho e a qualidade da imagem reconstruída.

Quanto maior a variância dos coeficientes *wavelet* de uma determinada direção do nível de decomposição 1, maior a importância dessa direção na qualidade da imagem reconstruída. Deste modo, a queda de qualidade da imagem reconstruída (que pode ser traduzida pela obtenção de menores valores de PSNR) ao se descartar uma determinada direção do nível de decomposição 1 aumenta com o aumento da variância dos coeficientes *wavelet* da direção.

Na Tabela 4.18, por exemplo, observa-se que a variância dos coeficientes db6 para a imagem Lena, no nível de decomposição 1 (nível de resolução mais fina), aumenta da direção 3 para a direção 1. Deste modo, ao se descartarem os coeficientes da direção 1 (direção de maior variância dos coeficientes), observa-se uma maior redução de qualidade da imagem reconstruída (menor valor de PSNR) que a obtida ao se descartarem os coeficientes das direções 2 ou 3. Por outro lado, ao se descartarem os coeficientes da direção 3 (direção de menor variância dos coeficientes *wavelet*), observa-se uma menor redução de qualidade da imagem reconstruída (maior valor de PSNR) que a obtida ao se excluírem os coeficientes das direções 1 ou 2.

Cumprе salientar que diversos trabalhos, como por exemplo [65], apontam a direção 3 (diagonal) como a menos importante para a qualidade da imagem reconstruída, no sentido de apresentar o menor conteúdo de energia, a menor contribuição para a qua-

Tabela 4.18: Importância das direções (sub-bandas) do nível de decomposição (resolução) 1 para a imagem Lena. *Wavelet* utilizada: Daubechies 6 (db6).

Nível de decomposição (resolução)	Direção	Sub-banda correspondente	Variância	PSNR (dB) ao se descartarem os coeficientes <i>wavelet</i>
1	1 (Horiz.)	S_{11}	131,49	32,95
1	2 (Vert.)	S_{12}	58,61	36,45
1	3 (Diag.)	S_{13}	43,14	37,77

lidade da imagem reconstruída. Alguns trabalhos (e.g. [64, 75]) têm relatado resultados de codificação obtidos com a exclusão da sub-banda S_{13} . No artigo clássico de Antonini *et al.* [64], em um exemplo de sistemática de compressão de imagens envolvendo quantização vetorial e *wavelets*, os autores excluem a sub-banda S_{13} (ou seja, atribuem a ela uma taxa de codificação de 0 bpp) no procedimento de alocação de bits para as sub-bandas da decomposição *wavelet* multiresolucional. No entanto, a importância da direção 3 depende da imagem em questão: observou-se, por exemplo, durante as avaliações realizadas no presente trabalho, que S_{13} é tão importante quanto S_{11} para a qualidade da imagem Barbara reconstruída. A importância das demais sub-bandas também depende da imagem considerada.

4.4.3 Resultados de Codificação

Uma visão geral do sistema de codificação de imagens baseado em *wavelets* e quantização vetorial considerado no presente trabalho é apresentada na Figura 4.11. Após aplicação de uma decomposição *wavelet* multiresolucional (DWT) com \mathcal{L} níveis de resolução, as sub-bandas S_{ij} , com $i = 1, 2, \dots, \mathcal{L}$ e $j = 1, 2, 3$, são quantizadas por meio de dicionários multiresolucionais. Em outras palavras, à exceção da sub-banda $S_{\mathcal{L}0}$ (correspondente à componente de aproximação no nível de resolução mais baixa), cada sub-banda obtida com a decomposição *wavelet* é quantizada com uso de um dicionário específico, previamente projetado. A sub-banda $S_{\mathcal{L}0}$ é submetida a uma quantização escalar 8,0 bpp, visto que apresenta a maior contribuição para a qualidade da imagem reconstruída. De fato, numa decomposição *wavelet* multiresolucional, em virtude de a variância aumentar de um nível de resolução mais fina para um nível de

resolução mais grosseira, observa-se que os níveis de resolução mais grosseira são os que apresentam maior conteúdo de energia, sendo, portanto, os mais importantes para a qualidade do sinal reconstruído. Além disso, no nível de resolução mais grosseira, a direção 0 (que contém os coeficientes resultantes de filtragens passa-baixa) constitui a sub-banda que apresenta a maior variância e os coeficientes de maior valor. Esta sub-banda, portanto, apresenta os coeficientes mais importantes para a qualidade da imagem reconstruída. Daí optar-se por utilizar para esses coeficientes a quantização escalar a uma taxa de 8,0 bpp. Observa-se na Figura 4.11 que a imagem reconstruída é obtida aplicando-se a transformada *wavelet* discreta inversa (IDWT) nas sub-bandas reconstruídas (quantizadas).

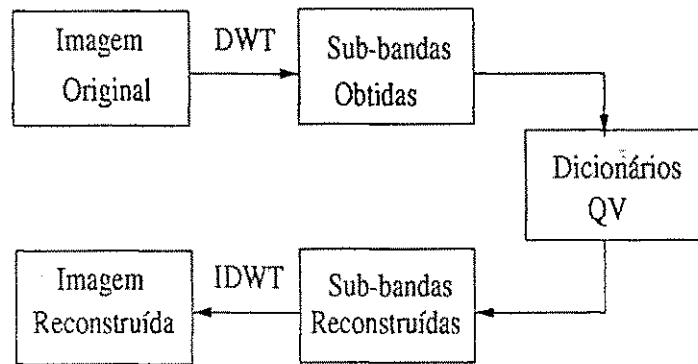


Figura 4.11: Diagrama de blocos do sistema de codificação de imagem em consideração.

Considere uma imagem com $M \times M$ pixels. O número de coeficientes *wavelet* em S_{ij} , com $1 \leq i \leq \mathcal{L}$, é igual a $\frac{M \times M}{2^i \times 2^i}$. Seja $R_{S_{ij}}$ a taxa (em bpp ou, mais precisamente, bit/coeficiente) do quantizador vetorial alocada para a sub-banda S_{ij} , $1 \leq i \leq \mathcal{L}$ e $1 \leq j \leq 3$, e seja $R_{S_{\mathcal{L}0}}$ a taxa (em bpp) do quantizador escalar alocada para a sub-banda $S_{\mathcal{L}0}$. A taxa de codificação total R_T (em bpp) do sistema de codificação de imagens utilizando DWT (de \mathcal{L} níveis de resolução) e QV é dada por

$$R_T = \frac{1}{M \times M} \left(\frac{M \times M}{2^{\mathcal{L}} \times 2^{\mathcal{L}}} R_{S_{\mathcal{L}0}} + \sum_{i=1}^{\mathcal{L}} \sum_{j=1}^3 \frac{M \times M}{2^i \times 2^i} R_{S_{ij}} \right), \quad (4.8)$$

isto é,

$$R_T = \frac{R_{S_{\mathcal{L}0}}}{2^{2\mathcal{L}}} + \sum_{i=1}^{\mathcal{L}} \sum_{j=1}^3 \frac{R_{S_{ij}}}{2^{2i}}. \quad (4.9)$$

A Tabela 4.19 e a Figura 4.12 exemplificam um esquema de alocação de taxa (em bpp) para as sub-bandas S_{ij} , com $1 \leq i \leq 3$ e $1 \leq j \leq 3$, utilizando-se DWT de

3 níveis de resolução (ou seja, $\mathcal{L} = 3$) e quantização vetorial (QV). Observe-se que os valores de número de níveis N e dimensão K do dicionário definem a taxa para cada componente. De acordo com a Equação (4.9), admitindo-se que em S_{30} utiliza-se quantização escalar (QE) com 8,0 bpp, a taxa de codificação total para a estratégia de alocação considerada na Tabela 4.19 é

$$R_T = \frac{8,0}{2^{2 \times 3}} + 3 \times \frac{0,5}{2^{2 \times 1}} + 3 \times \frac{1,75}{2^{2 \times 2}} + 3 \times \frac{2,0}{2^{2 \times 3}} = 0,921875 \text{ bpp.} \quad (4.10)$$

Tabela 4.19: Um exemplo de alocação de taxa para as sub-bandas S_{ij} , com $1 \leq i \leq 3$ e $1 \leq j \leq 3$. Os valores de N e K do quantizador vetorial definem a taxa para cada sub-banda.

Sub-banda	N	K	Taxa (bpp)
S_{11}	256	16	$R_{S_{11}} = 0,5$
S_{12}	256	16	$R_{S_{12}} = 0,5$
S_{13}	256	16	$R_{S_{13}} = 0,5$
S_{21}	128	4	$R_{S_{21}} = 1,75$
S_{22}	128	4	$R_{S_{22}} = 1,75$
S_{23}	128	4	$R_{S_{23}} = 1,75$
S_{31}	256	4	$R_{S_{31}} = 2,0$
S_{32}	256	4	$R_{S_{32}} = 2,0$
S_{33}	256	4	$R_{S_{33}} = 2,0$

Em todas as simulações realizadas, foram utilizadas imagens 256×256 , originalmente codificadas a 8,0 bpp. Em se tratando de QV simples (quantização vetorial no domínio original, ou seja, no domínio espacial; quantização dos *pixels*), bem como de DWT + QV (quantização vetorial no domínio *wavelet*; quantização dos coeficientes *wavelet*), todos os dicionários foram projetados utilizando como conjunto de treino, as imagens Lena, Airplane (apresentada na Figura 4.13) e Mandrill.

Resultados Obtidos com o Algoritmo LBG

Os resultados apresentados nesta seção, concernentes à quantização vetorial simples ou à quantização vetorial realizada no domínio *wavelet*, foram obtidos com dici-

S_{30} QE 8,0 bpp	S_{31} N=256 K=4	S_{21} N=128	S_{11} N=256 Bloco 4X4 (K=16) QV
S_{32} N=256 K=4	S_{33} N=256 K=4	Bloco 2X2 (K=4) QV	
S_{22} N=128 Bloco 2X2 (K=4) QV	S_{23} N=128 Bloco 2X2 (K=4) QV		
S_{12} N=256 Bloco 4X4 (K=16) QV			S_{13} N=256 Bloco 4X4 (K=16) QV

Figura 4.12: Um exemplo de estratégia de alocação de bits para as sub-bandas de uma decomposição *wavelet* (em 3 níveis) de uma imagem. Para QV das sub-bandas S_{ij} , com $1 \leq i \leq 3$ e $1 \leq j \leq 3$, os coeficientes *wavelet* são lidos em blocos quadrados (2×2 ou 4×4) de dimensão K . A sub-banda S_{30} é submetida a QE com 8,0 bpp.

onários projetados com o tradicional algoritmo LBG. Em todas as simulações envolvendo DWT + QV, uma decomposição *wavelet* multiresolucional de 3 níveis (isto é, $\mathcal{L} = 3$) foi realizada, utilizando-se um banco de filtros *wavelet* correspondentes à família Daubechies 6.

O primeiro conjunto de simulações consistiu em realizar uma análise comparativa de diversos algoritmos de codificação de imagens a uma taxa de 0,3125 bpp. Para se obter essa taxa de codificação utilizando quantização vetorial no domínio *wavelet*, a sub-banda S_{30} foi codificada utilizando quantização escalar a 8,0 bpp, as demais sub-bandas do nível de decomposição (nível de resolução) 3 foram codificadas utilizando dicionários com $N = 256$ e $K = 4$ (ou seja, taxa de 2,0 bpp), as sub-bandas do nível de decomposição 2 foram codificadas utilizando dicionários com $N = 256$ e $K = 16$ (ou seja, taxa de 0,5 bpp), ao passo que as sub-bandas do nível de decomposição 1 foram excluídas (ou seja, taxa de 0 bpp) do processo de codificação.

Nas Figuras 4.14, 4.15 e 4.16, são apresentadas algumas imagens reconstruídas à



Figura 4.13: Imagem Airplane original, 256×256 pixels, 8,0 bpp.

taxa de 0,3125 bpp. Foram utilizadas quatro técnicas de codificação diferentes: QV no domínio original da imagem (isto é, QV simples), QV no domínio *wavelet* (isto é, DWT + QV), algoritmo JPEG [59] e algoritmo SPIHT (*set partitioning in hierarchical trees*), introduzido por Said e Pearlman em [70], o qual utiliza codificação (dos coeficientes *wavelet*) do tipo *zerotree*⁶ [60, 68, 191]. Conforme se pode observar, as imagens reconstruídas com utilização de DWT + QV apresentam qualidade superior às imagens reconstruídas com uso de QV simples. De fato, as imagens obtidas com a técnica DWT + QV não apresentam o incômodo “bloqueamento” característico da QV simples a essa taxa. É possível observar também que as imagens reconstruídas com aplicação de DWT + QV apresentam qualidade superior à observada nas imagens obtidas com uso de JPEG que, por utilizar a transformada DCT em blocos, também produz um incômodo efeito de “bloqueamento” da imagem. Uma comparação das imagens reconstruídas mediante uso de JPEG e QV simples mostra que o algoritmo JPEG produz imagens com melhor qualidade visual. Dentre as técnicas de codificação consideradas nesse conjunto de simulações, verifica-se que o algoritmo SPIHT apresenta-se como a mais eficiente, produzindo imagens reconstruídas com boa qualidade à taxa de 0,3125 bpp.

A Tabela 4.20 apresenta os resultados de PSNR para algumas imagens recons-

⁶A codificação do tipo *zerotree* também é utilizada em outros algoritmos que levam a excelentes resultados de codificação, dentre os quais destacam-se o algoritmo EZW (*embedded zerotree wavelet*) [69] e o algoritmo apresentado por Xiong *et al.* em [58]



(a) QV simples



(b) DWT + QV

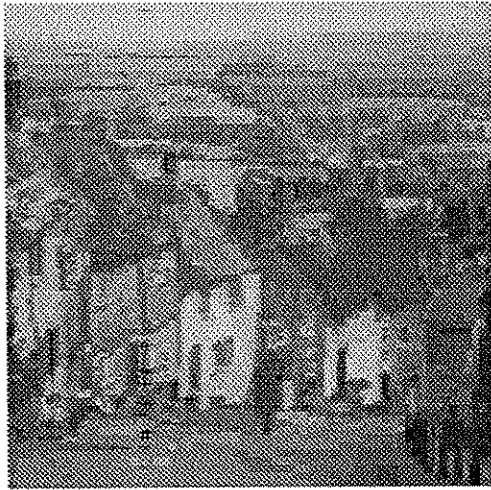


(c) JPEG



(d) SPIHT

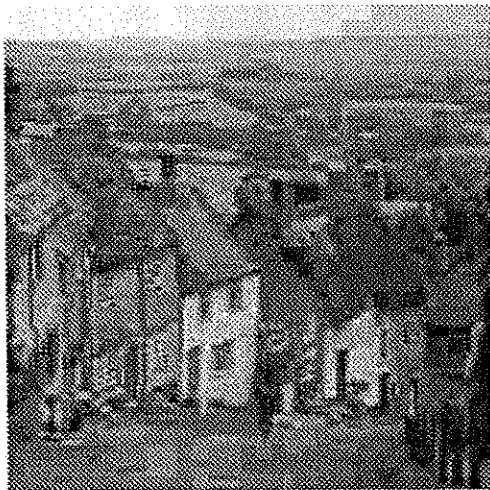
Figura 4.14: Imagem Lena codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.



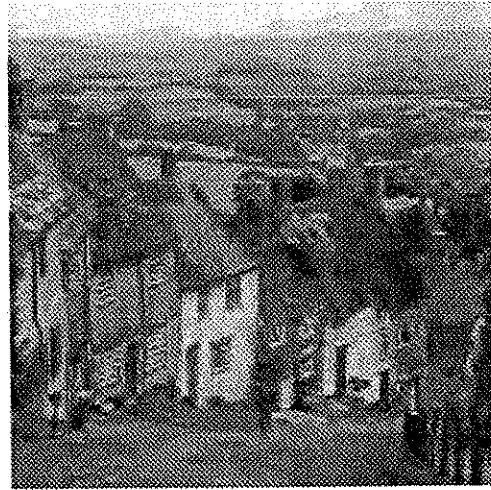
(a) QV simples



(b) DWT + QV



(c) JPEG

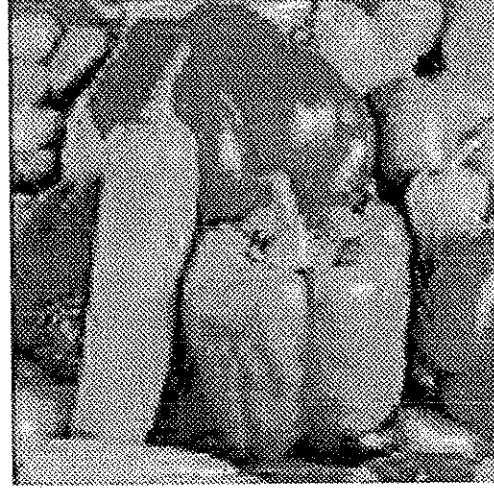


(d) SPIHT

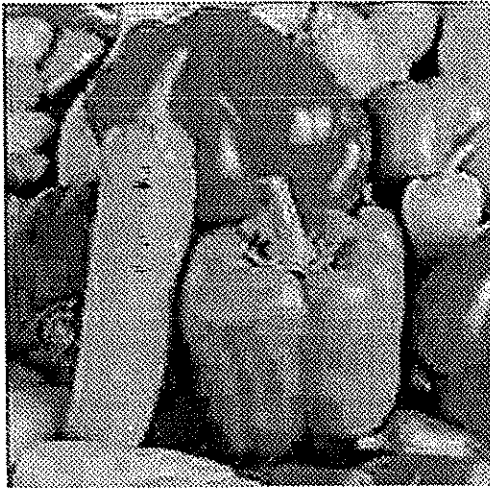
Figura 4.15: Imagem Goldhill codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.



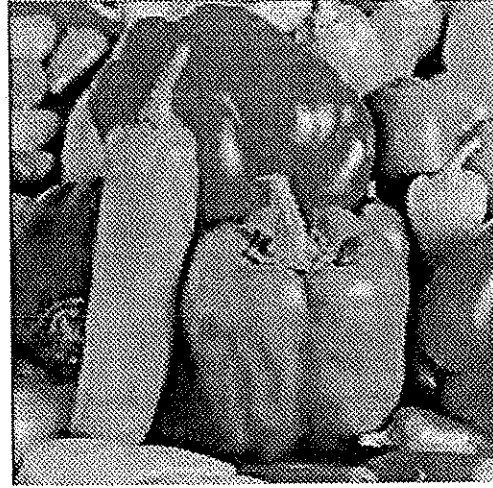
(a) QV simples



(b) DWT + QV



(c) JPEG



(d) SPIHT

Figura 4.16: Imagem Peppers codificada a 0,3125 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.

truídas, à taxa de 0,3125 bpp, ao serem utilizadas as 4 técnicas de codificação previamente mencionadas: QV simples, DWT + QV, algoritmo JPEG e algoritmo SPIHT. Convém salientar que o algoritmo JPEG não permite estabelecer uma taxa de codificação alvo, isto é, uma taxa de codificação precisa, de modo que todos os valores de taxas do algoritmo JPEG apresentados no presente trabalho constituem valores aproximados.

Para todas as imagens consideradas, a Tabela 4.20 mostra que os valores de PSNR obtidos com uso de DWT + QV são superiores aos obtidos com o uso de QV simples. Para as imagens pertencentes ao conjunto de treino (Lena, Airplane e Mandrill) dos dicionários, a técnica de codificação usando DWT + QV levou a valores mais altos de PSNR quando comparada ao algoritmo JPEG – para a imagem Lena, em particular, o uso de DWT + QV permitiu obter uma imagem reconstruída com relação sinal-ruído de pico cerca de 1,4 dB acima da obtida utilizando-se o algoritmo JPEG. Por outro lado, para as imagens não pertencentes ao conjunto de treino (Barb, Goldhill, Peppers e Boat), os resultados de PSNR obtidos com uso de DWT + QV foram iguais ou inferiores aos obtidos com o algoritmo JPEG. No entanto, testes subjetivos informais (sessões de inspeção visual) realizados com essas imagens apontaram uma superioridade da codificação DWT + QV sobre JPEG.

Com relação à Tabela 4.20, convém ressaltar que, ao contrário dos algoritmos JPEG e SPIHT, tanto a quantização vetorial simples quanto a quantização vetorial no domínio *wavelet* (DWT + QV) não utilizaram técnicas de codificação entrópica. Conforme apresentado em [65], o uso de codificação entrópica (como por exemplo o código de Huffman) no cenário de DWT + QV pode representar um aumento de taxa de compressão (ou seja, redução de taxa de codificação) de aproximadamente 30% quando comparado a DWT + QV sem codificação entrópica.

O segundo conjunto de simulações consistiu em realizar uma análise comparativa de diversas técnicas (QV simples, DWT + QV, JPEG e SPIHT) a uma taxa de 0,5 bpp. Para se obter essa taxa de codificação utilizando DWT + QV, a sub-banda S_{30} foi codificada utilizando quantização escalar a 8,0 bpp, as demais sub-bandas do nível de decomposição (nível de resolução) 3 foram codificadas utilizando dicionários com $N = 256$ e $K = 4$ (ou seja, taxa de 2,0 bpp), as sub-bandas do nível de decomposição 2 foram codificadas utilizando dicionários com $N = 64$ e $K = 4$ (ou seja, taxa de 1,5 bpp), ao passo que as sub-bandas do nível de decomposição 1 foram excluídas (ou seja, taxa de 0 bpp) do processo de codificação.

Tabela 4.20: Valores de PSNR (dB) apresentados pelas imagens reconstruídas, à taxa de 0,3125 bpp, para diversas técnicas de codificação. Em QV simples e DWT + QV, foram utilizados dicionários LBG.

Imagem	PSNR (dB)			
	QV simples	DWT + QV	JPEG	SPIHT
Lena	25,84	29,10	27,72	30,42
Airplane	23,89	25,38	25,12	27,58
Mandrill	20,55	21,16	20,40	21,42
Barb	21,81	22,29	22,29	24,62
Goldhill	24,49	25,36	25,84	27,59
Peppers	24,15	25,13	26,28	29,08
Boat	22,98	23,44	24,07	26,28

Nas Figuras 4.17, 4.18 e 4.19, são apresentadas algumas imagens reconstruídas à taxa de 0,5 bpp. Observa-se que as imagens obtidas com uso de JPEG e DWT + QV apresentam qualidade similar, sendo levemente superiores às imagens obtidas com uso de QV simples. Dentre as diversas sistemáticas de codificação consideradas, o algoritmo SPIHT apresentou-se com a técnica que resulta em imagens reconstruídas com melhor qualidade.

Os resultados de PSNR à taxa de 0,5 bpp são apresentados na Tabela 4.21. Em virtude de PSNR nem sempre apresentar uma forte correlação com critérios de avaliação subjetiva, os valores apresentados na Tabela 4.21 devem ser cuidadosamente analisados. Para a imagem Peppers, por exemplo, apesar de o algoritmo JPEG levar a uma imagem reconstruída com PSNR cerca de 3 dB acima da obtida com DWT + QV, uma inspeção visual revela que a imagem obtida com JPEG tem qualidade apenas levemente superior à apresentada pela imagem reconstruída com utilização de DWT + QV.

Resultados Obtidos com o Algoritmo SSC

Esta seção apresenta resultados de codificação ao serem utilizados dicionários multiresolucionais SSC. O desempenho desses dicionários, em termos de PSNR de algumas imagens reconstruídas à taxa de codificação de 0,3125 bpp, é comparado ao desempenho dos dicionários LBG.



(a) QV simples



(b) DWT + QV

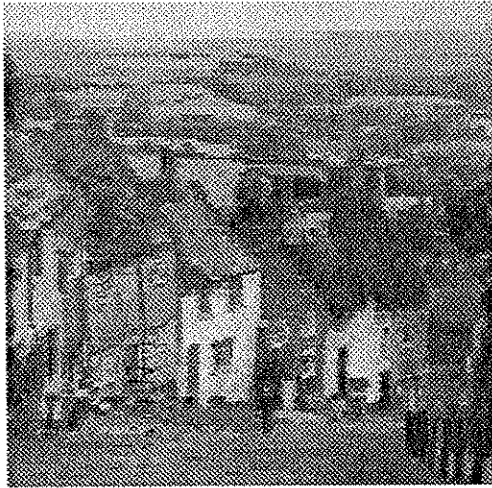


(c) JPEG

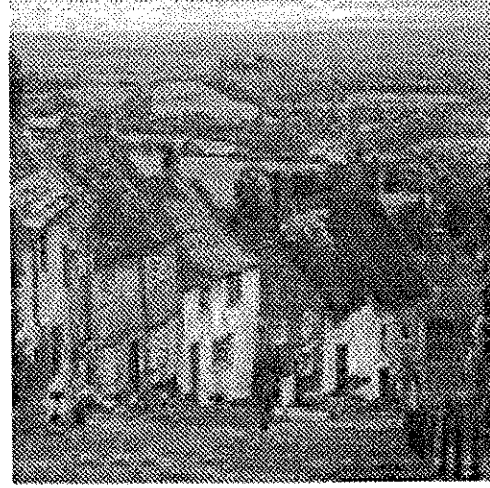


(d) SPIHT

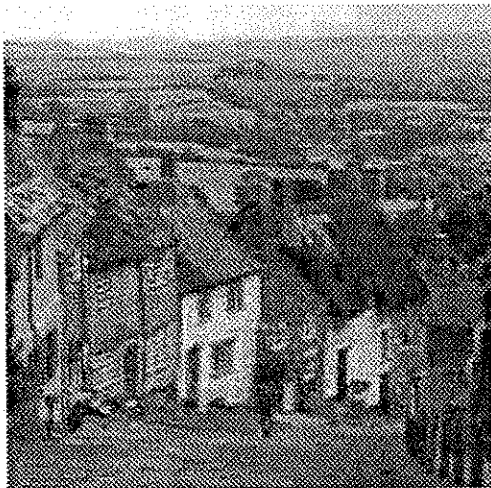
Figura 4.17: Imagem Lena codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.



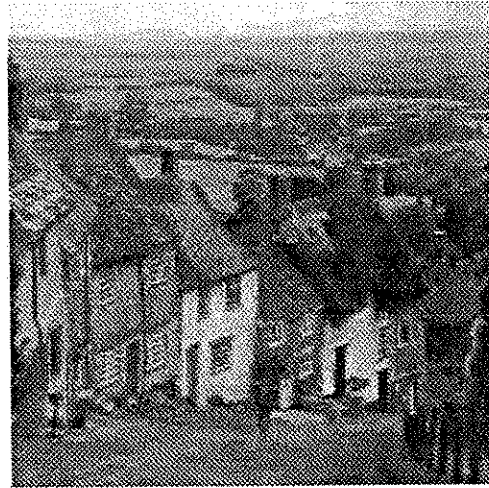
(a) QV simples



(b) DWT + QV

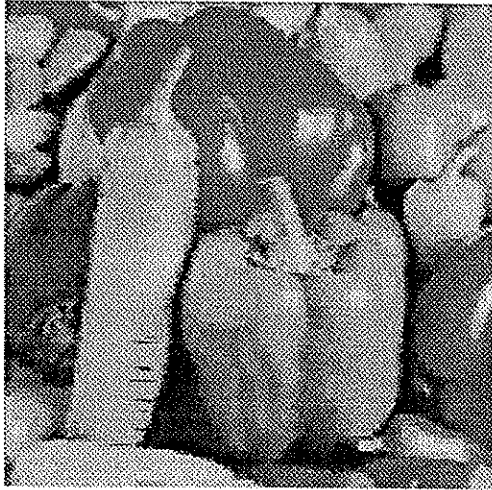


(c) JPEG

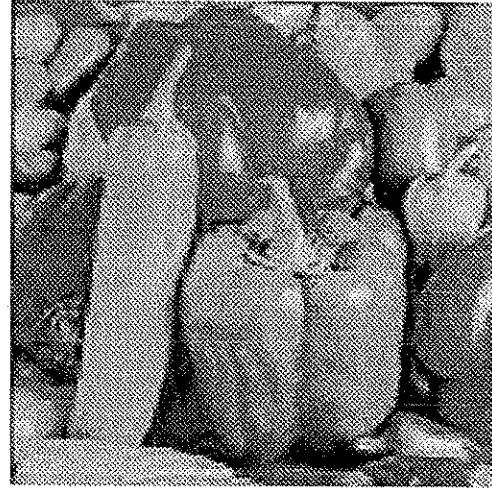


(d) SPIHT

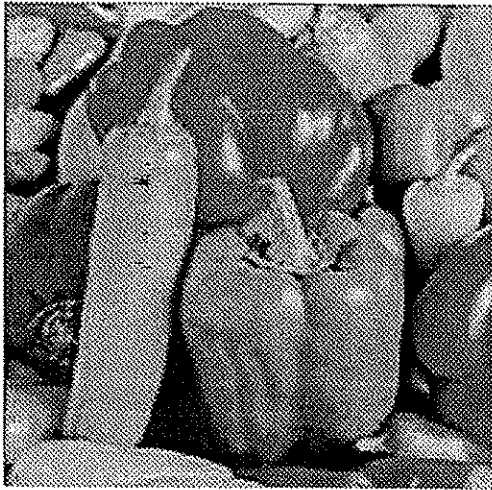
Figura 4.18: Imagem Goldhill codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.



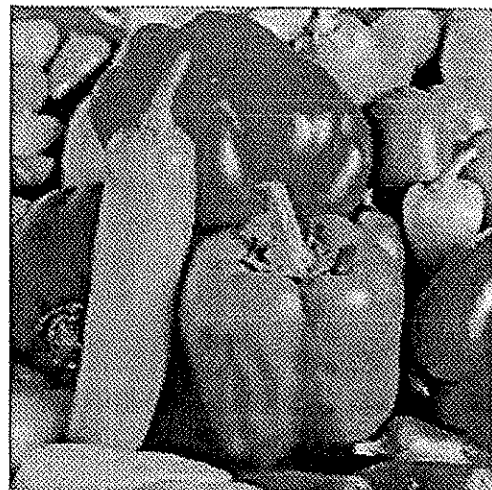
(a) QV simples



(b) DWT + QV



(c) JPEG



(d) SPIHT

Figura 4.19: Imagem Peppers codificada a 0,5 bpp. Em QV simples e DWT + QV, foram utilizados dicionários LBG.

Tabela 4.21: Valores de PSNR (dB) apresentados pelas imagens reconstruídas, à taxa de 0,5 bpp, para diversas técnicas de codificação. Em QV simples e DWT + QV, foram utilizados dicionários LBG.

Imagem	PSNR (dB)			
	QV simples	DWT + QV	JPEG	SPIHT
Lena	28,36	29,08	30,04	32,85
Airplane	25,57	25,64	27,88	30,12
Mandrill	22,18	21,40	21,28	22,64
Barb	22,87	22,84	23,89	26,49
Goldhill	26,01	26,25	27,77	29,34
Peppers	25,89	26,18	29,17	31,99
Boat	24,32	24,38	25,97	28,11

Conforme mostra a Tabela 4.22, com exceção das imagens Lena e Boat (para as quais os dicionários LBG e SSC apresentam praticamente o mesmo desempenho), os dicionários multiresolucionais SSC levam a imagens reconstruídas com maiores valores de PSNR quando comparadas com as obtidas com uso de dicionários multiresolucionais LBG. Para as imagens Airplane e Mandrill, por exemplo, são obtidos ganhos de 0,94 dB e 0,51 dB, respectivamente, com a substituição dos dicionários LBG pelos dicionários SSC.

Tabela 4.22: Desempenho dos dicionários LBG e SSC em codificação de imagens usando DWT + QV. Valores de PSNR de algumas imagens reconstruídas à taxa de 0,3125 bpp.

Imagem	PSNR (dB)	
	LBG	SSC
Lena	29,10	29,02
Airplane	25,38	26,32
Mandrill	21,16	21,67
Barbara	22,29	23,00
Boat	23,44	23,51

4.5 Quantização Vetorial dos Parâmetros LSF

Nesta seção são apresentados resultados concernentes à aplicação dos algoritmos LBG e SSC ao projeto de dicionários para QV dos parâmetros LSF.

Os conjuntos de treino e de avaliação dos dicionários LSF correspondem a cerca de 2400s e 30s de voz, respectivamente [192]. A análise LPC de ordem 10 foi realizada usando o método da autocorrelação a cada 20 ms. Os conjuntos de treino e de avaliação dos dicionários correspondem, respectivamente, a 120061 e 1505 vetores LSF.

Para quantização dos parâmetros LSF, utilizou-se a técnica SVQ (SPLIT VQ) [121]. Cada vetor LSF foi dividido em duas partes. A primeira parte corresponde aos 4 primeiros LSFs e a segunda corresponde aos demais 6 LSFs.

Foram projetados dicionários de diversos tamanhos (N) para os vetores de dimensões 4 e 6.

O alvo do projeto de dicionário para QV dos parâmetros LSF é minimizar a distorção espectral entre os vetores de treino e suas correspondentes versões quantizadas. Como o cálculo da distorção espectral envolve elevada complexidade computacional, o projeto de dicionários é levado a efeito utilizando a distorção euclidiana (ou ainda versões ponderadas desta medida), bem mais amena que a distorção espectral.

Nas Tabelas 4.23 e 4.24 são apresentados resultados em termos de distorção espectral (SD) média, percentagem de *outliers* com distorção espectral entre 2 dB e 4 dB e percentagem de *outliers* com distorção espectral superior a 4 dB. É importante ressaltar que quando o número de bits disponível para quantização dos vetores LSF não é divisível por 2, aloca-se 1 bit a mais para o primeiro subvetor (de dimensão 4). Por exemplo, para uma quantização LSF com um total de 23 bits por quadro (de 20 ms), alocam-se 12 bits para o primeiro subvetor LSF e 11 bits para o segundo subvetor (de dimensão 6) LSF. Para uma quantização LSF com um total de 24 bits por quadro, são alocados 12 bits ($N = 1024$) para cada subvetor LSF. As Tabelas 4.23 e 4.24 mostram que, em geral, os dicionários LBG levam a valores de distorção espectral média iguais ou inferiores aos obtidos com uso de dicionários SSC. Observou-se, durante as avaliações realizadas, que os dicionários SSC levam a valores de distorção euclidiana média iguais ou inferiores aos obtidos com dicionários LBG. O melhor desempenho do algoritmo SSC (quando comparado ao algoritmo LBG) em termos de distorção euclidiana (medida de distorção utilizada no projeto dos dicionários LSF), portanto, não se refletiu em um melhor desempenho em termos de distorção espectral.

Um aspecto deve ser destacado nas Tabelas 4.23 e 4.24: os dicionários SSC levam a uma percentagem de *outliers* inferior à obtida com a utilização de dicionários LBG.

Tabela 4.23: Desempenho dos dicionários LBG ao ser reconstruído o conjunto de treino por meio de SPLIT VQ (4,6).

Nº de bits	SD	<i>Outliers</i>	
		2-4 dB	> 4 dB
20	1,37	12,62 %	0,05 %
21	1,33	11,40 %	0,04 %
22	1,20	6,79 %	0,02 %
23	1,17	6,18 %	0,02 %
24	1,04	3,45 %	0,01 %
25	1,00	3,08 %	0,01 %
26	0,88	1,42 %	0,00 %

Tabela 4.24: Desempenho dos dicionários SSC ao ser reconstruído o conjunto de treino por meio de SPLIT VQ (4,6).

Nº de bits	SD	<i>Outliers</i>	
		2-4 dB	> 4 dB
20	1,39	11,94%	0,03%
21	1,35	10,66%	0,03%
22	1,22	5,38%	0,01%
23	1,18	4,76%	0,00%
24	1,04	1,71%	0,00%
25	1,00	1,49%	0,00%
26	0,87	0,31%	0,00%

As Tabelas 4.25 e 4.26 mostram que, ao ser reconstruído o conjunto de avaliação, os dicionários SSC levam a uma distorção espectral média inferior ou igual à apresentada pelo algoritmo LBG para a faixa compreendida entre 20 e 23 bits. Para a faixa 24-26 bits, os dicionários LBG apresentam um melhor desempenho quando comparados aos dicionários SSC em termos de distorção espectral média. É importante destacar

nas Tabelas 4.25 e 4.26 que em geral os dicionários SSC levam a uma percentagem de *outliers* inferior à obtida com a utilização de dicionários LBG.

Tabela 4.25: Desempenho dos dicionários LBG ao ser reconstruído o conjunto de avaliação por meio de SPLIT VQ (4,6).

Nº de bits	SD	<i>Outliers</i>	
		2-4 dB	> 4 dB
20	1,43	15,15%	0,07%
21	1,40	14,55%	0,07%
22	1,27	9,44%	0,00%
23	1,25	8,90%	0,00%
24	1,14	5,38%	0,00%
25	1,12	5,12%	0,00%
26	1,03	3,72%	0,00%

Tabela 4.26: Desempenho dos dicionários SSC ao ser reconstruído o conjunto de avaliação por meio de SPLIT VQ (4,6).

Nº de bits	SD	<i>Outliers</i>	
		2-4 dB	> 4 dB
20	1,40	12,09%	0,07%
21	1,36	11,23%	0,00%
22	1,27	7,38%	0,00%
23	1,25	6,84%	0,07%
24	1,15	3,85%	0,00%
25	1,13	3,72%	0,00%
26	1,04	2,52%	0,00%

4.6 Capacidade de Obtenção de Padrões Representativos do Conjunto de Treino

Nesta seção é apresentada uma avaliação dos algoritmos SOA e SSC no que diz respeito à capacidade de “aprender” a obter padrões representativos do conjunto de treino.

4.6.1 Desempenho do Algoritmo SOA em Sinais de Voz

As Figuras 4.20 e 4.21 mostram que o dicionário SOA incorpora satisfatoriamente algumas características típicas do sinal de voz. De fato, um maior número de vetores-código é alocado nas regiões de maior predominância dos vetores do sinal de voz usado no treinamento do dicionário, ou seja, a maior parte dos vetores-código está adequadamente alocada nas regiões de baixas amplitudes do sinal de voz. Além disso, observa-se que as componentes dos vetores de reconstrução apresentam-se fortemente correlacionadas – os vetores-código estão posicionados ao longo das direções principais do sinal de voz [193, 194].

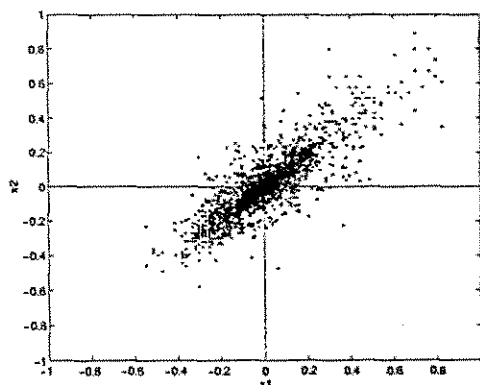


Figura 4.20: Conjunto de treino, correspondente à palavra “aplausos” (0,89 s, 3560 vetores). As coordenadas x_1 and x_2 representam a primeira e a segunda componentes dos vetores de treino \mathbf{x} , respectivamente.

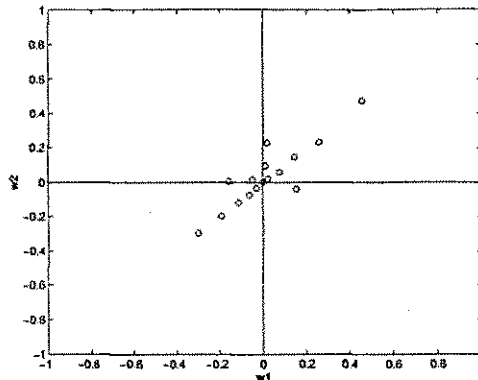


Figura 4.21: Dicionário obtido com o algoritmo SOA: $K = 2$ e $N = 16$. As coordenadas w_1 and w_2 representam, respectivamente, a primeira e a segunda componentes dos vetores-código w_i , $i = 1, 2, \dots, 16$.

4.6.2 Desempenho do Algoritmo SOA em Sinais com Distribuição Conhecida

Em diversos trabalhos [4,17,175,195–198], o desempenho de algoritmos para projeto de quantizadores vetoriais é avaliado utilizando-se fontes com distribuições conhecidas, dentre as quais destacam-se a fonte gaussiana e a fonte de Gauss-Markov de 1^ª ordem, também conhecida como fonte de Markov de 1^ª ordem, fonte de Gauss autoregressiva de 1^ª ordem ou fonte AR(1) [2, 195]. Para efeito de simplicidade, essa fonte será referenciada simplesmente como fonte de Gauss-Markov ao longo do presente trabalho.

O processo discreto de Gauss-Markov $\{X(n)\}$ é definido pela equação

$$X(n) = aX(n-1) + W(n), \quad \forall n, \quad (4.11)$$

em que a denota o coeficiente de correlação e $\{W(n)\}$ é uma seqüência de variáveis aleatórias com distribuição gaussiana, independentes e identicamente distribuídas, com média zero.

O processo de Gauss-Markov pode ser obtido por meio da passagem do ruído gaussiano branco $\{W(n)\}$ através de um filtro com resposta ao impulso [2]

$$h_a(n) = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (4.12)$$

O processo gaussiano (descorrelacionado) é obtido fazendo $a = 0$ na Equação (4.11).

Em todas as simulações relatadas nesta seção, foram utilizados conjuntos de treino com 30000 amostras apresentando variância 0,05 e média zero. O processo de Gauss-Markov utilizado para avaliação dos dicionários projetados apresenta coeficiente de correlação $a = 0,9$.

As Figuras 4.22, 4.23 e 4.24 mostram que o algoritmo SOA produz dicionários que representam adequadamente a distribuição de padrões do conjunto de treino.

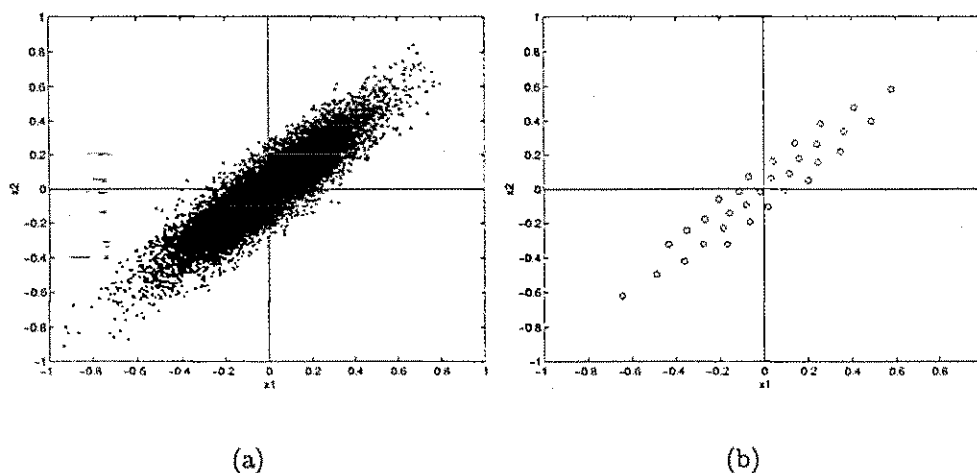


Figura 4.22: Distribuição de Gauss-Markov (coeficiente de correlação $a = 0,9$): (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$.

4.6.3 Desempenho do Algoritmo SSC

As Figuras 4.25 e 4.26 mostram que o algoritmo SSC produz dicionários representativos do conjunto de treino. Essa capacidade de o algoritmo SSC levar a dicionários que incorporam adequadamente as características do conjunto de treino também é observada no contexto de reconhecimento da identidade vocal, conforme atestam os resultados apresentados na Seção 4.7: os dicionários de padrões acústicos obtidos com o algoritmo SSC representam eficientemente os locutores cadastrados pelo sistema de reconhecimento (baseado em quantização vetorial paramétrica) considerado no presente trabalho, levando a resultados satisfatórios em termos de taxas de identificação médias.

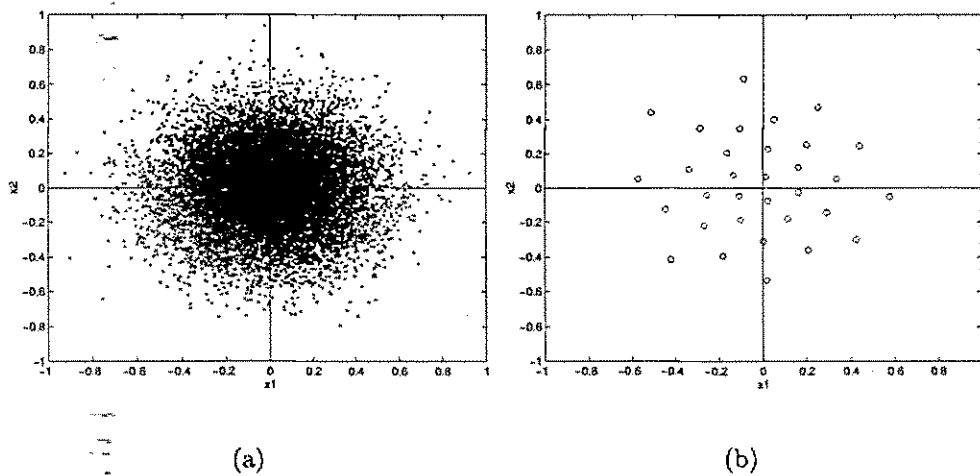


Figura 4.23: Distribuição gaussiana: (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$.

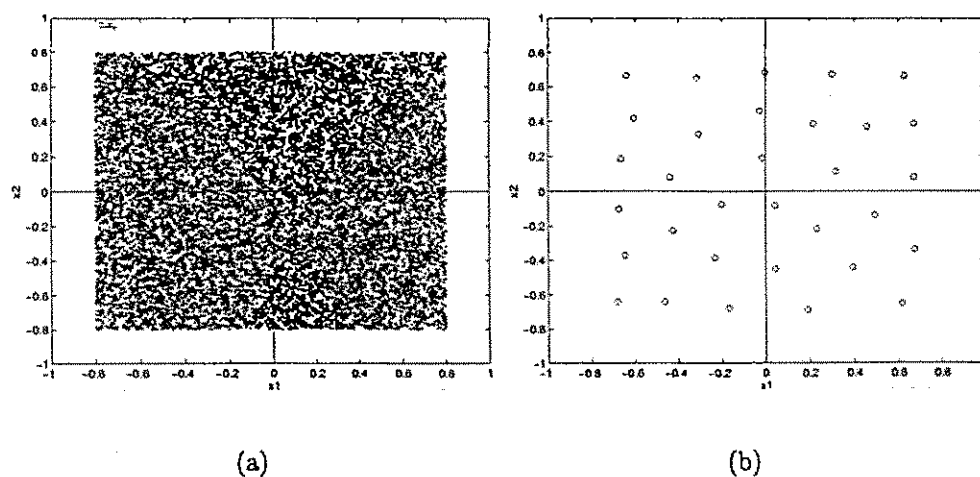


Figura 4.24: Distribuição uniforme: (a) Conjunto de treino, constituído de 15000 vetores, (b) Dicionário SOA, $K = 2$ e $N = 32$.

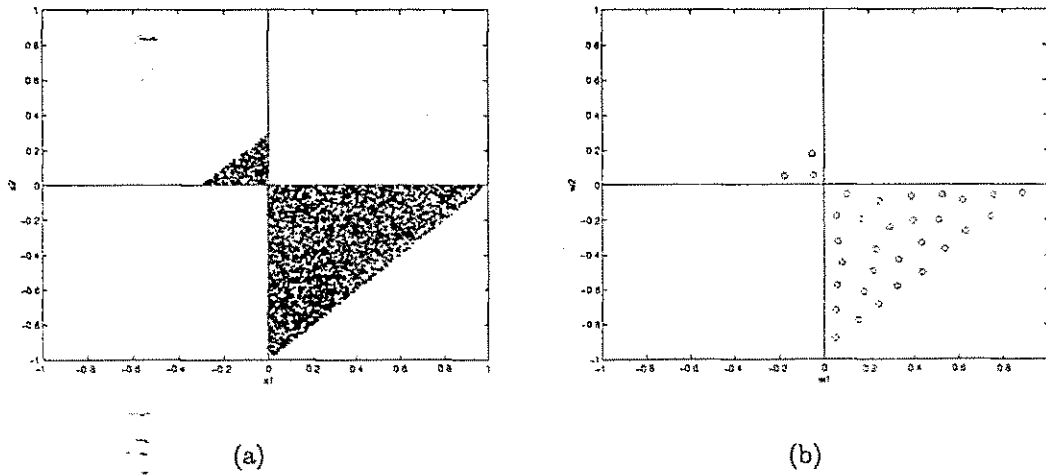


Figura 4.25: Padrões distribuídos sob a forma de uma “borboleta”: (a) Conjunto de treino, (b) Dicionário SSC, $K = 2$ e $N = 32$.

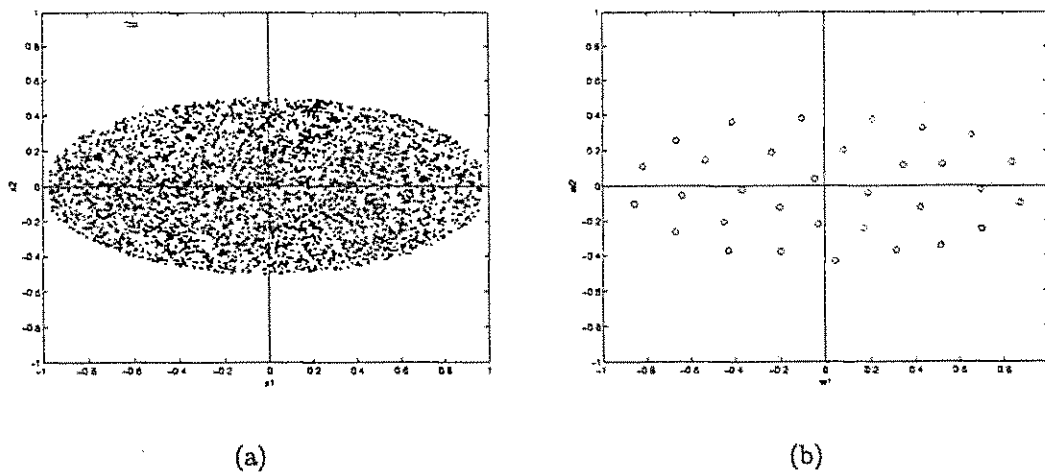


Figura 4.26: Padrões distribuídos sob a forma de uma elipse: (a) Conjunto de treino, (b) Dicionário SSC, $K = 2$ e $N = 32$.

4.7 Quantização Vetorial Aplicada ao Reconhecimento de Locutor

O sistema de identificação de locutor, dependente do texto e baseado em quantização vetorial paramétrica, utilizou a sentença “quero usar a máquina”, com duração aproximada de 1,5 s. Essa sentença foi escolhida por apresentar vários tipos de sons (sonoros, explosivos, surdos). Os sinais de voz foram obtidos em sessões de gravação distintas, utilizando uma placa *Soundblaster*[®] [199] (taxa de amostragem: 11,0 kHz; resolução: 16 bits, mono). A base de dados foi formada por 20 locutores, sendo 10 do sexo masculino e 10 do sexo feminino. Todos os locutores, a maioria constituída de estudantes universitários com idades próximas, foram utilizados para treinamento e teste do sistema de reconhecimento. Durante a fase de treinamento do sistema, a sentença “quero usar a máquina” foi proferida 5 vezes por cada locutor. A fase de teste do sistema contemplou a utilização de 20 elocuições dessa sentença para cada locutor. Os resultados apresentados dizem respeito à utilização de um conjunto fechado de locutores.

A etapa de pré-processamento do sinal de voz consistiu da realização de pré-ênfase do sinal, seguida de segmentação em blocos e janelamento, discutidas a seguir.

As componentes de frequência mais altas do sinal de voz caracterizam-se por apresentarem baixas amplitudes e, por esse motivo, são mais facilmente corrompidas por ruído. Muito embora os sinais de voz apresentem maior concentração de energia nas frequências mais baixas, as componentes de frequência mais altas são responsáveis pela geração de sons surdos (fricativos). Nesse contexto, a pré-ênfase do sinal desempenha um importante papel, visando tornar mais plano o espectro do sinal [110, 135]. Neste trabalho, a pré-ênfase foi realizada por meio da fórmula usual

$$s_p(n) = s(n) - 0,95 \times s(n - 1), \quad (4.13)$$

em que $s(n)$ representa o sinal de voz discreto no tempo e $s_p(n)$ representa o sinal de voz após a pré-ênfase. Observa-se que foi utilizado um fator de pré-ênfase de 0,95.

Uma vez realizada a pré-ênfase do sinal de entrada, este foi segmentado em blocos de 20 ms ($N_A = 220$ amostras) e foi analisado a cada 10 ms. Para cada bloco de amostras, foi feito um janelamento [110, 135], isto é, cada bloco foi multiplicado por uma janela de Hamming $w_h(n)$, visando minimizar os efeitos adversos resultantes da segmentação abrupta, que causa descontinuidades no espectro do sinal de voz. O

janelamento utilizado neste trabalho obedeceu à equação

$$w_h(n) = 0,54 - 0,46 \times \cos(2\pi n/(N_A - 1)), 0 \leq n \leq N_A - 1. \quad (4.14)$$

Para cada janela de Hamming de 20 ms, com superposição de 50%, foi realizada a extração de características. Precisamente, de cada janela foi extraído um conjunto de 12 parâmetros acústicos. A quantização vetorial, portanto, foi levada a efeito com dimensão $K = 12$.

Para cada um dos 20 locutores considerados, foram projetados dicionários com $N = 64$ vetores-código. Foram avaliados 5 parâmetros acústicos distintos: coeficientes LPC, cepestrais (denotados por CEP), cepestrais ponderados (denotados por CEP-P), delta cepestrais⁷ (denotados por DCEP) e delta cepestrais ponderados⁸ (denotados por DCEP-P).

Os resultados apresentados a seguir dizem respeito ao sistema de reconhecimento de locutor (SRL) operando sem uso de limiar de rejeição. A avaliação de desempenho, portanto, diz respeito ao SRL operando com um conjunto-fechado de locutores.

É oportuno mencionar que trabalhos anteriores [149, 200] apresentaram uma avaliação dos algoritmos SOA, SSC e LBG no contexto de um SRL operando com uso de limiar de rejeição. Optou-se por apresentar no presente trabalho resultados obtidos sem uso de limiar de rejeição de modo a avaliar de forma mais coerente a potencialidade dos algoritmos SOA, SSC e FS-SSC (como alternativas para o algoritmo LBG) para o propósito de projeto de dicionários de padrões acústicos. A avaliação de desempenho dos algoritmos para projeto de dicionários apresentada no presente trabalho é de fato mais coerente que a avaliação apresentada em [149, 200] visto que no presente trabalho o desempenho do SRL depende exclusivamente da qualidade dos dicionários projetados, não dependendo (como em [149, 200]) da escolha do limiar de rejeição utilizado.

4.7.1 Resultados

Os resultados apresentados na Tabela 4.27 permitem tecer os seguintes comentários:

⁷Os coeficientes delta cepestrais foram obtidos a partir dos coeficientes cepestrais utilizando a Equação (2.29), com $G = 0,375$ e $Q = 2$ [149, 162].

⁸A operação de ponderação foi levada a efeito utilizando-se uma janela BPL, em que os coeficientes delta cepestrais são ponderados com uso de uma função senoidal, o que permite a *de-ênfase* dos coeficientes delta cepestrais de baixa e alta ordem.

- Considerando os coeficientes LPC, a maior taxa média de identificação é obtida com o uso de dicionários FS-SSC;
- No que diz respeito aos coeficientes cepestrais, a maior taxa média de identificação é obtida com uso de dicionários SSC;
- Quanto aos coeficientes cepestrais ponderados, tanto os dicionários LBG quanto os dicionários FS-SSC levam a uma taxa de identificação superior à obtida com uso de dicionários SSC e SOA;
- No tocante aos coeficientes delta cepestrais, os dicionários SSC levam à maior taxa média de identificação;
- Dentre todos os parâmetros acústicos considerados, os coeficientes delta cepestrais ponderados constituem o único caso em que os dicionários LBG apresentam um desempenho superior (em termos de taxa média de identificação) ao apresentado pelos dicionários SOA, SSC e FS-SSC;
- Dentre todos os parâmetros acústicos considerados, os coeficientes cepestrais apresentam-se como os mais adequados para representar a identidade vocal dos locutores, uma vez que levam às maiores taxas médias de identificação. Os coeficientes LPC, por sua vez, constituem os parâmetros acústicos menos eficientes para o propósito de reconhecimento de locutores, tendo em vista que levam às menores taxas médias de identificação;
- Ao serem utilizados dicionários LBG, a maior diferença de taxas médias de identificação entre o grupo dos locutores femininos (LF) e o grupo dos locutores masculinos (LM) é apresentada pelos coeficientes cepestrais ponderados, seguidos dos coeficientes delta cepestrais ponderados. Ainda com relação aos dicionários LBG, os grupos LF e LM apresentam taxas médias de identificação próximas ao serem utilizados coeficientes cepestrais e delta cepestrais;
- Considerando os dicionários SOA, a menor diferença de taxas médias de identificação entre os grupos LF e LM é apresentada pelos coeficientes cepestrais ponderados, seguidos dos coeficientes delta cepestrais. Observando os resultados apresentados pelos dicionários SOA, os grupos LF e LM apresentam taxas de identificação bastante distintas (diferença de 10%) ao serem utilizados coeficientes LPC. Outra diferença acentuada (de 6%) entre as taxas de identificação

apresentadas por LF e LM pode ser observada ao serem considerados os coeficientes delta cepestrais ponderados;

- No tocante aos dicionários SSC, uma pequena diferença de desempenho entre LF e LM (em termos de taxas médias de identificação) é apresentada pelos coeficientes CEP-P, DCEP e CEP. Os coeficientes LPC, por sua vez, apresentam uma diferença considerável de desempenho (6%) entre LF e LM;
- Quanto aos dicionários FS-SSC, os coeficientes cepestrais são os que apresentam a menor diferença de desempenho (em termos de taxas de identificação) entre LF e LM. Já os coeficientes delta cepestrais ponderados são os que apresentam a maior diferença (de 8%) de desempenho entre LF e LM;
- As taxas de identificação apresentadas por LM são maiores que as taxas apresentadas por LF para os coeficientes LPC, CEP e DCEP-P ao serem utilizados dicionários LBG;
- Ao serem usados dicionários SSC (bem como SOA), as taxas de identificação apresentadas por LM são superiores às taxas apresentadas por LF para todos os parâmetros acústicos (com exceção de DCEP) em consideração;
- Em se tratando de dicionários FS-SSC, as taxas de identificação apresentadas no grupo de locutores masculinos superam as taxas apresentadas no grupo dos locutores femininos para todos os parâmetros acústicos considerados, com exceção de CEP-P.

Os comentários previamente apresentados são resumidos a seguir. Considerando o SRL operando sem uso de limiar de rejeição, para cada parâmetro acústico considerado (com exceção de DCEP-P e CEP-P), a substituição dos dicionários LBG pelos dicionários de padrões acústicos obtidos com algum dos algoritmos não-supervisionados (SOA, SSC ou FS-SSC) leva a uma melhoria de desempenho em termos de taxas médias de identificação. Além disso, observou-se que os coeficientes cepestrais apresentam-se como os parâmetro acústicos que levam ao melhor desempenho do SRL baseado em QV paramétrica, destacando-se também por apresentarem uma pequena diferença em termos de taxas médias de identificação entre o grupo de locutores femininos e o grupo de locutores masculinos.

Visando avaliar a representatividade das taxas médias de identificação obtidas, utilizou-se como medida de dispersão o coeficiente de variação (CV). A Tabela 4.28 mostra que as taxas de identificação médias obtidas com uso de coeficientes cepstrais são as mais representativas, tendo em vista que levam aos menores valores de CV.

4.8 Algoritmo FS-SSC Aplicado ao Projeto de Dicionários para QV de Voz

Os algoritmos FS-SSC, SSC e LBG foram aplicados ao projeto de dicionários destinados à quantização vetorial de forma de onda de voz. Para tanto, foram projetados dicionários para várias taxas de codificação (expressas em bit/amostra) de voz, variando-se a dimensão (K) e o número de níveis (N) dos quantizadores vetoriais projetados. A aquisição (resolução 8,0 bit/amostra e taxa de amostragem 8 kHz) dos sinais de voz utilizados nas simulações foi realizada usando uma estação de trabalho Sun[®], instalada com utilitários de processamento de áudio. A qualidade dos sinais de voz reconstruídos foi avaliada por meio da relação sinal-ruído segmental (SNRseg) [2, 110].

O primeiro conjunto de simulações teve como objetivo avaliar a frequência com que os vetores-código são atualizados nos algoritmos SSC e FS-SSC. As Figuras 4.27 e 4.28 apresentam, respectivamente, para o algoritmo SSC e para o algoritmo FS-SSC, o número de vezes em que cada vetor-código de um dicionário com $K = 4$ e $N = 64$ é atualizado (selecionado como vencedor), ao final de 14560 atualizações, correspondentes a 2 iterações ou 2 passagens completas de um conjunto de treino constituído de 7280 vetores de dimensão $K = 4$. Nas Figuras 4.27 e 4.28, a linha horizontal corresponde ao número médio de seleções como vencedor. Comparando as figuras, observa-se que o algoritmo FS-SSC leva a uma maior uniformidade (constatada também para outras combinações de K e N) quanto ao número de vezes em que cada vetor-código é selecionado como vencedor.

O segundo conjunto de simulações procurou avaliar a homogeneidade da distribuição dos vetores de treino nas diversas células de Voronoi. Utilizou-se, para tanto, a entropia normalizada dos vetores-código (vide Apêndice para maiores detalhes). Conforme mostra a Tabela 4.29, o algoritmo FS-SSC produz dicionários que levam a uma distribuição mais homogênea (maiores valores de entropia normalizada) dos vetores de entrada nas diversas células de Voronoi. Portanto, a utilização da medida de distorção

Tabela 4.27: Taxas médias de identificação para o SRL baseado em QV paramétrica. Avaliação realizada (em um conjunto de 20 locutores cadastrados) sem uso de limiar de rejeição.

Algoritmo	LPC		CEP		CEP-P		DCEP		DCEP-P	
	LF	LM	LF	LM	LF	LM	LF	LM	LF	LM
LBG	88,5%	90,0%	98,5%	99,5%	99,0%	91,5%	94,0%	93,5%	94,0%	98,0%
	89,3%		99,0%		95,3%		93,8%		96,0%	
SOA	86,0%	96,0%	97,0%	100,0%	94,5%	95,0%	93,0%	91,5%	91,5%	97,5%
	91,0%		98,5%		94,8%		92,3%		94,5%	
SSC	88,0%	94,0%	98,5%	100,0%	94,5%	95,0%	97,5%	96,5%	93,0%	96,5%
	91,0%		99,3%		94,8%		97,0%		94,8%	
FS-SSC	90,0%	95,5%	98,0%	100,0%	96,5%	94,0%	91,5%	95,5%	90,0%	98,0%
	92,8%		99,0%		95,3%		93,5%		94,0%	

Tabela 4.28: CV para as taxas médias de identificação para o SRL baseado em QV paramétrica. Avaliação realizada (em um conjunto de 20 locutores cadastrados) sem uso de limiar de rejeição.

Algoritmo	CV para as taxas médias de identificação				
	LPC	CEP	CEP-P	DCEP	DCEP-P
LBG	15,5%	3,5%	13,9%	13,6%	6,7%
SOA	16,1%	5,0%	17,1%	17,8%	10,3%
SSC	15,0%	3,4%	17,1%	7,4%	10,9%
FS-SSC	13,1%	4,5%	14,0%	11,1%	12,3%

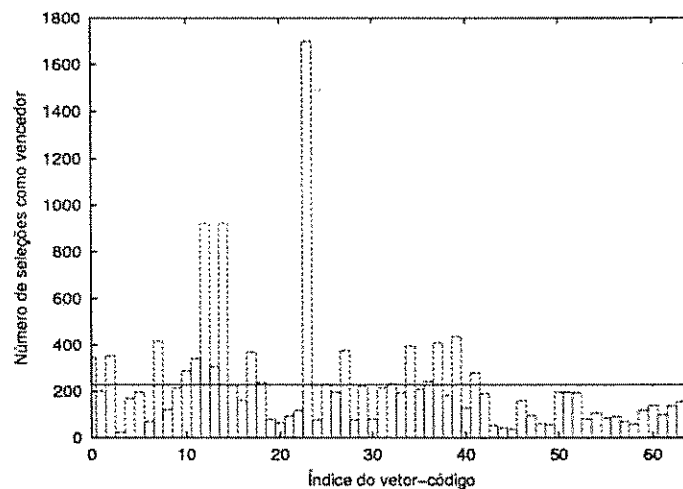


Figura 4.27: Número de vezes em que cada vetor-código de um dicionário SSC com $K = 4$ e $N = 64$ é selecionado como vencedor.

modificada (Equação (3.11), usada pelo algoritmo FS-SSC) em substituição à medida de distorção convencional (Equação (3.8), usada pelo algoritmo SSC) contribui para a redução do número de células de Voronoi pequenas, isto é, contribui para a redução de vetores-código subutilizados.

Observou-se, durante as simulações realizadas, que o algoritmo FS-SSC apresenta uma menor sensibilidade ao dicionário inicial quando comparado ao algoritmo SSC. De fato, uma maior homogeneidade dos valores de SNRseg do sinal de voz reconstruído, considerando o uso de três dicionários iniciais diferentes, é apresentado pelos dicionários FS-SSC.

É importante ressaltar que a velocidade de convergência (número de iterações) do

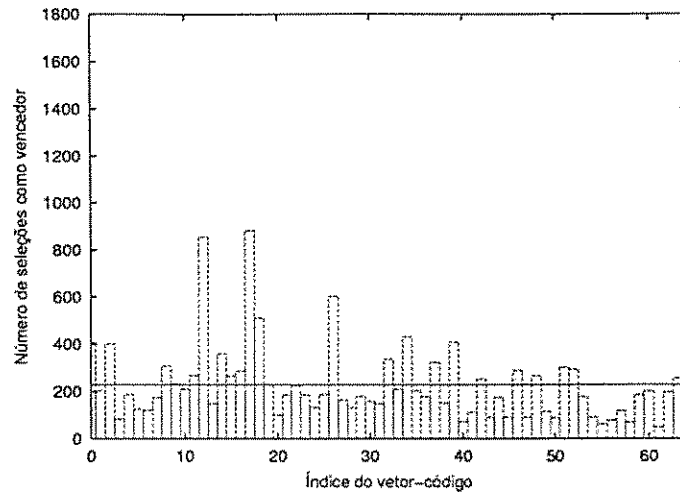


Figura 4.28: Número de vezes em que cada vetor-código de um dicionário FS-SSC com $K = 4$ e $N = 64$ é selecionado como vencedor.

algoritmo LBG depende fortemente do dicionário inicial (conforme pode ser observado na Tabela 5.5 do Capítulo 5). Nos algoritmos FS-SSC e SSC não existe dependência entre número de iterações e dicionário inicial, tendo em vista que um dos parâmetros dos algoritmos FS-SSC e SSC é o número de iterações. Observou-se, durante as simulações, que os algoritmos SSC e FS-SSC necessitam de um número de iterações inferior ao requerido pelo algoritmo LBG. Para os valores de K e N considerados, os algoritmos FS-SSC e SSC necessitam de apenas 2 iterações para projetar dicionários com qualidade próxima ou superior à apresentada pelos dicionários LBG.

As Figuras 4.29 e 4.30 apresentam resultados de SNRseg do sinal de voz reconstruído para diversas taxas de codificação $R = \frac{1}{K} \log_2 N$, em bit/amostra. Em se tratando de QV a elevadas taxas de codificação observa-se nessas figuras que o algoritmo FS-SSC apresenta-se como uma alternativa mais adequada que o algoritmo SSC: para altas taxas de codificação, os sinais reconstruídos com dicionários FS-SSC apresentam valores de SNRseg superiores aos apresentados pelos sinais reconstruídos usando dicionários SSC.

Tabela 4.29: Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) fixada a dimensão $K = 2$ para os dicionários SSC e FS-SSC.

K	N	\tilde{H}	
		SSC	FS-SSC
2	8	0,81	0,87
2	16	0,75	0,90
2	32	0,82	0,89
2	64	0,83	0,90
2	128	0,85	0,92
4	8	0,74	0,96
4	16	0,75	0,91
4	32	0,83	0,93
4	64	0,81	0,92
4	128	0,80	0,93

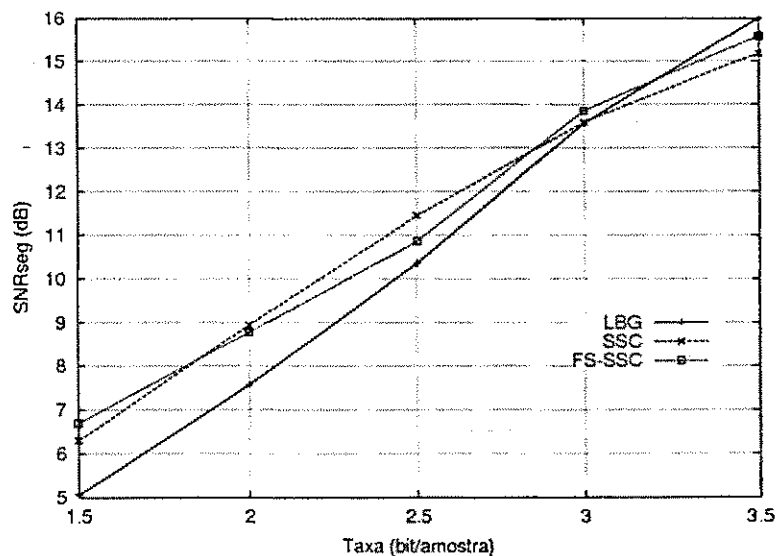


Figura 4.29: Desempenho dos algoritmos SSC, FS-SSC e LBG em QV de forma de onda de voz: SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 2$.

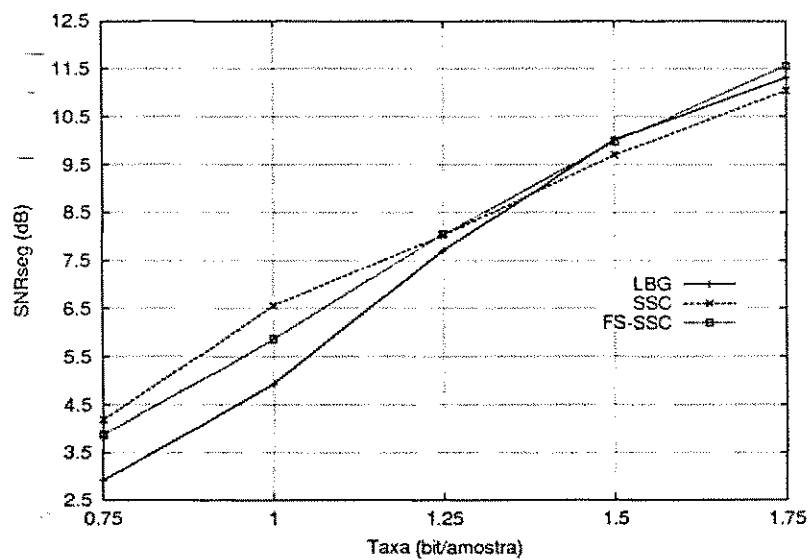


Figura 4.30: Desempenho dos algoritmos SSC, FS-SSC e LBG em QV de forma de onda de voz: SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com dimensão $K = 4$.

Capítulo 5

Complexidade Computacional do Algoritmo SSC

No Capítulo 4 mostrou-se que o algoritmo SSC constitui uma técnica adequada para projeto de dicionários. Em codificação de sinais e em reconhecimento de locutores baseado em quantização vetorial paramétrica, mostrou-se que melhores resultados podem ser obtidos com a substituição dos dicionários LBG por dicionários SSC.

No presente capítulo, é realizada uma avaliação comparativa de complexidade computacional dos algoritmos SSC e LBG. São derivadas expressões analíticas (em função do tamanho do dicionário, da dimensão dos seus vetores-código, do número de vetores do conjunto de treino e do número de iterações desses algoritmos) que estabelecem as condições que devem ser obedecidas para que o algoritmo SSC seja mais eficiente que o algoritmo LBG quanto à complexidade, avaliada por meio da contabilização do número de operações – divisões, multiplicações, comparações, adições e subtrações – realizadas para o treinamento de dicionários.

5.1 Avaliação de Complexidade

Seja n_{LBG} o número total de iterações realizadas pelo algoritmo LBG. É oportuno lembrar que o número de iterações do algoritmo LBG depende do parâmetro de convergência ϵ bem como do dicionário inicial. No caso do algoritmo SSC, por outro lado, o número de iterações é especificado *a priori*, sob a forma do parâmetro n_{SSC} .

Neste trabalho, em se tratando do algoritmo LBG, a distância entre um determinado vetor de treino \mathbf{x} e o vetor-código \mathbf{w}_i é avaliada por meio da distorção de erro quadrático

(distância euclidiana)

$$d(\mathbf{x}, \mathbf{w}_i) = \sum_{j=1}^K (x_j - w_{ij})^2, \quad (5.1)$$

em que w_{ij} é a j -ésima componente do vetor-código \mathbf{w}_i e x_j é a j -ésima componente do vetor de treino \mathbf{x} .

A seguir é apresentada uma avaliação de complexidade dos algoritmos LBG e SSC.

5.1.1 Algoritmo LBG

No *Passo 2* do algoritmo LBG (conforme descrito no Capítulo 2), para que seja determinada a classe a que pertence um vetor de treino, é necessário encontrar a distância entre este vetor e cada um dos N vetores-código do dicionário e depois comparar as distâncias de modo a encontrar o vetor-código mais semelhante: um vetor de treino \mathbf{x} é alocado para a i -ésima classe (célula de Voronoi S_i) se $d(\mathbf{x}, \mathbf{w}_i) < d(\mathbf{x}, \mathbf{w}_j)$, $\forall j \neq i$. A determinação da classe a que pertence um vetor de entrada (vetor de treino) requer, portanto, N cálculos de distância e $N - 1$ comparações. Ao ser utilizada a distância euclidiana, cada cálculo de distância requer K multiplicações, K subtrações e $K - 1$ adições. A alocação de um vetor de entrada em uma das N classes requer, portanto, KN multiplicações, KN subtrações, $(K - 1)N$ adições e $N - 1$ comparações. Tendo em vista que o número de vetores de treino é M , a cada iteração do algoritmo LBG o processo de alocação dos vetores de treino nas respectivas classes requer KNM multiplicações, KNM subtrações, $(K - 1)NM$ adições e $(N - 1)M$ comparações. Em n_{LBG} iterações, este processo de alocação no *Passo 2* do algoritmo LBG requer $KNMn_{\text{LBG}}$ multiplicações, $KNMn_{\text{LBG}}$ subtrações, $(K - 1)NMn_{\text{LBG}}$ adições e $(N - 1)Mn_{\text{LBG}}$ comparações.

Analisa-se agora o número de operações requeridas para a determinação de D_n no *Passo 2* do algoritmo LBG. A distorção D_n é calculada somando as distâncias entre cada vetor de treino e o correspondente vetor-código da classe à qual foi alocado, o que requer $M - 1$ somas de distâncias. Considerando n_{LBG} iterações, a determinação de D_n corresponde a $(M - 1)n_{\text{LBG}}$ adições.

Considera-se agora o *Passo 3*. O teste de convergência do algoritmo LBG requer uma comparação de $(D_{n-1} - D_n)/D_n$ com ϵ . Para que a comparação seja realizada, são necessárias 1 subtração e 1 divisão. Considerando n_{LBG} iterações, o número total de operações requeridas para a realização do teste de convergência do algoritmo LBG

corresponde, portanto, a n_{LBG} comparações, n_{LBG} subtrações e n_{LBG} divisões.

O centróide da i -ésima ($1 \leq i \leq N$) classe é um vetor \tilde{w}_i tal que sua j -ésima componente corresponde à média aritmética das j -ésimas componentes de todos vetores de treino alocados (no *Passo 2*) em S_i . Seja M_i o número de vetores de treino alocados em S_i . Assim,

$$\sum_{i=1}^N M_i = M. \quad (5.2)$$

A j -ésima ($1 \leq j \leq K$) componente \tilde{w}_{ij} do centróide \tilde{w}_i é dada pela média

$$\tilde{w}_{ij} = \frac{1}{M_i} \sum_{x_m \in S_i} x_{mj}, \quad (5.3)$$

em que x_{mj} representa a j -ésima componente do vetor de treino x_m . A determinação do centróide da i -ésima classe requer K cálculos de média de componentes (cada vetor tem K componentes), o que requer $K(M_i - 1)$ adições e K divisões. O número total de adições para a determinação dos N centróides é

$$\sum_{i=1}^N K(M_i - 1) = K \left(\sum_{i=1}^N M_i - \sum_{i=1}^N 1 \right). \quad (5.4)$$

Pela Equação (5.2) segue que para cada iteração do algoritmo LBG, o *Passo 4* requer $K(M - N)$ adições. O número total de divisões no *Passo 4*, por sua vez, é KN . Admitindo-se n_{LBG} iterações, o número total de operações do *Passo 4* corresponde a $K(M - N)n_{\text{LBG}}$ adições e KNn_{LBG} divisões. Além destas operações, é importante ressaltar que o número (M_i , necessário para cálculo do centróide) de vetores de treino alocados para a i -ésima classe é determinado mediante uso de um contador¹, o que implica M_i adições. Considerando as N classes, são utilizados N contadores (um para cada classe), que consomem $\sum_{i=1}^N M_i = M$ adições. Portanto, ao final de n_{LBG} iterações, o *Passo 4* do algoritmo LBG requer, também, Mn_{LBG} adições.

5.1.2 Algoritmo SSC

No algoritmo SSC o número de operações requeridas para a determinação do vencedor, isto é, para a determinação do vetor w_i : $d(x, w_i) < d(x, w_j), \forall j \neq i^*$, é

¹Para ser mais preciso, esta contagem é feita no *Passo 2*, na etapa de alocação dos vetores de treino nas respectivas classes.

igual a $KNMn_{SSC}$ multiplicações, $KNMn_{SSC}$ subtrações, $(K - 1)NMn_{SSC}$ adições e $(N - 1)Mn_{SSC}$ comparações².

Para cada vetor de treino, o vencedor (vetor-código mais semelhante a \mathbf{x} segundo o critério de distância mínima) é atualizado de acordo com as Equações (3.6) e (3.7). Esta atualização requer K adições (uma adição para cada componente de vetor, conforme Equação (3.6)), K multiplicações (uma multiplicação para cada componente de vetor, conforme Equação (3.7)) e K subtrações (uma subtração para cada componente de vetor, conforme Equação (3.7)). Ao final de n_{SSC} iterações e considerando M vetores de treino, a atualização dos vencedores requer KMn_{SSC} adições, KMn_{SSC} multiplicações e KMn_{SSC} subtrações.

Ao final de cada iteração, a taxa de aprendizagem é calculada de acordo com

$$\eta(n) = \eta(1) + (n - 1) \frac{\eta(n_{SSC}) - \eta(1)}{n_{SSC} - 1}. \quad (5.5)$$

Observe que o cálculo da taxa de aprendizagem não é realizado ao final da última iteração. Vale lembrar que $\eta(1)$ e $\eta(n_{SSC})$ são parâmetros do algoritmo SSC. O valor $\frac{\eta(n_{SSC}) - \eta(1)}{n_{SSC} - 1}$ é calculado uma única vez (requerendo 2 subtrações e 1 divisão), ao final da primeira iteração, sendo armazenado para utilização ao final das demais iterações. Ao final de cada iteração (com exceção da última), para que a taxa de aprendizagem seja calculada são necessárias 1 adição, 1 subtração e 1 multiplicação (por $\frac{\eta(n_{SSC}) - \eta(1)}{n_{SSC} - 1}$). Assim, o número total de operações envolvidas nos $n_{SSC} - 1$ cálculos de $\eta(n)$ corresponde a $2 + n_{SSC} - 1$ subtrações, 1 divisão, $n_{SSC} - 1$ adições e $n_{SSC} - 1$ multiplicações.

5.1.3 LBG versus SSC

As Tabelas 5.1 e 5.2 apresentam um resumo do número de adições (ad.), subtrações (sub.), divisões (div.), multiplicações (mult.) e comparações (comp.) requeridas pelos algoritmos LBG e SSC, respectivamente.

A Tabela 5.3 apresenta o número total de operações dos algoritmos LBG e SSC. Observa-se claramente que, em termos de número de divisões executadas, o algoritmo SSC é mais eficiente que o algoritmo LBG. De fato, o algoritmo SSC realiza uma única operação de divisão.

²Vide comentários referentes ao número total de operações necessárias para determinar a classe a que pertence um vetor de entrada, no *Passo 2* do algoritmo LBG.

Tabela 5.1: Número de operações requeridas pelo algoritmo LBG.

	<i>Passo 2</i>
mult.	$KNMn_{\text{LBG}}$
sub.	$KNMn_{\text{LBG}}$
ad.	$[(K - 1)NM + (M - 1)]n_{\text{LBG}}$
comp.	$(N - 1)Mn_{\text{LBG}}$
	<i>Passo 3</i>
comp.	n_{LBG}
sub.	n_{LBG}
div.	n_{LBG}
	<i>Passo 4</i>
ad.	$[(K + 1)M - KN]n_{\text{LBG}}$
div.	KNn_{LBG}

5.2 Condições para as quais SSC é mais Eficiente que LBG em Termos de Multiplicações, Subtrações, Adições e Comparações

Nesta seção, são determinadas as condições para as quais o algoritmo SSC requer um número de multiplicações, subtrações, adições e comparações inferior ao requerido pelo algoritmo LBG. Cada uma destas operações é considerada (por meio dos resultados apresentados na Tabela 5.3) separadamente, conforme segue.

5.2.1 Número de Multiplicações

Para que o número de multiplicações do algoritmo SSC seja menor que o número de multiplicações do algoritmo LBG, deve-se ter

$$[1 + (1 + N)KM]n_{\text{SSC}} - 1 < KNMn_{\text{LBG}}, \quad (5.6)$$

ou seja,

$$n_{\text{SSC}} < \frac{1 + KNMn_{\text{LBG}}}{1 + (1 + N)KM}. \quad (5.7)$$

Tabela 5.2: Número de operações requeridas pelo algoritmo SSC.

Determinação do vencedor	
mult.	$KNMn_{SSC}$
sub.	$KNMn_{SSC}$
ad.	$(K - 1)NMn_{SSC}$
comp.	$(N - 1)Mn_{SSC}$
Atualização do vencedor	
ad.	KMn_{SSC}
mult.	KMn_{SSC}
sub.	KMn_{SSC}
Cálculo da taxa de aprendizagem	
sub.	$n_{SSC} + 1$
div.	1
ad.	$n_{SSC} - 1$
mult.	$n_{SSC} - 1$

Em projetos típicos de dicionários, são utilizados grandes conjuntos de treino (elevados valores de M), de tal forma que $KNMn_{LBG} \gg 1$ e $(1+N)KM \gg 1$. Assim, (5.7) se reduz a

$$n_{SSC} < \frac{N}{1+N}n_{LBG}. \quad (5.8)$$

5.2.2 Número de Subtrações

Para que o número de subtrações do algoritmo SSC seja menor que o número de subtrações do algoritmo LBG, deve-se ter

$$[1 + (1 + N)KM]n_{SSC} + 1 < (1 + KNM)n_{LBG}, \quad (5.9)$$

ou seja,

$$n_{SSC} < \frac{(1 + KNM)n_{LBG} - 1}{1 + (1 + N)KM}. \quad (5.10)$$

Em projetos típicos de dicionários (em que são utilizados grandes conjuntos de treino, o que implica elevados valores de M), $(1 + KNM)n_{LBG} \gg 1$ e $(1 + N)KM \gg 1$,

Tabela 5.3: Número total de operações requeridas pelos algoritmos LBG e SSC.

	LBG	SSC
mult.	$KNMn_{\text{LBG}}$	$[1 + (1 + N)KM]n_{\text{SSC}} - 1$
sub.	$(1 + KNM)n_{\text{LBG}}$	$[1 + (1 + N)KM]n_{\text{SSC}} + 1$
ad.	$[(1 + KN)(M - 1) + (K - N + 1)M]n_{\text{LBG}}$	$[1 + (K - 1)NM + KM]n_{\text{SSC}} - 1$
div.	$(1 + KN)n_{\text{LBG}}$	1
comp.	$[1 + (N - 1)M]n_{\text{LBG}}$	$(N - 1)Mn_{\text{SSC}}$

de modo que (5.10) se reduz a

$$n_{\text{SSC}} < \frac{(1 + KNM)n_{\text{LBG}}}{(1 + N)KM}. \quad (5.11)$$

Tendo em vista que $1 + KNM \approx KMN$, (5.11) se reduz a

$$n_{\text{SSC}} < \frac{N}{1 + N}n_{\text{LBG}}. \quad (5.12)$$

5.2.3 Número de Adições

Para que o número de adições do algoritmo SSC seja menor que o número de adições do algoritmo LBG, deve-se ter

$$[1 + (K - 1)NM + KM]n_{\text{SSC}} - 1 < [(1 + KN)(M - 1) + (K - N + 1)M]n_{\text{LBG}}, \quad (5.13)$$

ou seja,

$$n_{\text{SSC}} < \frac{1 + [(1 + KN)(M - 1) + (K - N + 1)M]n_{\text{LBG}}}{1 + (K - 1)NM + KM}. \quad (5.14)$$

Considerando projetos típicos de dicionários (isto é, considerando elevados valores de M), segue que $[(1 + KN)(M - 1) + (K - N + 1)M]n_{\text{LBG}} \gg 1$ e $(K - 1)NM + KM \gg 1$, de modo que (5.14) se reduz a

$$n_{\text{SSC}} < \frac{[(1 + KN)(M - 1) + (K - N + 1)M]n_{\text{LBG}}}{(K - 1)NM + KM}. \quad (5.15)$$

Tendo em vista que $M - 1 \approx M$, (5.15) se reduz a

$$n_{\text{SSC}} < \frac{[(1 + KN)M + (K - N + 1)M]n_{\text{LBG}}}{(K - 1)NM + KM}. \quad (5.16)$$

Após algumas manipulações, (5.16) se reduz a

$$n_{\text{SSC}} < \frac{[(K-1)(1+N)+3]}{(K-1)N+K} n_{\text{LBG}}. \quad (5.17)$$

5.2.4 Número de Comparações

Para que o número de comparações do algoritmo SSC seja menor que o número de comparações do algoritmo LBG, deve-se ter

$$(N-1)Mn_{\text{SSC}} < [1+(N-1)M]n_{\text{LBG}}, \quad (5.18)$$

ou seja,

$$n_{\text{SSC}} < \frac{[1+(N-1)M]n_{\text{LBG}}}{(N-1)M}. \quad (5.19)$$

Como $1+(N-1)M \approx (N-1)M$ em projetos típicos de dicionários (ou seja, para elevados valores de M), a condição (5.19) se reduz a

$$n_{\text{SSC}} < n_{\text{LBG}}. \quad (5.20)$$

5.2.5 Considerações Gerais

Considere as Inequações (5.8) e (5.12), referentes, respectivamente, ao número de multiplicações e subtrações. À medida que N aumenta, $\frac{N}{1+N}$ tende a 1. Deste modo, à medida que N aumenta, as condições (5.8) e (5.12) tendem a se reduzir simplesmente à condição $n_{\text{SSC}} < n_{\text{LBG}}$. Considere agora o pior caso, ou seja, o menor valor possível de N , isto é, $N = 1$. Para este valor de N as condições (5.8) e (5.12) são satisfeitas para

$$n_{\text{SSC}} < \frac{n_{\text{LBG}}}{2}. \quad (5.21)$$

Portanto, como $\frac{n_{\text{LBG}}}{2} \leq \frac{N}{1+N}n_{\text{LBG}}$, é lícito afirmar que, uma vez satisfeita a condição (5.21), segue que a restrição em (5.8) e (5.12) é automaticamente satisfeita. Portanto, o número de multiplicações e subtrações realizadas pelo algoritmo SSC é inferior ao número de multiplicações e subtrações do algoritmo LBG caso a condição (5.21) seja satisfeita.

Considere a Inequação (5.17). Conforme mostra a Figura 5.1, para $K \geq 1$ e $N \geq 1$, segue que

$$\frac{[(K-1)(1+N)+3]}{(K-1)N+K} > 1, \quad (5.22)$$

de modo que

$$\frac{[(K-1)(1+N)+3]}{(K-1)N+K} n_{\text{LBG}} > n_{\text{LBG}}. \quad (5.23)$$

Assim, uma vez assegurado que

$$n_{\text{SSC}} < n_{\text{LBG}}, \quad (5.24)$$

a condição (5.17) é automaticamente satisfeita. Portanto, uma vez satisfeita a condição (5.24), o número de adições (como também o número de comparações³) realizadas pelo algoritmo SSC é inferior ao número de adições (comparações) do algoritmo LBG. Tendo em vista que a condição (5.21) é mais severa que a condição (5.24) é lícito

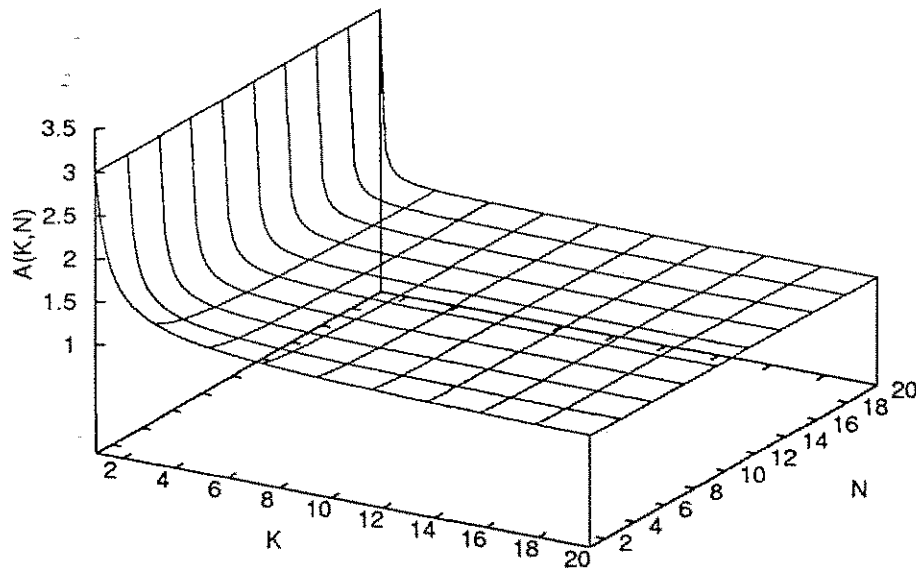


Figura 5.1: Gráfico de $A(K, N) = \frac{[(K-1)(1+N)+3]}{(K-1)N+K}$.

afirmar que a condição (5.21) deve ser satisfeita para que o número de operações do algoritmo SSC seja inferior ao número de operações do algoritmo LBG.

Avaliações realizadas neste trabalho mostraram que condição (5.21) em geral é satisfeita em projeto de dicionários aplicados à codificação de imagens, à codificação de sinal com distribuição de Gauss-Markov e à codificação de forma de onda de voz. Para os casos considerados na Tabela 5.4, por exemplo, o algoritmo SSC requer um número de iterações menor que metade do número de iterações executadas pelo algoritmo LBG.

³Conforme mostra a Inequação (5.20).

Tabela 5.4: Número de iterações realizadas pelos algoritmos LBG e SSC para alguns valores de K e N (com a correspondente taxa de codificação R indicada). São consideradas três fontes diferentes.

Fonte	K	N	R	Iterações	
				LBG	SSC
Imagem	16	32	0,3125 bpp	79	2
	16	64	0,375 bpp	68	2
	16	128	0,4375 bpp	134	5
Gauss- Markov	6	64	1,0 bit/amostra	66	2
	7	128	1,0 bit/amostra	39	2
	8	256	1,0 bit/amostra	17	2
Voz	2	8	1,5 bit/amostra	29	2
	2	16	2,0 bit/amostra	44	2
	4	64	1,5 bit/amostra	68	2
	4	128	1,75 bit/amostra	63	2

Na seção a seguir são apresentados resultados concernentes ao projeto de dicionários aplicados à codificação de forma de onda de voz.

5.3 Resultados

Os projetos de dicionários considerados nesta seção foram realizados utilizando um conjunto de treino correspondente a 10 frases foneticamente balanceadas (18,76s, 150080 amostras), extraídas de [190], proferidas por 10 locutores, sendo 5 do sexo masculino e 5 do sexo feminino.

A Tabela 5.5 mostra que a velocidade de convergência (número de iterações) do algoritmo LBG depende fortemente do dicionário inicial utilizado. No algoritmo SSC, por outro lado, o número de iterações é especificado *a priori*, como um parâmetro do algoritmo. Para todos os valores de dimensão (K) e número de níveis (N) considerados, o algoritmo SSC necessitou de apenas duas iterações para produzir dicionários com qualidade próxima ou superior à apresentada pelos dicionários LBG. Em outras palavras, apesar de serem obtidos após um número de iterações menor que metade do

número de iterações (passagens completas do conjunto de treino) realizadas pelo algoritmo LBG, os dicionários SSC levam a sinais reconstruídos com qualidade (avaliada por meio da relação sinal-ruído segmental [2, 110]) próxima ou superior à apresentada pelos sinais reconstruídos com uso de dicionários LBG.

Tabela 5.5: Sensibilidade do algoritmo LBG a três dicionários iniciais diferentes (DI, DII e DIII) em termos de número de iterações (n_{LBG}).

K	N	n_{LBG}		
		DI	DII	DIII
2	8	29	23	23
2	16	45	44	33
2	32	60	73	68
2	64	81	108	98
2	128	126	180	118
4	8	18	31	16
4	16	31	32	27
4	32	43	57	41
4	64	67	64	68
4	128	82	88	63

A Tabela 5.6 mostra claramente que as Inequações (5.8), (5.12) e (5.17) de fato podem consideradas em lugar de (5.7), (5.10) e (5.14), respectivamente, isto é, o segundo membro de cada uma das Inequações (5.8), (5.12) e (5.17) de fato constituem excelentes aproximações para o segundo membro de cada uma das Inequações (5.7), (5.10) e (5.14), respectivamente⁴. A Tabela 5.6 apresenta expressões calculadas para $K = 2$. Tendo em vista que o conjunto de treino considerado nesta seção é constituído de 150080 amostras de voz, o número de vetores de treino é $M = 75040$. Conforme mencionado anteriormente, os dicionários SSC foram projetados com $n_{\text{SSC}} = 2$. Quanto ao algoritmo LBG, para cada valor de N considerado, o valor de n_{LBG} utilizado na tabela corresponde ao número de iterações (dentre as 3 possibilidades, referentes ao uso de 3 dicionários iniciais diferentes: DI, DII e DIII) que levaram ao dicionário de

⁴Observou-se também que o segundo membro de (5.20) representa uma excelente aproximação para o segundo membro de (5.19).

melhor qualidade. Assim, utilizou-se n_{LBG} igual a 29, 44, 73, 81 e 180 para N igual a 8, 16, 32, 64 e 128, respectivamente.

A Tabela 5.7 também respalda a aproximação do segundo membro de cada uma das Inequações (5.7), (5.10) e (5.14) pelo segundo membro de cada uma das Inequações (5.8), (5.12) e (5.17), respectivamente. Na Tabela 5.7, $M = 37520$ (um conjunto de treino de 150080 amostras contém 37520 vetores de treino de dimensão $K = 4$). Os valores de n_{LBG} utilizados correspondem a 31, 32, 43, 68 e 63 para N igual a 8, 16, 32, 64 e 128, respectivamente.

Ressalte-se que, ao ser utilizado um conjunto de treino relativamente pequeno (correspondente à palavra *aplausos*, contendo 7120 amostras concernentes a 0,89s de voz, que leva a $M = 3560$ para $K = 2$ e $M = 1780$ para $K = 4$), observou-se, também, que as Inequações (5.8), (5.12) e (5.17) de fato podem consideradas em lugar de (5.7), (5.10) e (5.14), respectivamente.

Um outro conjunto de avaliações realizadas neste capítulo teve como objetivo comparar o número de operações (multiplicações, subtrações, adições, divisões e comparações) realizadas pelos algoritmos LBG e SSC quando aplicados ao projeto de dicionários destinados à codificação de forma de onda de voz. As Tabelas 5.8 e 5.9 mostram que, para todos os valores de K e N (com a correspondente taxa de codificação R , expressa em bit/amostra) considerados, o algoritmo SSC realiza um número de operações inferior ao número de operações realizadas pelo algoritmo LBG. Os resultados apresentados nas Tabelas 5.8 e 5.9, obtidos por meio da contabilização do número de operações dos algoritmos SSC e LBG quando aplicados ao projeto de dicionários destinados à quantização vetorial de forma de onda de voz, respaldam as formulações analíticas desenvolvidas neste trabalho.

5.4 Conclusões

Neste capítulo mostrou-se que o algoritmo SSC requer a execução de apenas uma operação de divisão. Mostrou-se também que para dimensão (K) e tamanho (N) de dicionário determinados, o algoritmo SSC é mais eficiente que o algoritmo LBG em termos de número de multiplicações e subtrações caso a condição $n_{\text{SSC}} < \frac{n_{\text{LBG}}}{2}$ seja satisfeita, ou seja, caso o número de iterações do algoritmo SSC seja menor que a metade do número de iterações do algoritmo LBG. Além disso, foi demonstrado que o algoritmo SSC executa um número de adições e comparações inferior ao executado

pelo algoritmo LBG sempre que $n_{SSC} < n_{LBG}$.

Avaliações realizadas neste trabalho, considerando alguns projetos de dicionários aplicados à codificação de forma de onda de voz e à codificação de imagens, mostraram que condição $n_{SSC} < \frac{n_{LBG}}{2}$ em geral é satisfeita e, conseqüentemente, a condição $n_{SSC} < n_{LBG}$ também é satisfeita. O algoritmo SSC, deste modo, requer um número de operações (divisões, multiplicações, subtrações, adições e comparações) inferior ao requerido pelo algoritmo LBG.

Por meio da realização de projeto de dicionários (variando K e N) destinados à QV de forma de onda de voz, os resultados concernentes à contabilização do número de operações executadas pelos algoritmos SSC e LBG respaldaram as conclusões resultantes das expressões analíticas desenvolvidas neste capítulo.

Tabela 5.6: Expressões utilizadas para a avaliação comparativa de complexidade dos algoritmos LBG e SSC para projeto de dicionários aplicados a QV de forma de onda de voz, utilizando dimensão $K = 2$.

N	$\frac{1+KNMn_{\text{LBG}}}{1+(1+N)KM}$	$\frac{N}{1+N}n_{\text{LBG}}$	$\frac{(1+KNM)n_{\text{LBG}}-1}{1+(1+N)KM}$	$\frac{1+[(1+KN)(M-1)+(K-N+1)M]n_{\text{LBG}}}{1+(K-1)NM+KM}$	$\frac{[(K-1)(1+N)+3]}{(K-1)N+K}n_{\text{LBG}}$
8	25,778	25,778	25,778	34,799	34,800
16	41,412	41,412	41,412	48,888	48,889
32	70,788	70,788	70,788	77,292	77,294
64	79,754	79,754	79,754	83,452	83,455
128	178,605	178,605	178,605	182,764	182,769

Tabela 5.7: Expressões utilizadas para a avaliação comparativa de complexidade dos algoritmos LBG e SSC para projeto de dicionários aplicados a QV de forma de onda de voz, utilizando dimensão $K = 4$.

N	$\frac{1+KNMn_{\text{LBG}}}{1+(1+N)KM}$	$\frac{N}{1+N}n_{\text{LBG}}$	$\frac{(1+KNM)n_{\text{LBG}}-1}{1+(1+N)KM}$	$\frac{1+[(1+KN)(M-1)+(K-N+1)M]n_{\text{LBG}}}{1+(K-1)NM+KM}$	$\frac{[(K-1)(1+N)+3]}{(K-1)N+K}n_{\text{LBG}}$
8	27,556	27,556	27,556	33,213	33,214
16	30,118	30,118	30,118	33,230	33,231
32	41,697	41,697	41,697	43,859	43,860
64	66,954	66,954	66,954	68,691	68,694
128	62,512	62,512	62,512	63,323	63,325

Tabela 5.8: Número de operações requeridas pelo algoritmo LBG ao serem projetados dicionários destinados à codificação de voz baseada em QV.

K	N	R	n_{LBG}	mult.	sub.	ad.	div.	comp.
2	8	1,5	29	$3,48 \cdot 10^7$	$3,48 \cdot 10^7$	$2,61 \cdot 10^7$	$4,93 \cdot 10^2$	$1,52 \cdot 10^7$
2	16	2,0	44	$1,06 \cdot 10^8$	$1,06 \cdot 10^8$	$6,60 \cdot 10^7$	$1,45 \cdot 10^3$	$4,95 \cdot 10^7$
2	32	2,5	73	$3,51 \cdot 10^8$	$3,51 \cdot 10^8$	$1,97 \cdot 10^8$	$4,75 \cdot 10^3$	$1,70 \cdot 10^8$
2	64	3,0	81	$7,78 \cdot 10^8$	$7,78 \cdot 10^8$	$4,13 \cdot 10^8$	$1,04 \cdot 10^4$	$3,83 \cdot 10^8$
2	128	3,5	180	$3,46 \cdot 10^9$	$3,46 \cdot 10^9$	$1,78 \cdot 10^9$	$4,63 \cdot 10^4$	$1,72 \cdot 10^9$
4	8	0,75	31	$3,72 \cdot 10^7$	$3,72 \cdot 10^7$	$3,49 \cdot 10^7$	$1,02 \cdot 10^3$	$8,14 \cdot 10^6$
4	16	1,0	32	$7,68 \cdot 10^7$	$7,68 \cdot 10^7$	$6,48 \cdot 10^7$	$2,08 \cdot 10^3$	$1,80 \cdot 10^7$
4	32	1,25	43	$2,07 \cdot 10^8$	$2,07 \cdot 10^8$	$1,65 \cdot 10^8$	$5,55 \cdot 10^3$	$5,00 \cdot 10^7$
4	64	1,5	68	$6,53 \cdot 10^8$	$6,53 \cdot 10^8$	$5,05 \cdot 10^8$	$1,75 \cdot 10^4$	$1,61 \cdot 10^8$
4	128	1,75	63	$1,21 \cdot 10^9$	$1,21 \cdot 10^9$	$9,22 \cdot 10^8$	$3,23 \cdot 10^4$	$3,00 \cdot 10^8$

Tabela 5.9: Número de operações requeridas pelo algoritmo SSC ao serem projetados dicionários destinados à codificação de voz baseada em QV.

K	N	R	n_{SSC}	mult.	sub.	ad.	div.	comp.
2	8	1,5	2	$2,70 \cdot 10^6$	$2,70 \cdot 10^6$	$1,50 \cdot 10^6$	1	$1,05 \cdot 10^6$
2	16	2,0	2	$5,10 \cdot 10^6$	$5,10 \cdot 10^6$	$2,70 \cdot 10^6$	1	$2,25 \cdot 10^6$
2	32	2,5	2	$9,91 \cdot 10^6$	$9,91 \cdot 10^6$	$5,10 \cdot 10^6$	1	$4,65 \cdot 10^6$
2	64	3,0	2	$1,95 \cdot 10^7$	$1,95 \cdot 10^7$	$9,91 \cdot 10^6$	1	$9,46 \cdot 10^6$
2	128	3,5	2	$3,87 \cdot 10^7$	$3,87 \cdot 10^7$	$1,95 \cdot 10^7$	1	$1,91 \cdot 10^7$
4	8	0,75	2	$2,70 \cdot 10^6$	$2,70 \cdot 10^6$	$2,10 \cdot 10^6$	1	$5,25 \cdot 10^5$
4	16	1,0	2	$5,10 \cdot 10^6$	$5,10 \cdot 10^6$	$3,90 \cdot 10^6$	1	$1,13 \cdot 10^6$
4	32	1,25	2	$9,91 \cdot 10^6$	$9,91 \cdot 10^6$	$7,50 \cdot 10^6$	1	$2,33 \cdot 10^6$
4	64	1,5	2	$1,95 \cdot 10^7$	$1,95 \cdot 10^7$	$1,47 \cdot 10^7$	1	$4,73 \cdot 10^6$
4	128	1,75	2	$3,87 \cdot 10^7$	$3,87 \cdot 10^7$	$2,91 \cdot 10^7$	1	$9,53 \cdot 10^6$

Capítulo 6

Análise de Componentes Principais Aplicada à Quantização Vetorial

6.1 Introdução

A análise de componentes principais (PCA, *principal component analysis*) é um método de extração de características ou compressão de dados e tem sido bastante utilizada em diversas aplicações, tais como: reconhecimento de padrões, análise estatística de dados, compressão de voz e imagem, síntese de voz, identificação de sistemas e problemas de visão computacional [201, 202]. A análise de componentes principais, também conhecida como método de Karhunen-Loève, baseia-se na extração dos auto-vetores (componentes principais) e dos autovalores da matriz de covariância dos dados de entrada [2, 201, 203].

Seja \mathbf{x} um vetor K -dimensional. Esse vetor pode ser representado sem erro por uma combinação linear de K vetores independentes \mathbf{z}_i da seguinte forma:

$$\mathbf{x} = \sum_{i=1}^K f_i \mathbf{z}_i = \mathbf{Z}^T \mathbf{f}, \quad (6.1)$$

em que $\mathbf{x}^T = [x_1 x_2 \cdots x_K]$, $\mathbf{f}^T = [f_1 f_2 \cdots f_K]$ e os vetores \mathbf{z}_i são as colunas da matriz $\mathbf{Z}^T = [\mathbf{z}_1 \mathbf{z}_2 \cdots \mathbf{z}_K]$. O superescrito T denota o operador de transposição de matriz.

Os vetores-colunas \mathbf{z}_i , $i = 1, 2, \dots, K$, da matriz \mathbf{Z}^T geram o espaço K -dimensional que contém o vetor \mathbf{x} e podem, portanto, ser chamados vetores de base. Uma vez assegurada independência dos K vetores \mathbf{z}_i , então $\det[\mathbf{Z}^T] \neq 0$, de modo que \mathbf{Z}^T é uma matriz inversível. Admitindo-se a condição de ortonormalidade para os vetores

Uma propriedade interessante dos autovalores é [2]

$$\text{traço}(\mathbf{C}_{xx}) = \sum_{i=1}^K \lambda_i. \quad (6.8)$$

Para processos com média zero, segue a relação

$$\text{traço}(\mathbf{C}_{xx}) = K\sigma_x^2, \quad (6.9)$$

Portanto, para qualquer K , o valor médio dos autovalores é igual à variância do processo, isto é,

$$\sigma_x^2 = \frac{1}{K} \sum_{i=1}^K \lambda_i. \quad (6.10)$$

É importante mencionar que a transformada KLT *diagonaliza* a matriz de correlação \mathbf{R}_{ff} , ou seja, os coeficientes da transformada são descorrelacionados [2]. De fato,

$$\mathbf{R}_{ff} = E[\mathbf{f}\mathbf{f}^T] = E[\mathbf{Z}\mathbf{x}\mathbf{x}^T\mathbf{Z}^T] = \mathbf{Z}\mathbf{R}_{xx}\mathbf{Z}^T. \quad (6.11)$$

Para processos com média zero, $\mathbf{R}_{xx} = \mathbf{C}_{xx}$, portanto,

$$\mathbf{R}_{ff} = \mathbf{Z}\mathbf{C}_{xx}\mathbf{Z}^T. \quad (6.12)$$

A partir da Equação (6.7), segue a relação

$$\mathbf{R}_{ff} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_K^T \end{bmatrix} \begin{bmatrix} \lambda_1 \mathbf{z}_1 & \lambda_2 \mathbf{z}_2 & \cdots & \lambda_K \mathbf{z}_K \end{bmatrix}. \quad (6.13)$$

Assim, a partir da Equação (6.2),

$$\mathbf{R}_{ff} = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \cdots & . \\ 0 & & \lambda_3 & \cdots & . \\ \vdots & & & & \\ 0 & . & . & \cdots & \lambda_K \end{bmatrix}. \quad (6.14)$$

A transformada de Karhunen-Loève (também conhecida como transformada Hotelling), portanto, é uma transformação ortogonal que reduz um grande conjunto de variáveis correlacionadas a um menor conjunto de componentes descorrelacionadas [2, 204].

6.2 Projeto de Dicionários Usando Análise de Componentes Principais

No método de projeto de dicionários apresentado no presente trabalho, os autovetores (componentes principais) são usados para encontrar $L \leq K$ direções (no espaço R^K) ao longo das quais os vetores do dicionário (vetores de reconstrução, vetores-código) são alocados. Os autovalores correspondentes a esses autovetores são usados para ajustar adequadamente as posições dos vetores-código ao longo das direções principais. É válido mencionar que, de acordo com [2], os vetores de base da KLT (autovetores) apresentam a maior semelhança possível com os segmentos de forma de onda típicos do sinal de entrada cuja estatística foi utilizada para determinação da transformada KLT.

Descrição do Algoritmo

O algoritmo proposto para projeto de dicionário consiste na seguinte seqüência de passos:

1. Defina a dimensão K e o tamanho do dicionário N ;
2. A partir da matriz de covariância C_{xx} de ordem K de um sinal típico de voz X , determine os autovalores λ_i e os autovetores z_i ; $i = 1, 2, \dots, K$, obtidos por

$$C_{xx}z_i = \lambda_i z_i; \quad (6.15)$$

3. Defina $L \leq K$ vetores z_i (componentes principais) ao longo de cujas direções os vetores-código devem ser alocados. O número L é escolhido de acordo com o valor percentual relativo $\hat{\lambda}_i$ de cada autovalor λ_i :

$$\hat{\lambda}_i = \frac{\lambda_i}{\sum_{i=1}^K \lambda_i}. \quad (6.16)$$

Apenas os L autovetores mais significativos, os quais definem as L direções principais, são escolhidos, de acordo com os valores mais significativos de $\hat{\lambda}_i$;

4. Para cada z_i escolhido, determine um vetor $\tilde{z}_i = r_i z_i$ ($i = 1, 2, \dots, L$), em que o escalar r_i é o recíproco do valor absoluto da componente de maior valor absoluto. O vetor \tilde{z}_i , portanto, tem no mínimo uma componente com valor absoluto 1 e as demais componentes apresentam valores absolutos situados na faixa de 0 a 1;

5. Seja N_i o número de vetores-código a serem alocados ao longo da direção definida pelo vetor \tilde{z}_i . Cada N_i é escolhido como uma fração de N , proporcionalmente a $\hat{\lambda}_i$, de modo que

$$N = \sum_{i=1}^L N_i; \quad (6.17)$$

6. Finalmente, N_i vetores-código w_{i,n_i} são alocados em cada i -ésima direção principal de acordo com

$$w_{i,n_i} = f(n_i, \lambda_i) \tilde{z}_i, \quad (6.18)$$

em que $i = 1, 2, \dots, L$, $n_i = 1, 2, \dots, N_i$ e os escalares $f(n_i, \lambda_i)$ são determinados assumindo-se que cada i -ésima direção principal está relacionada a um conjunto de vetores de entrada cujas componentes têm distribuição gaussiana com média igual à média do sinal de voz e variância $\sigma_i^2 = \lambda_i$. Os escalares $f(n_i, \lambda_i)$, $n_i = 1, 2, \dots, N_i$, são determinados de tal forma que a área sob a função densidade de probabilidades gaussiana seja igualmente dividida em N_i intervalos.

Em suma, um conjunto de L autovetores é determinado e um número de vetores-código N_i (uma fração de N) é então alocado em cada direção principal definida pelos autovetores z_i ($i = 1, 2, \dots, L$) de acordo com considerações efetuadas nos autovalores (variâncias).

O método proposto difere do algoritmo descrito em [193] uma vez que introduz uma espécie de normalização no conjunto de autovetores, conforme descrito no passo 4. A multiplicação dos autovetores normalizados \tilde{z}_i pelos escalares $f(n_i, \lambda_i)$ permite que o dicionário resultante (dicionário projetado) incorpore simultaneamente duas características importantes do sinal de voz: as direções principais (via autovetores) e a distribuição estatística dos vetores (via autovalores) em cada direção.

6.2.1 Resultados

Nesta seção, a técnica de projeto de dicionários baseada em análise de componentes principais é denotada por PCA, enquanto que sua versão original, proposta em [193], é denotada por PCA-ORI. A aquisição (resolução 8,0 bit/amostra e taxa de amostragem 8 kHz) dos sinais de voz utilizados nas simulações foi realizada usando uma estação de trabalho Sun[®], instalada com utilitários de processamento de áudio. Para avaliação dos dicionários projetados, utilizou-se um conjunto de teste relativamente longo, cons-

tituído de 29120 amostras (3,64 s), correspondente ao conjunto de sentenças “*O sol ilumina a fachada de tarde. Trabalhou mais do que podia*”.

O primeiro experimento teve como objetivo avaliar a coerência do algoritmo PCA. Para tanto, obteve-se a representação, no espaço de padrões euclidiano bidimensional, de um sinal de voz (cujo histograma é apresentado na Figura 6.1), correspondente a 10 frases foneticamente balanceadas (extraídas de [190] e pronunciadas por 10 locutores diferentes, sendo 5 masculinos e 5 femininos), apresentado na Figura 6.2, e de um dicionário obtido com o algoritmo PCA, ilustrado na Figura 6.3. Conforme se pode observar, o dicionário projetado apresenta algumas características típicas dos sinais de voz, tais como predominância de vetores com componentes de baixa amplitude (um maior número de vetores-código é alocado nas regiões de maior densidade de vetores da Figura 6.2) e correlação entre amostras consecutivas (os vetores-código estão alocados, de forma coerente, ao longo da direção correspondente à componente principal do sinal).

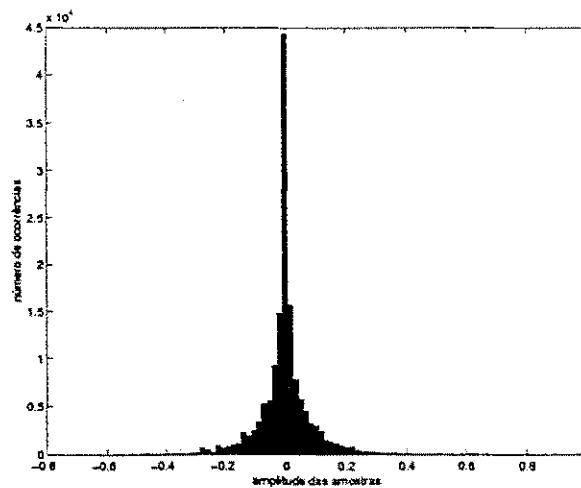


Figura 6.1: Histograma de um sinal de voz.

O segundo experimento contemplou a realização de um estudo comparativo de desempenho dos algoritmos PCA e PCA-ORI. As Figuras 6.4 e 6.5 apresentam uma avaliação em termos de relação sinal-ruído segmental (SNRseg) e relação sinal-ruído total (SNRtot) do sinal de voz reconstruído para diversas taxas de codificação considerando QV com dimensão $K = 2$ e $K = 4$, respectivamente.

Conforme mostra a Figura 6.4(a), fixada a dimensão $K = 2$, à exceção da taxa de 1,5 bit/amostra, o algoritmo PCA leva a maiores valores de SNRseg que os obtidos

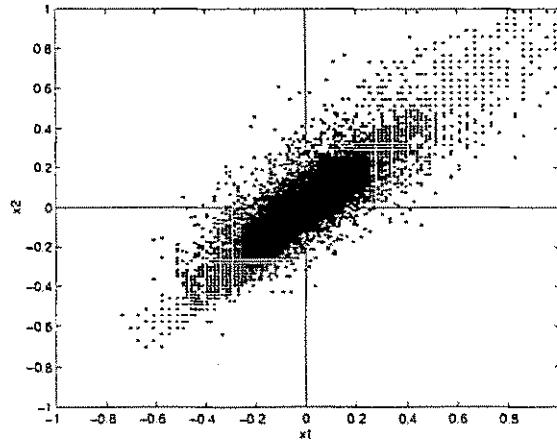


Figura 6.2: Sinal de voz consistindo de 10 frases foneticamente balanceadas (18,76s, 75040 vetores). As coordenadas x_1 e x_2 representam a primeira e a segunda componentes dos vetores de treino $\mathbf{x} \in R^2$, respectivamente.

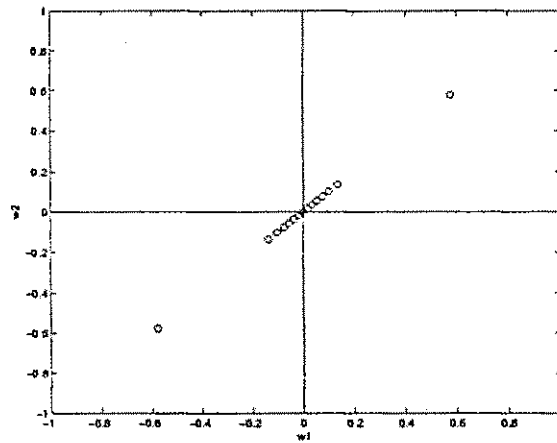
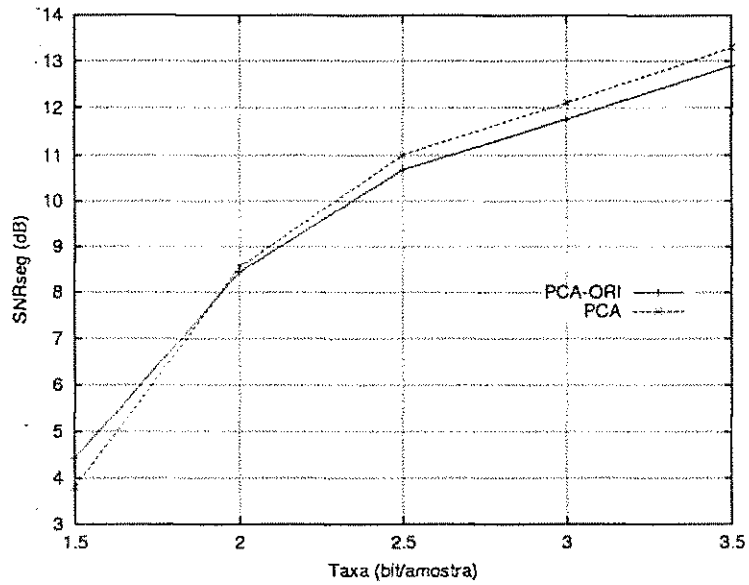
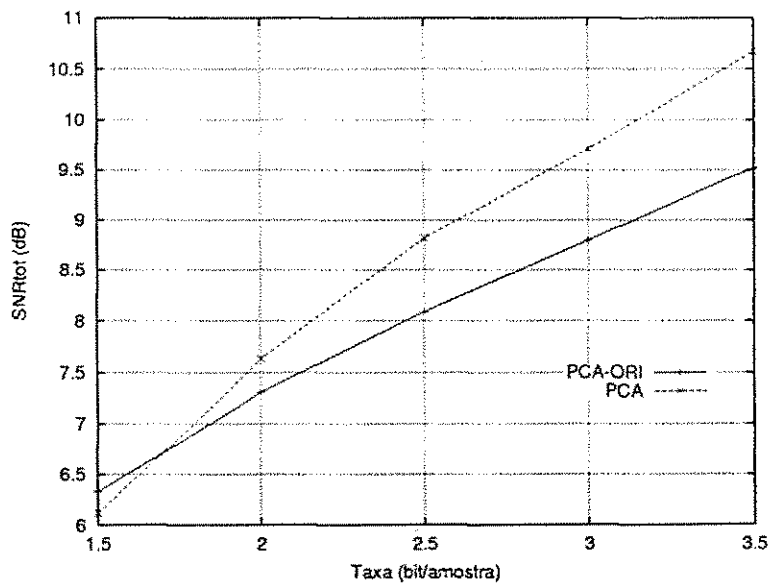


Figura 6.3: Dicionário obtido com o algoritmo PCA: $K = 2$ e $N = 16$. As coordenadas w_1 e w_2 representam, respectivamente, a primeira e a segunda componentes dos vetores-código $\mathbf{w}_i \in R^2$, $i = 1, 2, \dots, 16$.

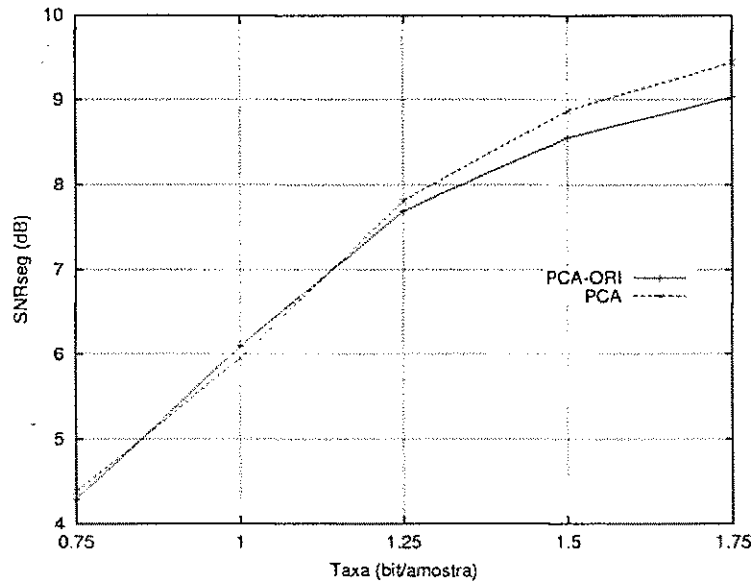


(a) SNRseg

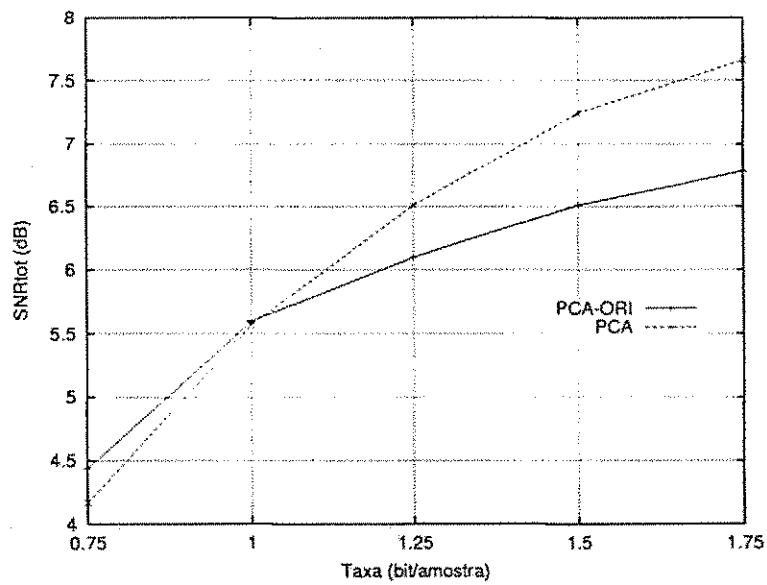


(b) SNRtot

Figura 6.4: Desempenho dos algoritmos PCA e PCA-ORI em termos de SNRseg e SNRtot do sinal reconstruído para diversas taxas em QV com $K = 2$.

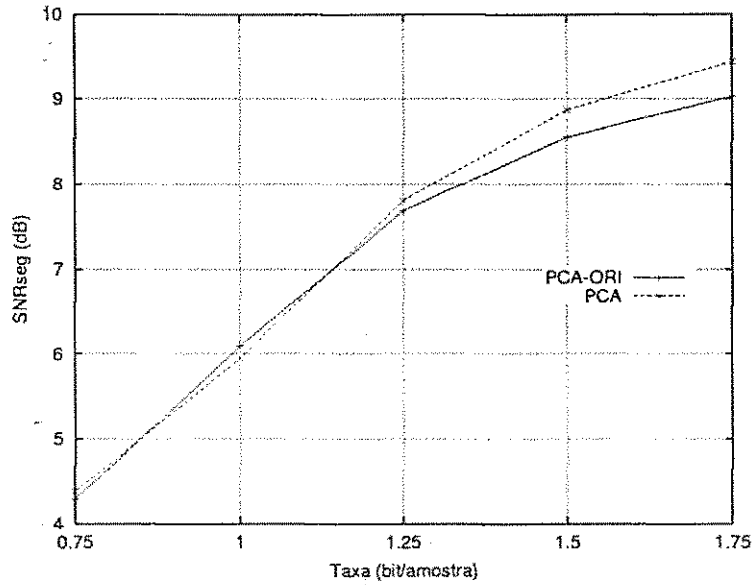


(a) SNRseg

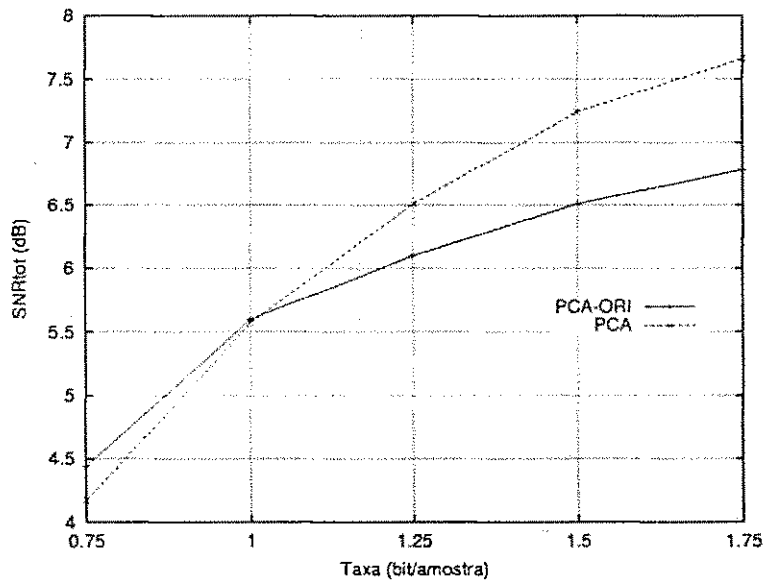


(b) SNRtot

Figura 6.5: Desempenho dos algoritmos PCA e PCA-ORI em termos de SNRseg e SNRtot do sinal reconstruído para diversas taxas em QV com $K = 4$.



(a) SNRseg



(b) SNRtot

Figura 6.5: Desempenho dos algoritmos PCA e PCA-ORI em termos de SNRseg e SNRtot do sinal reconstruído para diversas taxas em QV com $K = 4$.

com o algoritmo PCA-ORI. Para taxas de codificação na faixa de 3,0 bit/amostra a 3,5 bit/amostra, a curva de desempenho do algoritmo PCA situa-se cerca de 0,4 dB acima da curva de desempenho do algoritmo PCA-ORI. Em se tratando de SNR_{tot}, a Figura 6.4(b) mostra que, à exceção da taxa de 1,5 bit/amostra, o algoritmo PCA apresenta desempenho superior ao apresentado pelo algoritmo PCA-ORI. Observa-se que a superioridade dos dicionários PCA sobre os dicionários PCA-ORI aumenta com a taxa de codificação. Para as taxas de 3,0 bit/amostra e 3,5 bit/amostra, o algoritmo PCA supera o algoritmo PCA-ORI em cerca de 0,9 dB e 1,2 dB, respectivamente.

No que diz respeito à curva de SNR_{seg} versus taxa de codificação para dimensão $K = 4$, apresentada na Figura 6.5(a), para taxas compreendidas entre 0,75 bit/amostra e 1,25 bit/amostra, os algoritmos PCA e PCA-ORI apresentam desempenho semelhante. De fato, para essa faixa de taxas, a diferença de desempenho dos algoritmos PCA e PCA-ORI é no máximo cerca de 0,15 dB. Para as taxas de 1,5 bit/amostra e 1,75 bit/amostra, obtêm-se, respectivamente, ganhos de 0,3 dB e 0,4 dB ao serem utilizados dicionários PCA em substituição aos dicionários PCA-ORI no processo de quantização vetorial. Quanto à relação sinal-ruído total para dimensão $K = 4$, a Figura 6.5(b) mostra que, à exceção da taxa 0,75 bit/amostra, o algoritmo PCA apresenta-se como a melhor alternativa. Observa-se que a superioridade do algoritmo PCA sobre o algoritmo PCA-ORI aumenta com a taxa de codificação. Para as taxas de 1,5 bit/amostra e 1,75 bit/amostra, o algoritmo PCA supera o algoritmo PCA-ORI em cerca de 0,7 dB e 0,9 dB, respectivamente.

Os resultados apresentados nas Figuras 6.4 e 6.5 mostram que a superioridade do algoritmo PCA sobre o algoritmo PCA-ORI manifesta-se mais acentuadamente em termos de SNR_{tot} que de SNR_{seg}. Isto significa que o algoritmo PCA é mais eficiente que o algoritmo PCA-ORI principalmente no fato de levar a uma representação mais adequada das amostras de alta amplitude do sinal de voz.

Uma avaliação comparativa de desempenho dos algoritmos PCA, SOA, SSC e LBG pode ser realizada observando-se as Figuras 6.6 e 6.7, que apresentam curvas de relação sinal-ruído segmental do sinal de voz reconstruído versus taxa de codificação para QV com dimensão $K = 2$ e $K = 4$, respectivamente.

A Figura 6.6 mostra que o algoritmo PCA apresenta desempenho superior ao apresentado pelo algoritmo LBG para as taxas de 2,0 bit/amostra e 2,5 bit/amostra. Para essas taxas de codificação do quantizador vetorial com dimensão $K = 2$, a curva de desempenho do algoritmo PCA é próxima da curva de desempenho dos algoritmos

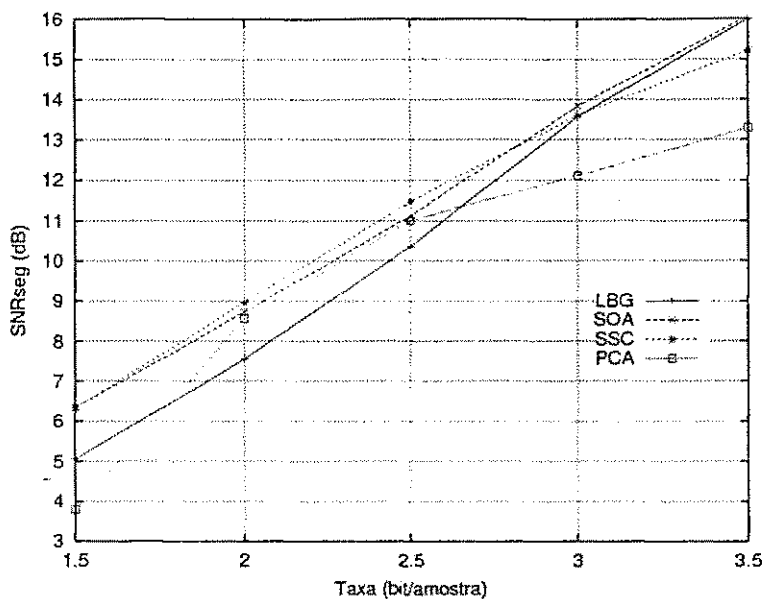


Figura 6.6: Desempenho dos algoritmos PCA, SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com $K = 2$.

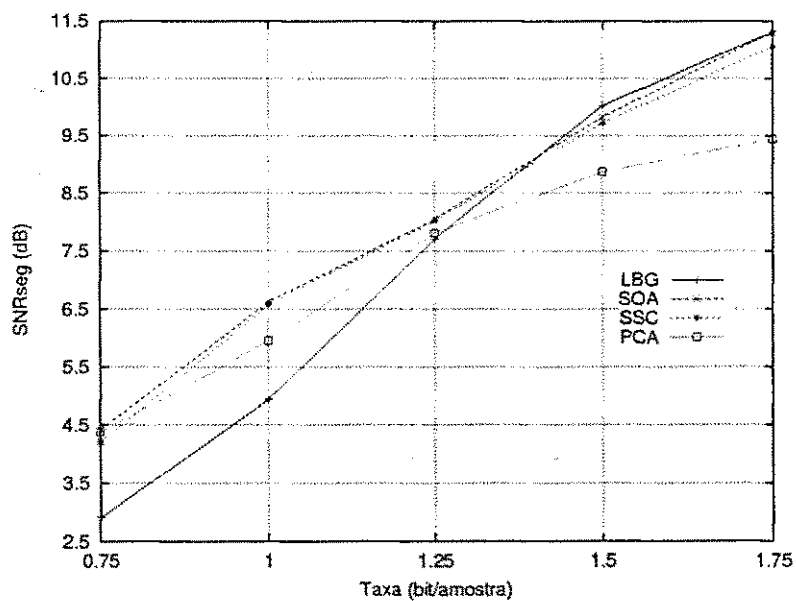


Figura 6.7: Desempenho dos algoritmos PCA, SOA, SSC e LBG em QV de forma de onda de voz (domínio temporal): SNRseg do sinal reconstruído versus taxa de codificação para o quantizador vetorial com $K = 4$.

SOA e SSC. Para as taxas de 3,0 bit/amostra e 3,5 bit/amostra, bem como para a taxa de 1,5 bit/amostra, os dicionários LBG (como também os dicionários SOA e SSC) levam a sinais reconstruídos com maiores valores de SNRseg que os obtidos com uso de dicionários PCA.

Conforme se observa na Figura 6.7, para taxas situadas entre 0,75 bit/amostra e 1,25 bit/amostra, o algoritmo PCA apresenta-se como uma alternativa mais adequada que o algoritmo LBG: para as taxas de 0,75 bit/amostra e 1,0 bit/amostra, os dicionários PCA superam os dicionários LBG em cerca de 1,5 dB e 1,0 dB, respectivamente. Para as taxas de 0,75 bit/amostra e 1,25 bit/amostra, o algoritmo PCA apresenta desempenho semelhante ao apresentado pelos algoritmos SOA e SSC. Para taxas acima de 1,25 bit/amostra, os dicionários PCA apresentam-se como os menos eficientes dentre todos os dicionários considerados.

No que diz respeito à realização de testes de escuta informais, observou-se, para todas as taxas de codificação consideradas do quantizador vetorial com dimensão $K = 2$, que os sinais de voz reconstruídos utilizando-se dicionários PCA apresentam qualidade superior (notadamente em termos da percepção do ruído de quantização ou ruído de fundo) à apresentada pelos sinais reconstruídos com uso de dicionários LBG e SOA. No entanto, para elevadas taxas de codificação, ao se usar dimensão $K = 2$, a superioridade dos dicionários PCA sobre os dicionários LBG e SOA, em termos de qualidade subjetiva do sinal reconstruído, torna-se menos acentuada. Os testes subjetivos também mostraram que, para QV com dimensão $K = 4$, os dicionários PCA levam a sinais reconstruídos acometidos da incômoda presença de um “sinal artificial”, como um “tom musical” (um “apito”), o qual é mais facilmente percebido nos segmentos de silêncio e que não apresenta qualquer semelhança com o sinal de voz nem mesmo com o ruído de quantização convencional. Esse efeito, denotado na literatura como *ruído musical* ou *ruído tonal* [205,206], é produzido quando as componentes espectrais nas regiões de maior concentração de amostras de baixas amplitudes não são adequadamente reconstruídas. Esse fenômeno pode manifestar-se sob a forma de “buracos espectrais” no espectrograma do sinal reconstruído [207,208]. Nesse contexto, é importante mencionar que foram observados padrões atípicos nos espectrogramas dos sinais reconstruídos com uso de dicionários PCA com dimensão $K = 4$. Ressalte-se que, para a dimensão $K = 4$, os sinais reconstruídos com dicionários PCA apresentam ruído de quantização convencional menos incômodo que o presente nos sinais reconstruídos com uso de dicionários LBG e SOA. Esse resultado da avaliação subjetiva também

foi confirmado por meio de avaliação objetiva. De fato, as Tabelas 6.1 e 6.2 apontam um aspecto interessante do algoritmo PCA: ao contrário do que tipicamente ocorre ao serem utilizados dicionários LBG (como também SOA), os dicionários PCA levam a sinais reconstruídos com SNRseg superior a SNRtot. Esse comportamento foi observado em quase todas taxas de codificação avaliadas para as dimensões $K = 2$, $K = 4$ e $K = 8$ e mostra claramente que os dicionários PCA tendem a preservar mais fielmente os segmentos de voz de baixa energia quando comparados aos dicionários LBG e SOA.

Para QV com dimensão $K = 8$, os testes de escuta também apontaram a presença de *ruído musical* nos sinais reconstruídos com uso de dicionários PCA.

As sessões de escuta levaram a uma observação interessante: à medida que a dimensão do dicionário projetado se afasta de 2, o algoritmo PCA tende a não mais se apresentar como uma boa técnica de projeto de dicionários.

Tabela 6.1: SNRseg e SNRtot dos sinais reconstruídos utilizando-se dicionários PCA, para diversos valores de K e N .

K	N	SNRseg (dB)	SNRtot (dB)
2	32	11,01	8,82
2	64	12,11	9,71
2	128	13,29	10,67
4	32	7,81	6,51
4	64	8,87	7,24
4	128	9,44	7,66

Foram realizadas simulações visando avaliar a utilização dos dicionários PCA como dicionários iniciais do algoritmo LBG. A Tabela 6.3 mostra que os dicionários PCA podem ser usados como uma boa alternativa de inicialização do algoritmo LBG, visto que podem levar a um bom desempenho em termos de velocidade de convergência do algoritmo LBG. Para o projeto de dicionários com $K = 4$ e $N = 128$, a aplicação da inicialização PCA (isto é, utilizando-se D_{PCA} como dicionário inicial) leva a 23 iterações do algoritmo LBG, o que representa uma maior velocidade de convergência quando comparada à inicialização D_{II} , que requer 82 iterações do algoritmo LBG. Apesar de a inicialização D_{PCA} poder levar a um ganho em termos de SNRseg do sinal reconstruído, quando comparada às outras escolhas de dicionário inicial consideradas,

Tabela 6.2: SNRseg e SNRtot dos sinais reconstruídos utilizando-se dicionários LBG, para diversos valores de K e N .

K	N	SNRseg (dB)	SNRtot (dB)
2	32	10,36	13,39
2	64	13,57	14,40
2	128	15,99	15,23
4	32	7,71	9,81
4	64	10,02	10,52
4	128	11,31	11,49

constatou-se, nas sessões de escuta informais, que a inicialização D_{PCA} pode resultar em sinais reconstruídos com qualidade inferior à obtida utilizando-se as inicializações D_I , D_{II} e D_{III} , notadamente para as dimensões $K = 4$ e $K = 8$.

Um aspecto interessante dos dicionários PCA diz respeito à homogeneidade da distribuição dos vetores de entrada (vetores da fonte a ser quantizada) nas células de Voronoi. Os valores de entropia normalizada (vide Apêndice A para maiores detalhes) apresentados nas Tabelas 6.4 e 6.5 mostram que os dicionários PCA levam a uma distribuição mais homogênea dos vetores da fonte nas regiões de Voronoi, quando comparados aos dicionários SOA. De fato, os dicionários PCA levam a maiores valores de entropia normalizada.

A Figura 6.8 mostra que o algoritmo PCA gera dicionários estruturados, ao contrário do algoritmo LBG (como também dos algoritmos SOA¹ e SSC), que gera dicionários sem estrutura, conforme ilustra a Figura 6.9.

Comparando as Figuras 6.8 e 6.10, observa-se que o dicionário PCA leva a uma melhor representação das componentes de alta amplitude do sinal de voz quando comparado ao dicionário PCA-ORI. De fato, o dicionário PCA apresenta uma grande concentração de vetores cujas componentes apresentam valor absoluto inferior a 0,2, além de apresentar dois vetores cujas componentes têm valor absoluto de aproximadamente 0,6. O dicionário PCA-ORI, por sua vez, apresenta concentração de vetores cujas componentes têm valor absoluto inferior a cerca de 0,1, apresentando, também,

¹Em [194] é apresentada uma representação do dicionário projetado com o algoritmo SOA, para $K = 5$ e $N = 76$.

Tabela 6.3: Sensibilidade do algoritmo LBG a quatro dicionários iniciais diferentes (D_I , D_{II} , D_{III} e D_{PCA}) em termos de número total de iterações para diversos valores de tamanho do dicionário (N) e dimensão (K). O subscrito PCA denota a inicialização do algoritmo LBG usando dicionários PCA.

K	N	Número de iterações			
		D_I	D_{II}	D_{III}	D_{PCA}
2	32	112	53	53	69
2	64	140	75	119	49
2	128	123	78	96	51
4	32	59	62	52	53
4	64	72	85	59	39
4	128	77	82	39	23
8	32	38	26	36	26
8	64	62	50	26	29
8	128	30	29	34	13

Tabela 6.4: Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) e dimensão (K) para os dicionários PCA.

K	N	\tilde{H}
2	32	0,94
2	64	0,93
4	32	0,94
4	64	0,95
8	32	0,96
8	64	0,95

Tabela 6.5: Entropia normalizada (\tilde{H}) dos vetores-código para diversos valores de número de níveis (N) e dimensão (K) para os dicionários SOA.

K	N	\tilde{H}
2	32	0,88
2	64	0,91
4	32	0,87
4	64	0,88
8	32	0,84
8	64	0,89

dois vetores cujas componentes têm módulo aproximadamente igual a 0,35. Em virtude de representarem adequadamente as amostras de alta amplitude dos sinais de voz, os dicionários PCA levam a melhores resultados em termos de SNR_{tot} dos sinais reconstruídos quando comparados aos resultados concernentes ao uso de dicionários PCA-ORI, conforme mostram os valores obtidos no segundo experimento relatado no início desta seção.

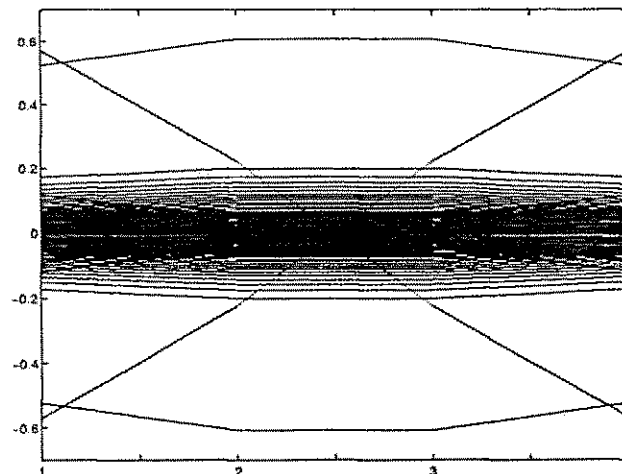


Figura 6.8: Dicionário projetado com o algoritmo PCA: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.

Dentre as principais características do algoritmo PCA, podem ser destacadas:

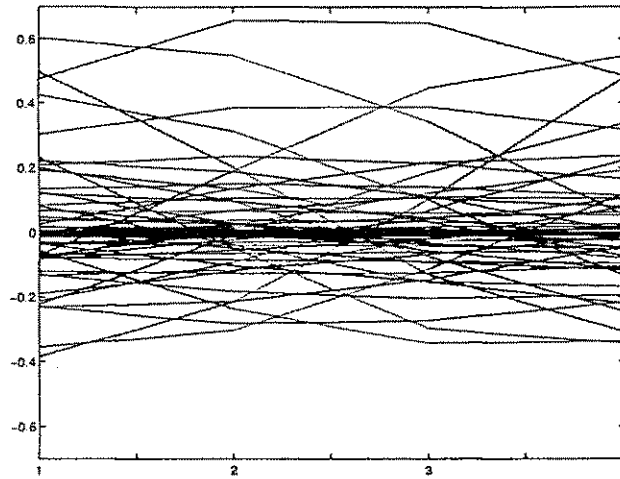


Figura 6.9: Dicionário projetado com o algoritmo LBG: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.

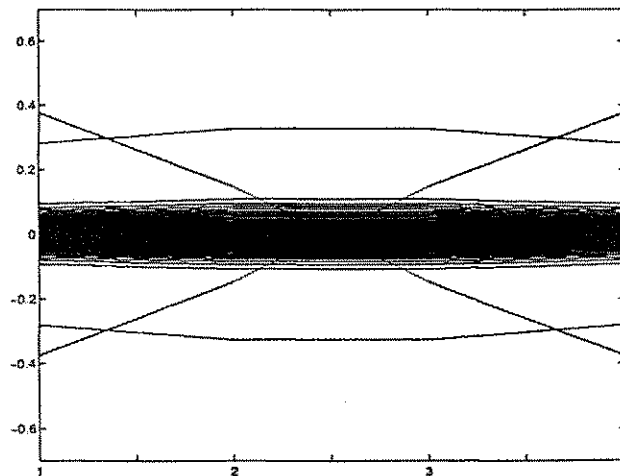


Figura 6.10: Dicionário projetado com o algoritmo PCA-ORI: $K = 4$ e $N = 64$. Cada curva do conjunto de 64 curvas é obtida conectando-se os pontos correspondentes às componentes (amostras) dos vetores-código.

- simplicidade e rapidez;
- geração de dicionários estruturados;
- inexistência de parâmetros a serem ajustados;
- independência de dicionário inicial.

De fato, o algoritmo PCA é bastante simples e intuitivo. Os vetores-código são “calculados”, de acordo com a estatística do sinal (de acordo com os autovalores e autovetores da matriz de covariância de um sinal de voz), diferindo consideravelmente de abordagens tradicionais, tais como os algoritmos LBG e de Kohonen, que necessitam de um conjunto de treino para atualizar iterativamente os vetores-código. O algoritmo PCA também não apresenta o problema de ajuste de parâmetros e não requer a definição de um dicionário inicial, como ocorre nos algoritmos SSC, SOA e de Kohonen. Além disso, testes subjetivos informais mostraram que, em se tratando de quantização vetorial de forma de onda com dimensão $K = 2$, os sinais de voz reconstruídos utilizando-se dicionários PCA apresentam qualidade superior à apresentada pelos sinais reconstruídos com uso de dicionários LBG e SOA.

6.3 Um Método para Redução da Complexidade Computacional da Etapa de Codificação da QV

Na seção anterior, mostrou-se que uma das características dos dicionários PCA é o fato de serem estruturados, ou seja, terem uma disposição bastante organizada no espaço de padrões. Nesta seção, a organização estruturada dos dicionários PCA é utilizada para minimizar os requisitos de capacidade de memória exigida para armazenar o dicionário bem como para reduzir significativamente a complexidade computacional da etapa de codificação da QV (no processo de determinação do vizinho mais próximo).

O projeto de dicionários PCA pode ser realizado considerando uma distribuição gaussiana com média zero (no passo 6 do algoritmo PCA) e impondo a condição de que o número de vetores-código alocados em cada direção principal seja par. Obedecendo a essas condições, as seguintes características podem então ser atribuídas a um dicionário PCA:

1. Ordenamento dos N vetores-código de acordo com a Figura 6.11, em que N_{POS} , N_{ZERO} e N_{NEG} denotam o número de vetores-código com média² positiva, o número de vetores-código com média zero e o número de vetores-código com média negativa, respectivamente. Observa-se na Figura que N_{ZERO} representa uma pequena parcela do total de $N = N_{\text{POS}} + N_{\text{ZERO}} + N_{\text{NEG}}$ vetores-código do dicionário. Em alguns casos, $N_{\text{ZERO}} = 0$;
2. Disposição dos vetores-código em ordem decrescente de valores médios das componentes;
3. Simetria marcante, conforme mostra a Tabela 6.6: associado a um vetor-código w_i , $1 \leq i \leq N/2$, existe um vetor-código w_{N+1-i} tal que $w_{N+1-i} = -w_i$.

Os primeiros $N/2$ vetores-código do dicionário, portanto, têm componentes cujo valor médio é positivo ou zero. Esses vetores serão denotados a partir de agora por $w_{i,\text{PZ}}$, com $1 \leq i \leq N/2$, em que o subscrito PZ é uma alusão ao valor médio positivo ou zero. Os últimos $N/2$ vetores-código do dicionário, por outro lado, têm componentes cujo valor médio é negativo ou zero. Esses vetores serão denotados a partir de agora por $w_{i,\text{NZ}}$, com $N/2 + 1 \leq i \leq N$, em que o subscrito NZ é uma alusão ao valor médio negativo ou zero. Além disso, em decorrência da simetria atribuída aos dicionários PCA, associado a cada vetor-código $w_{i,\text{PZ}}$ existe um correspondente vetor-código $w_{N+1-i,\text{NZ}}$ tal que $w_{i,\text{PZ}} = -w_{N+1-i,\text{NZ}}$, $1 \leq i \leq N/2$.

Em virtude dessa simetria, é necessário armazenar apenas metade dos vetores-código. De fato, armazenando-se apenas os vetores-código $w_{i,\text{PZ}}$, os vetores-código $w_{i,\text{NZ}}$ são facilmente determinados. Com isso, reduz-se pela metade os requisitos de memória de armazenamento para o dicionário. A simetria dos dicionários PCA também contribui para a concepção de uma estratégia para redução da complexidade computacional da etapa de codificação (determinação do vizinho mais próximo) da QV, conforme mostra a seção a seguir. No método descrito no presente trabalho, apenas a metade do dicionário correspondente aos vetores $w_{i,\text{PZ}}$ é efetivamente armazenada na memória de referência do codificador. O decodificador, por outro lado, dispõe dos N vetores-código do dicionário.

²Ao longo deste capítulo, a média de um vetor deve ser entendida como a média aritmética de suas componentes.

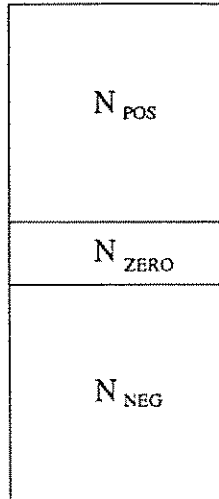


Figura 6.11: Organização de um dicionário PCA com $N = N_{POS} + N_{ZERO} + N_{NEG}$ vetores-código.

6.3.1 Algoritmo para Redução da Complexidade Computacional da Etapa de Codificação da QV

Com o método de codificação introduzido no presente trabalho, a codificação de um determinado vetor de entrada (vetor da fonte a ser quantizada) \mathbf{x} é realizada da seguinte forma: dado \mathbf{x} , se $\text{média}(\mathbf{x}) \geq 0$ então³ o codificador realiza uma busca pelo vizinho mais próximo de \mathbf{x} apenas nos vetores-código $\mathbf{w}_{i,PZ}$, isto é, nos vetores-código efetivamente armazenados na memória de referência do codificador. Neste caso, é transmitida para o decodificador uma palavra-binária iniciada por 0, indicando para o decodificador que o vetor-código a ser produzido como representante de \mathbf{x} trata-se de um vetor do tipo $\mathbf{w}_{i,PZ}$, seguida de $\log_2(N/2)$ bits necessários para representar o índice i do vetor $\mathbf{w}_{i,PZ}$ selecionado da memória de referência. Por outro lado, se $\text{média}(\mathbf{x}) < 0$, a busca pelo vizinho mais próximo deveria contemplar uma procura dentre os vetores-código $\mathbf{w}_{i,NZ}$. Como estes vetores não se encontram armazenados na memória de referência do codificador, a codificação consiste em comparar o vetor $-\mathbf{x}$ (vetor simétrico de \mathbf{x}) com os vetores-código $\mathbf{w}_{i,PZ}$. Uma vez determinado o melhor representante para $-\mathbf{x}$, é transmitida uma palavra-binária iniciada por 1, indicando que o vetor-código a ser produzido pelo decodificador como representante de \mathbf{x} trata-

³Para um vetor $\mathbf{x} = [x_1 x_2 \cdots x_K]^T$, em que T denota a operação de transposição, $\text{média}(\mathbf{x}) = \frac{1}{K} \sum_{j=1}^K x_j$.

Tabela 6.6: Dicionário PCA com 16 vetores-código bidimensionais $w_i = [w_{i1} \ w_{i2}]^T$, com $1 \leq i \leq 16$. A palavra-binária do i -ésimo vetor-código, w_i , é denotada por b_i , enquanto que w_{ij} representa a j -ésima componente do vetor w_i .

i	w_{i1}	w_{i2}	b_i
1	0,5784	0,5784	0000
2	0,1365	0,1365	0001
3	0,1020	0,1020	0010
4	0,0765	0,0765	0011
5	0,0550	0,0550	0100
6	0,0357	0,0357	0101
7	0,0176	0,0176	0110
8	0,0010	0,0010	0111
9	-0,0010	-0,0010	1000
10	-0,0176	-0,0176	1001
11	-0,0357	-0,0357	1010
12	-0,0550	-0,0550	1011
13	-0,0765	-0,0765	1100
14	-0,1020	-0,1020	1101
15	-0,1365	-0,1365	1110
16	-0,5784	-0,5784	1111

se de um vetor do tipo $w_{i,NZ}$, seguida de uma seqüência de $\log_2(N/2)$ bits: cada bit dessa seqüência é o complemento do bit correspondente na seqüência de $\log_2(N/2)$ bits necessários para representar o índice i do vetor $w_{i,PZ}$ selecionado como o mais próximo de $-x$ segundo um critério de distorção mínima. É importante observar que, devido à simetria/organização do dicionário PCA, o fato de x ter como vizinho mais próximo um determinado vetor-código $w_{N+1-i,NZ}$ implica dizer que o simétrico $-x$ tem como vizinho mais próximo $w_{i,PZ} = -w_{N+1-i,NZ}$.

Para ilustrar o algoritmo de codificação proposto neste trabalho, considere o dicionário PCA da Tabela 6.6, que corresponde à tabela disponível para o decodificador. Já o codificador dispõe da Tabela 6.7, correspondente aos primeiros $N/2$ vetores-código, do tipo $w_{i,PZ}$, da Tabela 6.6. Suponha que o sistema de comunicação receba como en-

trada o vetor $\mathbf{x} = [0,0757 \ 0,0792]^T$. Após a avaliação da média⁴ das componentes desse vetor, o codificador decide que o vetor-código a ser selecionado como o representante (como a versão quantizada) de \mathbf{x} trata-se de um vetor do tipo $w_{i,PZ}$. Determina-se, assim, que o primeiro bit da palavra-binária transmitida para o decodificador é 0. A busca do vizinho mais próximo é então realizada no dicionário efetivamente armazenado (Tabela 6.7) na memória de referência do codificador. Seguindo o critério de distorção mínima o vetor $[0,0765 \ 0,0765]^T$, com representação binária 011, é selecionado. O codificador, portanto, transmite para o decodificador a palavra-binária 0011, em que o primeiro bit informa que a versão quantizada de \mathbf{x} é do tipo $w_{i,PZ}$ (isto é, um dos primeiro $N/2$ vetores do dicionário de N vetores do decodificador) e os três últimos bits são uma representação binária do índice do vetor selecionado (da tabela do codificador) como vizinho mais próximo de \mathbf{x} . No outro lado do sistema de comunicação baseado em QV, o decodificador, que dispõe do dicionário da Tabela 6.6, ao receber a palavra-binária 0011, produz como saída o vetor $[0,0765 \ 0,0765]^T$.

Suponha, agora, que o vetor de entrada seja $\mathbf{x} = [-0,4795 \ -0,4957]^T$. Após avaliar a média desse vetor, o codificador decide que a versão quantizada de \mathbf{x} trata-se de um vetor do tipo $w_{i,NZ}$. Determina-se, assim, que o primeiro bit da palavra-binária transmitida para o decodificador é 1. É realizada uma busca pelo vizinho mais próximo de $-\mathbf{x}$ dentre os $N/2$ vetores-código disponíveis (Tabela 6.7) para o codificador: é selecionado o vetor-código $[0,5784 \ 0,5784]^T$ por ser o mais próximo (semelhante) a $[0,4795 \ 0,4957]^T = -\mathbf{x}$. O codificador transmite para o decodificador a palavra-binária 1111, em que o primeiro bit informa que o vetor-código selecionado como a versão quantizada de \mathbf{x} é do tipo $w_{i,NZ}$ e os três últimos bits correspondem ao complemento da palavra-binária 000 que identifica o vetor $[0,5784 \ 0,5784]^T$ no dicionário efetivamente disponível ao codificador. No outro lado do sistema de comunicação, o decodificador, ao receber a palavra-binária 1111, produz como saída o vetor $[-0,5784 \ -0,5784]^T$.

Número de Operações do Método Proposto

O método de codificação proposto no presente trabalho será denotado a partir de agora por 1/2PCA, em virtude de o codificador utilizar apenas metade do dicionário

⁴Tendo em vista que soma positiva de todas as componentes de um vetor implica média positiva das componentes, em lugar de determinar a média das componentes do vetor o método de codificação proposto calcula apenas a soma das componentes. Economiza-se, assim, uma operação de divisão para cada vetor de entrada.

Tabela 6.7: Vetores-código bidimensionais efetivamente armazenados na memória de referência do codificador.

i	w_{i1}	w_{i2}	Representação binária
1	0,5784	0,5784	000
2	0,1365	0,1365	001
3	0,1020	0,1020	010
4	0,0765	0,0765	011
5	0,0550	0,0550	100
6	0,0357	0,0357	101
7	0,0176	0,0176	110
8	0,0010	0,0010	111

previamente projetado com o algoritmo PCA.

Para codificar um determinado vetor de entrada, o algoritmo 1/2PCA testa, inicialmente, a média desse vetor. Para tanto, é necessário apenas avaliar a soma das componentes desse vetor:

$$\begin{cases} \text{soma}(\mathbf{x}) \geq 0 \Rightarrow \text{média}(\mathbf{x}) \geq 0 \\ \text{soma}(\mathbf{x}) < 0 \Rightarrow \text{média}(\mathbf{x}) < 0 \end{cases} \quad (6.19)$$

O teste da média requer, portanto, $K - 1$ adições (uma vez que a soma das componentes do vetor de entrada é dada por $\text{soma}(\mathbf{x}) = \sum_{j=1}^K x_j$) e 1 comparação (com 0).

Considerando a medida de distorção erro quadrático da Equação (2.5) e tendo em vista que o codificador realiza a busca pelo vizinho mais próximo em apenas $N/2$ vetores-código, observa-se que para codificar cada vetor de entrada são necessárias ainda $K(N/2)$ multiplicações, $K(N/2)$ subtrações, $(K - 1)(N/2)$ adições e $(N/2) - 1$ comparações.

O número total de adições do algoritmo de codificação 1/2PCA é, portanto, dado por $(K - 1) + (K - 1)(N/2)$, em que o primeiro termo da soma diz respeito ao “teste da média” e o segundo se refere ao processo de procura pelo vizinho mais próximo no dicionário com $N/2$ vetores-código. O número total de comparações, por sua vez, é igual a $1 + (N/2 - 1)$, em que o primeiro termo da soma se deve ao “teste da média” e o segundo termo é concernente às comparações de distância (distorção quadrática) do processo de busca pelo vizinho mais próximo.

A Tabela 6.8 apresenta um resumo do número total de operações envolvido no algoritmo de busca total⁵ (busca exaustiva pelo vizinho mais próximo, realizada de forma convencional em um dicionário com N vetores-código), denotado por BT, e no algoritmo 1/2PCA (em que apenas metade do dicionário, correspondente a $N/2$ vetores-código, é efetivamente utilizada pelo codificador).

Tabela 6.8: Número de operações requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, em função de K e N .

	N ^o de operações	
	BT	1/2PCA
×	KN	$KN/2$
-	KN	$KN/2$
+	$(K - 1)N$	$(K - 1)(1 + N/2)$
Comp.	$N - 1$	$N/2$

6.3.2 Resultados

A Tabela 6.8 mostra que o algoritmo 1/2PCA apresenta uma economia de 50% em termos do número de multiplicações (bem como subtrações) com relação ao algoritmo BT.

Conforme se pode observar na Tabela 6.9, à medida que N aumenta o algoritmo 1/2PCA tende a levar a uma economia de 50% em termos do número de adições em relação ao algoritmo BT. Na Tabela 6.9 observa-se que para $K = 4$ e $N = 128$ o algoritmo 1/2PCA requer 195 adições, o que representa uma economia de 49,21% em relação às 384 adições requeridas pelo algoritmo BT.

No que diz respeito ao número de comparações, a Tabela 6.10 mostra que para $N = 32$ e $N = 64$ são obtidas, respectivamente, economias de 48,38% e 49,20% com a utilização do algoritmo 1/2PCA em lugar do algoritmo BT. Observa-se que a economia do número de comparações do algoritmo 1/2PCA com relação ao algoritmo BT tende a 50% à medida que N aumenta.

Foram realizadas simulações que tiveram como objetivo avaliar a queda de qualidade do sinal de voz reconstruído obtida ao se utilizar o algoritmo 1/2PCA em substi-

⁵Maiores detalhes podem ser encontrados na Seção 2.2.

Tabela 6.9: Número de adições requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de N fixada a dimensão $K = 4$. Entre parênteses está indicada a economia do algoritmo 1/2PCA com relação ao algoritmo BT.

N	N ^o de adições	
	BT	1/2PCA
16	48	27 (43,75%)
32	96	51 (46,87%)
64	192	99 (48,43%)
128	384	195 (49,21%)

Tabela 6.10: Número de comparações requerido para codificar um vetor, ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de N . Entre parênteses está indicada a economia do algoritmo 1/2PCA com relação ao algoritmo BT.

N	N ^o de comparações	
	BT	1/2PCA
16	15	8 (46,67%)
32	31	16 (48,38%)
64	63	32 (49,20%)
128	127	64 (49,60%)

tuição ao algoritmo BT. A Tabela 6.11 apresenta valores de relação sinal-ruído segmental (SNRseg) do sinal de voz reconstruído para diversos valores de K e N . É possível observar que o algoritmo 1/2PCA pode eventualmente levar a uma pequena queda de SNRseg do sinal reconstruído quando comparado com o algoritmo BT. Para $K = 2$ e $N = 128$, por exemplo, obtém-se uma queda de 0,16 dB em termos de SNRseg do sinal reconstruído. As perdas de qualidade em termos de relação sinal-ruído segmental introduzidas pelo algoritmo 1/2PCA ocorrem quando o “teste da média” desse algoritmo leva um vetor de entrada a ser representado por um vetor-código do tipo $w_{i,PZ}$ quando de fato esse vetor de entrada deveria ser representado por um vetor-código do tipo $w_{i,NZ}$ caso o algoritmo BT (busca total pelo vizinho mais próximo no dicionário completo, de N vetores-código) fosse utilizado. Também ocorrem quando o “teste da média” do algoritmo 1/2PCA leva um vetor de entrada x a ser representado por um vetor-código do tipo $w_{i,NZ}$ quando de fato x deveria ser representado por um vetor-código do tipo $w_{i,PZ}$ caso o algoritmo BT fosse utilizado.

Tabela 6.11: SNRseg (dB) do sinal de voz reconstruído, obtidas ao serem utilizados os algoritmos BT e 1/2PCA, para diversos valores de K e N .

K	N	SNRseg (dB)	
		BT	1/2PCA
2	16	8,86	8,86
2	32	11,01	10,84
2	64	12,35	12,35
2	128	13,41	13,25
4	16	6,17	6,11
4	32	7,89	7,51
4	64	8,96	8,42
4	128	9,42	8,90

A Figura 6.12 ilustra um caso em que o algoritmo 1/2PCA determina como versão quantizada do vetor de entrada x um vetor-código que difere do que deveria ser determinado caso o algoritmo BT fosse utilizado. Na figura, as coordenadas x_1 e x_2 representam, respectivamente, a primeira e a segunda componentes de qualquer vetor (como x , w_8 e w_9) definido no espaço euclidiano R^2 . Observa-se que o algoritmo

1/2PCA, em virtude de realizar o “teste da média” (por meio do qual verifica que o vetor de entrada trata-se de um vetor com média negativa) e, conseqüentemente, informar ao decodificador que o vetor-código a ser produzido é um vetor do tipo $w_{i,NZ}$, faz com que o vetor-código escolhido como versão quantizada de x seja o vetor w_9 . O algoritmo BT, por sua vez, que realiza uma busca no dicionário completo, determina que o vetor-código w_8 seja produzido como a versão quantizada de x . No exemplo da Figura 6.12, portanto, observa-se que o “teste da média” faz com que o algoritmo 1/2PCA leve um vetor de entrada a ser representado por um vetor-código do tipo $w_{i,NZ}$ (vetor w_9) quando de fato o vetor x deveria ser representado por um vetor-código do tipo $w_{i,PZ}$ (vetor w_8) caso o algoritmo BT fosse utilizado. Na figura, a distância entre os vetores x e w_8 é denotada por d_8 e a distância entre os vetores x e w_9 é denotada por d_9 . Como $d_8 < d_9$ um maior erro de quantização é obtido com a utilização de w_9 . Em decorrência de poder eventualmente levar a um maior erro de quantização quando comparado ao obtido com o algoritmo BT, o algoritmo 1/2PCA pode levar a sinais reconstruídos com valores de SNRseg inferiores aos obtidos com uso do algoritmo BT.

Finalmente, é oportuno mencionar que o método de codificação proposto, embora tenha sido aplicado aos dicionários PCA, pode ser extensivo a quaisquer dicionários que apresentem simetria entre os vetores-código.

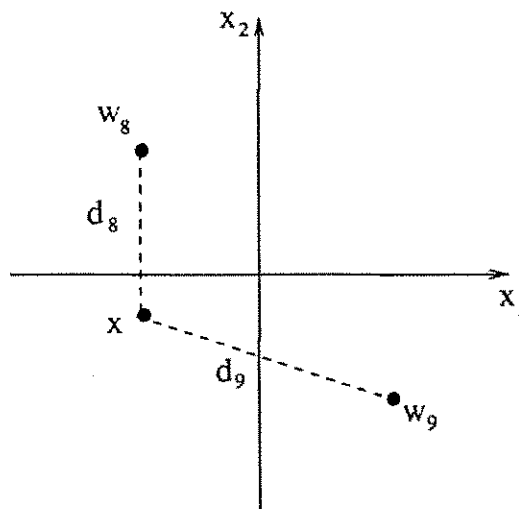


Figura 6.12: Exemplo em que os algoritmos 1/2PCA e BT levam a representações diferentes para um dado vetor de entrada x .

6.4 Avaliação dos Dicionários PCA para Uso em Transmissão de Voz por Canal Ruidoso

Nesta seção é realizada uma avaliação da *qualidade inerente* dos dicionários PCA sob o ponto de vista da robustez aos erros de transmissão de voz por um canal ruidoso. Nesse contexto, cumpre salientar que, em se tratando de compressão de voz baseada em QV, os algoritmos PCA e LBG visam gerar dicionários de qualidade, que levem a sinais reconstruídos com a menor distorção para uma determinada taxa de codificação. A dinâmica dos algoritmos PCA e LBG não leva em consideração a questão da sensibilidade dos dicionários projetados aos erros de transmissão. Daí utilizar-se, com efeito, no presente trabalho, o termo *robustez inerente* dos dicionários aos erros de canal.

A avaliação dos dicionários PCA quanto à robustez aos erros de canal é levada a efeito utilizando-se o modelo de canal binário simétrico. Para efeito de comparação são também utilizados dicionários LBG com o mesmo modelo de canal. Resultados de simulação apresentados neste trabalho mostram que os erros de transmissão introduzidos pelo canal levam a uma menor perda de qualidade do sinal de voz reconstruído ao serem utilizados dicionários PCA.

6.4.1 Robustez Inerente aos Erros de Canal

A organização estruturada dos dicionários PCA sugere que eles apresentem uma maior robustez inerente aos erros de transmissão quando comparados aos dicionários LBG. Observa-se, por exemplo, nas Tabelas 6.6 e 6.12, que nos dicionários PCA vetores-código com pequenas distâncias euclidianas têm uma melhor correspondência (relativamente ao algoritmo LBG) com palavras-binárias com pequenas distâncias de Hamming. A Tabela 6.6 também mostra que os dicionários PCA têm uma característica interessante: vetores-código simétricos têm palavras-binárias com máxima distância de Hamming, isto é, com distância de Hamming igual a $\log_2 N$ bits.

6.4.2 Resultados

Nesta seção são apresentados resultados de simulação envolvendo transmissão de voz por um canal binário simétrico (BSC, *binary symmetric channel*). Para avaliação da robustez inerente dos dicionários PCA e LBG aos erros de canal, utilizou-se um

Tabela 6.12: Dicionário LBG com 16 vetores-código bidimensionais $w_i = [w_{i1} \ w_{i2}]^T$, com $1 \leq i \leq 16$. A palavra-binária do i -ésimo vetor-código, w_i , é denotada por b_i , enquanto que w_{ij} representa a j -ésima componente do vetor w_i .

i	w_{i1}	w_{i2}	b_i
1	0,0006	-0,0003	0000
2	-0,1391	-0,1414	0001
3	-0,0740	-0,0901	0010
4	0,0468	0,0437	0011
5	0,0576	-0,2353	0100
6	0,1532	-0,0554	0101
7	-0,3344	-0,3209	0110
8	-0,2202	-0,2224	0111
9	-0,2029	-0,0461	1000
10	-0,0446	-0,0288	1001
11	0,2635	0,3676	1010
12	0,5461	0,5536	1011
13	0,1871	0,1941	1100
14	0,1223	0,1106	1101
15	-0,0178	0,1747	1110
16	0,3902	0,2229	1111

sinal de voz constituído de 29120 amostras (3,64 s), correspondente ao conjunto de sentenças “*O sol ilumina a fachada de tarde. Trabalhou mais do que podia*”.

A aquisição (resolução 8,0 bit/amostra e taxa de amostragem 8 kHz) dos sinais de voz utilizados nas simulações⁶ foi realizada usando uma estação de trabalho Sun[®], instalada com utilitários de processamento de áudio. A qualidade dos sinais de voz reconstruídos foi avaliada utilizando-se a relação sinal-ruído segmental (SNRseg).

O primeiro conjunto de avaliações teve como objetivo realizar uma análise comparativa dos dicionários PCA e LBG sob o ponto de vista do índice de desordem. Consubstanciado na Equação 2.8, utilizado como uma figura de mérito para avaliar o grau de organização de um dicionário tendo em vista o problema de transmissão por canal

⁶Foram utilizados sinais distintos para projeto e para avaliação dos dicionários.

ruidoso, o índice de desordem acumula as distâncias euclidianas entre vetores-código cujas palavras-binárias apresentam distância de Hamming igual a 1 bit⁷.

A Tabela 6.13 mostra que os dicionários PCA apresentam índice de desordem (em alguns casos muito) inferior ao apresentado pelos dicionários LBG, o que remete (conforme mostram os resultados apresentados ao final desta seção) à constatação de os dicionários PCA serem mais adequados que os dicionários LBG para o caso de o sistema de comunicação baseado em QV utilizar um canal ruidoso.

Considerando, por exemplo, que o codificador tenha transmitido a palavra-binária 1001, devido a um erro introduzido pelo canal ruidoso, admita-se que o decodificador tenha recebido a palavra-binária 1011, que dista de 1 bit da palavra-binária transmitida. Assim, utilizando o dicionário PCA da Tabela 6.6, o decodificador produz como saída o vetor $[-0,0550 \ - 0,0550]^T$ quando de fato deveria produzir como saída o vetor $[-0,0176 \ - 0,0176]^T$, caso a transmissão tivesse sido realizada sem erro. Por outro lado, usando o dicionário LBG da Tabela 6.12, o decodificador produz como saída o vetor $[0,5461 \ 0,5536]^T$ quando de fato deveria produzir como saída o vetor $[-0,0446 \ - 0,0288]^T$, caso a transmissão tivesse sido realizada sem erro. Observa-se, neste exemplo, que o erro de transmissão é mais crítico no caso LBG. De fato, ao ser utilizado o dicionário LBG, a transmissão sem erro leva a um vetor com duas componentes negativas de baixa amplitude; o erro de canal considerado leva a um vetor com componentes positivas de alta amplitude. No caso de ser utilizado dicionário PCA, tanto a transmissão sem erro quanto a transmissão com erro levam a um vetor com amostras negativas de baixa amplitude.

A Tabela 6.14 apresenta valores de SNRseg do sinal reconstruído (mais precisamente, o valor médio de SNRseg resultante de 50 transmissões do sinal para cada uma das diversas probabilidades de erro de bit, ϵ , consideradas) para dicionários LBG e PCA com $N = 16, 32, 64$ e 128 vetores de reconstrução de dimensão $K = 2$.

A queda de qualidade provocada pelo canal ruidoso no sinal de voz reconstruído pode ser avaliada por meio de

$$RP_{\text{SNRseg}}(\epsilon) = \frac{\text{SNRseg}(\text{tsr}) - \text{SNRseg}(\epsilon)}{\text{SNRseg}(\text{tsr})} \times 100\% \quad (6.20)$$

em que $RP_{\text{SNRseg}}(\epsilon)$ denota a redução percentual (em relação a uma transmissão sem

⁷Para baixas probabilidades de erro de bit de um canal ruidoso, a probabilidade de ocorrência de múltiplos erros nas palavras-binárias é pequena quando comparada com a probabilidade de ocorrência de erro em um bit.

Tabela 6.13: Índice de desordem para os dicionários LBG e PCA, para diversos valores de K e N .

K	N	I_{dis}	
		LBG	PCA
2	16	10,08	8,98
2	32	28,20	11,97
2	64	39,92	29,45
2	128	62,00	40,11
4	16	27,22	25,63
4	32	60,05	33,77
4	64	84,13	44,34
4	128	128,62	61,27

erro, denotada por tsr) da SNR_{seg} do sinal reconstruído após transmissão por um canal BSC com probabilidade de erro de bit igual a ϵ , enquanto $\text{SNR}_{\text{seg}}(\text{tsr}) = \text{SNR}_{\text{seg}}(\epsilon = 0, 0)$ denota a SNR_{seg} do sinal reconstruído após transmissão sem erro e $\text{SNR}_{\text{seg}}(\epsilon)$ representa a SNR_{seg} do sinal reconstruído após transmissão por um canal BSC com probabilidade de erro de bit igual a ϵ . Portanto, fixados N , K e ϵ , diz-se que o dicionário PCA apresenta uma maior robustez inerente aos erros de canal que a apresentada pelo dicionário LBG se o valor de $\text{RP}_{\text{SNR}_{\text{seg}}}(\epsilon)$ obtido com o dicionário PCA for inferior ao valor de $\text{RP}_{\text{SNR}_{\text{seg}}}(\epsilon)$ obtido com o dicionário LBG. Nesse sentido, a Tabela 6.15 mostra que os dicionários PCA são menos sensíveis aos erros de canal que os dicionários LBG, em virtude de apresentarem, para todos os valores de N , K e ϵ considerados, menores valores de $\text{RP}_{\text{SNR}_{\text{seg}}}(\epsilon)$. Para $K = 2$, $N = 32$ e $\epsilon = 0,0005$, por exemplo, o dicionário PCA apresenta

$$\text{RP}_{\text{SNR}_{\text{seg}}}(0,0005) = \frac{11,01 - 10,37}{11,01} \times 100\% = 5,81\%$$

enquanto que o algoritmo LBG apresenta

$$\text{RP}_{\text{SNR}_{\text{seg}}}(0,0005) = \frac{10,33 - 9,37}{10,33} \times 100\% = 9,29\%.$$

A Figura 6.13 ressalta a robustez inerente dos dicionários PCA aos erros de canal. Até mesmo para o caso de um dicionário PCA levar a um sinal reconstruído com

Tabela 6.14: SNRseg (dB) em função da probabilidade de erro de bit, ϵ , para os dicionários LBG e PCA com dimensão $K = 2$ e tamanho N .

N	Dicionário	SNRseg							
		$\epsilon = 0,0$	$\epsilon = 0,0001$	$\epsilon = 0,0005$	$\epsilon = 0,001$	$\epsilon = 0,005$	$\epsilon = 0,01$	$\epsilon = 0,05$	$\epsilon = 0,1$
16	LBG	7,52	7,40	6,87	6,21	2,58	-0,17	-7,32	-10,19
	PCA	8,86	8,73	8,27	7,74	4,32	1,39	-7,04	-10,15
32	LBG	10,33	10,15	9,37	8,49	3,46	-0,03	-7,98	-10,91
	PCA	11,01	10,87	10,37	9,78	6,08	3,00	-4,89	-8,07
64	LBG	13,55	13,26	12,34	11,18	4,95	0,98	-7,39	-10,18
	PCA	12,35	12,17	11,42	10,56	5,65	1,99	-6,15	-9,05
128	LBG	15,81	15,54	14,54	13,51	7,09	2,67	-5,82	-8,57
	PCA	13,41	13,19	12,43	11,54	6,67	3,41	-4,44	-7,43

Tabela 6.15: Redução percentual de SNRseg em relação à transmissão sem erro, para alguns valores de probabilidade de erro de bit, ϵ , para dicionários LBG e PCA com dimensão $K = 2$ e tamanho N .

N	Dicionário	$RP_{SNRseg}(\epsilon)$			
		$\epsilon = 0,0001$	$\epsilon = 0,0005$	$\epsilon = 0,001$	$\epsilon = 0,005$
16	LBG	1,59%	8,64%	17,42%	65,69%
	PCA	1,46%	6,65%	12,64%	51,24%
32	LBG	1,74%	9,29%	17,81%	66,50%
	PCA	1,27%	5,81%	11,17%	44,77%
64	LBG	2,14%	8,92%	17,49%	63,46%
	PCA	1,45%	7,53%	14,49%	54,25%
128	LBG	1,70%	8,03%	14,54%	55,15%
	PCA	1,64%	7,30%	13,94%	50,26%

SNRseg inferior ao obtido com uso de dicionário LBG para o caso de transmissão sem erro (vide por exemplo SNRseg para $N = 64$ considerando $\epsilon = 0,0$ na Tabela 6.14), a robustez aos erros de canal apresentada pelo dicionário PCA é tal que a partir de uma determinada probabilidade de erro de bit os valores de SNRseg apresentados pelo sinal reconstruído com dicionário PCA supera os valores de SNRseg apresentados pelo sinal reconstruído com dicionário LBG.

Sensibilidade de Cada Bit das Palavras-Binárias aos Erros de Canal

Nesta seção são apresentados resultados referentes à avaliação da sensibilidade de cada bit das palavras-binárias $\mathbf{b}_i = (b_1(i), b_2(i), \dots, b_m(i))$ aos erros da canal. Para tanto, os bits de todas as palavras-binárias foram sistematicamente invertidos na saída do codificador de fonte. Observe que $b_1(i)$ corresponde ao primeiro bit ou bit mais significativo (MSB, *most significant bit*) da palavra \mathbf{b}_i , enquanto $b_m(i)$ corresponde ao bit menos significativo (LSB, *least significant bit*) da palavra-binária \mathbf{b}_i de $m = \log_2 N$ bits.

Admita-se que o sinal a ser codificado seja constituído de M vetores. Obedecendo à regra do vizinho mais próximo, o codificador de fonte transmite M palavras-binárias para o decodificador. Nesta seção é apresentada uma avaliação da relação sinal-ruído segmental do sinal de voz reconstruído admitindo-se que seja invertido o j -ésimo bit

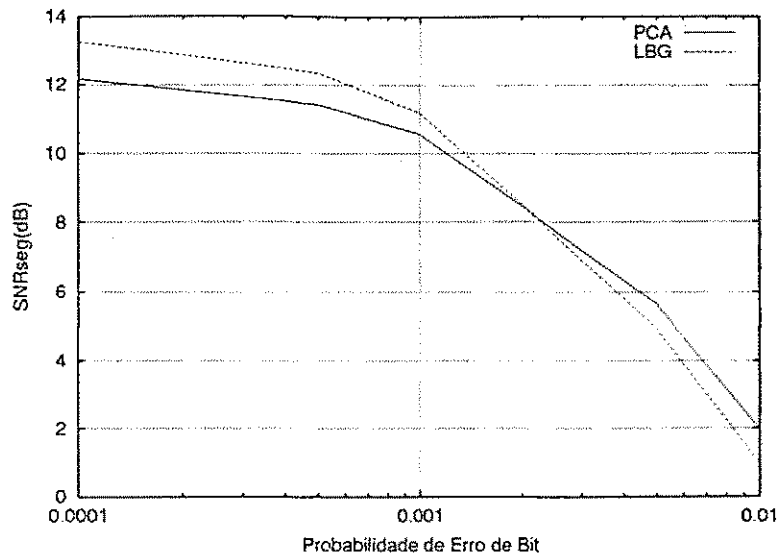


Figura 6.13: Relação sinal-ruído segmental do sinal de voz reconstruído, em função da probabilidade de erro de bit, para os dicionários LBG e PCA com $K = 2$ e $N = 64$.

(com $1 \leq j \leq m$) de cada uma das M palavras-binárias recebidas pelo decodificador do sistema de comunicações baseado em QV.

As Tabelas 6.16, 6.17, 6.18 e 6.19 mostram que, em se tratando dos dicionários PCA com dimensão $K = 2$, o 1º bit (MSB) é o mais importante no contexto de transmissão com erro. De fato, os menores valores de SNRseg (dB) do sinal de voz reconstruído com dicionários PCA são obtidos quando da inversão do 1º bit de cada palavra-binária \mathbf{b}_i . Nos dicionários PCA com dimensão $K = 2$, portanto, o 1º bit das palavras-binárias é o mais sensível aos erros de canal. Essa característica não é apresentada pelos dicionários LBG: na Tabela 6.17, referente a $N = 32$, o 2º bit das palavras-binárias LBG é o mais sensível aos erros de canal; na Tabela 6.18, concernente a $N = 64$, o 4º bit das palavras-binárias LBG é o mais sensível aos erros de canal.

A maior importância (no contexto de transmissão de voz por canal ruidoso) do 1º bit das palavras-binárias dos vetores-código PCA de dimensão $K = 2$ qualifica os dicionários PCA com $K = 2$ como adequados para aplicação de técnicas de UEP (*unequal error protection*).

Tabela 6.16: Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 16$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.

Condição imposta em b_i	SNRseg (dB)	
	PCA	LBG
Sem inversão de bit	8,86	7,52
Inversão do 1º bit (MSB)	-17,52	-16,26
Inversão do 2º bit	-12,05	-14,23
Inversão do 3º bit	-7,45	-14,71
Inversão do 4º bit (LSB)	-3,29	-10,81

Tabela 6.17: Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 32$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.

Condição imposta em b_i	SNRseg (dB)	
	PCA	LBG
Sem inversão de bit	11,01	10,33
Inversão do 1º bit (MSB)	-15,67	-12,16
Inversão do 2º bit	-10,76	-16,52
Inversão do 3º bit	-5,59	-13,77
Inversão do 4º bit	-0,77	-12,08
Inversão do 5º bit (LSB)	2,25	-14,13

Tabela 6.18: Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 64$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.

Condição imposta em b_i	SNRseg (dB)	
	PCA	LBG
Sem inversão de bit	12,35	13,55
Inversão do 1º bit (MSB)	-13,05	-12,23
Inversão do 2º bit	-10,44	-12,36
Inversão do 3º bit	-8,33	-14,01
Inversão do 4º bit	-9,03	-14,43
Inversão do 5º bit	-10,82	-10,80
Inversão do 6º bit (LSB)	-11,56	-13,35

Tabela 6.19: Sensibilidade de cada bit das palavras-binárias dos vetores-código LBG e PCA, considerando $K = 2$ e $N = 128$: valores de SNRseg (dB) do sinal reconstruído obedecendo às condições impostas.

Condição imposta em b_i	SNRseg (dB)	
	PCA	LBG
Sem inversão de bit	13,41	15,81
Inversão do 1º bit (MSB)	-12,97	-11,40
Inversão do 2º bit	-9,76	-12,44
Inversão do 3º bit	-7,05	-11,62
Inversão do 4º bit	-7,31	-11,48
Inversão do 5º bit	-8,63	-10,97
Inversão do 6º bit	-3,69	-9,65
Inversão do 7º bit (LSB)	2,30	-9,45

Capítulo 7

Conclusões

7.1 Introdução

O projeto de dicionários tem um papel importante para o bom desempenho de sistemas de processamento de sinais baseados em quantização vetorial (QV). Em sistemas de codificação de voz e imagens baseados em QV, a qualidade dos sinais reconstruídos depende dos dicionários projetados. Em sistemas de identificação de locutor que utilizam QV paramétrica, as taxas de identificação dependem dos dicionários de padrões acústicos de referência projetados para cada locutor cadastrado pelo sistema.

Neste trabalho foram apresentadas quatro técnicas para projeto de dicionários. A primeira técnica, denominada SOA, consiste na introdução de uma modificação do algoritmo apresentado por França e Aguiar Neto em [25]. A segunda, denominada SSC, apesar de preservar o mesmo acrônimo do algoritmo apresentado por França e Aguiar Neto em [24], introduziu uma modificação no algoritmo competitivo descrito em [24]. A terceira, denominada FS-SSC, é resultante da introdução do princípio de consciência de Grossberg no algoritmo SSC. A quarta, denotada por PCA, apresenta-se como uma modificação do algoritmo proposto por França e Aguiar Neto em [193]. A modificação consistiu em introduzir uma operação de normalização no conjunto de autovetores da matriz de covariância de um sinal de voz típico.

Foram apresentados resultados e avaliações referentes à codificação de voz e imagens baseada em QV, bem como resultados concernentes ao reconhecimento de locutores baseado em QV paramétrica.

Por meio da obtenção de expressões analíticas para o número de operações (divisões, multiplicações, comparações, adições e subtrações) realizadas pelos algoritmos LBG e

SSC durante o projeto do dicionário, foi desenvolvido um instrumental matemático que permitiu uma avaliação adequada da complexidade computacional desses algoritmos.

O trabalho também apresentou um método para redução da complexidade computacional envolvida na etapa de codificação da QV. O método proposto, denominado 1/2PCA em virtude de o codificador utilizar apenas metade de um dicionário PCA previamente projetado, explora eficientemente a organização estruturada que caracteriza os dicionários projetados com o algoritmo PCA.

7.2 Comentários sobre as Avaliações Realizadas

Em se tratando de codificação de forma de onda de voz baseada em quantização vetorial simples (isto é, QV no domínio temporal), mostrou-se que os dicionários SOA e SSC apresentam desempenho próximo. Para diversas taxas de codificação foi apontada uma superioridade dos dicionários SOA, SSC e FS-SSC sobre os dicionários LBG em termos de relação sinal-ruído segmental (SNRseg) do sinal de voz reconstruído.

Quanto à codificação de imagens baseada em QV simples (isto é, QV no domínio espacial), constatou-se que os algoritmos SOA e SSC apresentam desempenho satisfatório. As simulações realizadas mostraram que melhores resultados em termos de relação sinal-ruído de pico (PSNR) da imagem reconstruída podem ser obtidos com a substituição dos dicionários LBG por dicionários SOA ou SSC.

O presente trabalho também contemplou uma análise comparativa da qualidade *inerente* dos dicionários SOA, SSC e LBG sob o ponto de vista da complexidade computacional envolvida na fase de codificação da QV, ao serem utilizados dois algoritmos clássicos para otimização da procura do vizinho mais próximo. Simulações envolvendo QV de imagens mostraram que os dicionários SOA implicam uma maior redução da complexidade computacional (isto é, maior redução do número de multiplicações por amostra, quando da utilização dos dois algoritmos para procura eficiente do vizinho mais próximo) que a obtida ao serem utilizados dicionários SSC e LBG.

No que diz respeito à codificação de imagens usando DWT + QV (isto é, QV do domínio *wavelet*), observou-se que os dicionários multiresolucionais SSC podem levar a imagens reconstruídas com maiores valores de PSNR, quando comparados aos dicionários multiresolucionais LBG.

Em se tratando de QV dos parâmetros LSF, constatou-se que os dicionários SSC em geral levam a uma percentagem de *outliers* (com distorção espectral entre 2 dB e

4 dB como também com distorção espectral superior a 4 dB) inferior à obtida com o uso de dicionários LBG.

Quanto à quantização vetorial aplicada ao reconhecimento de locutor, os algoritmos SOA, SSC, FS-SSC e LBG foram utilizados para projetar dicionários (64 vetores-código de dimensão 12) de padrões acústicos. Foi considerado um sistema de identificação de locutor baseado em QV paramétrica no qual foram cadastrados 20 locutores. Foram considerados 5 parâmetros acústicos: coeficientes LPC, cepestrais, cepestrais ponderados, delta cepestrais e delta cepestrais ponderados. Para cada parâmetro acústico considerado (com exceção de delta cepestrais ponderados e cepestrais ponderados), a substituição dos dicionários LBG pelos dicionários de padrões acústicos obtidos com algum dos algoritmos não-supervisionados (SOA, SSC ou FS-SSC) leva a uma melhoria de desempenho em termos de taxas médias de identificação.

Por meio da realização de projeto de dicionários destinados à QV de forma de onda de voz, os resultados concernentes à contabilização do número de operações executadas pelos algoritmos SSC e LBG respaldaram as conclusões resultantes das expressões analíticas desenvolvidas no presente trabalho para a complexidade computacional desses algoritmos. Apontou-se uma superioridade do algoritmo SSC sobre o algoritmo LBG em termos de número de operações realizadas.

No que se refere ao algoritmo PCA, as simulações realizadas apontaram uma superioridade desse método sobre sua versão original, proposta em [193]. Para várias taxas de codificação, o algoritmo PCA apresentou um desempenho superior ao apresentado pelo algoritmo LBG e próximo do desempenho dos algoritmos SOA e SSC, em termos de SNRseg dos sinais reconstruídos. Sessões de escuta informais mostraram, para todas as taxas de codificação consideradas do quantizador vetorial com dimensão $K = 2$, que os sinais de voz reconstruídos utilizando-se dicionários PCA apresentam qualidade superior à apresentada pelos sinais reconstruídos com uso de dicionários LBG e SOA. Os testes de escuta informais, entretanto, apontaram a presença de um "sinal artificial", o ruído musical, nos sinais reconstruídos por meio de QV com dimensões $K = 4$ e $K = 8$. Mostrou-se que o algoritmo PCA gera dicionários estruturados, o que não ocorre com os algoritmos SOA, SSC e LBG.

Quanto ao algoritmo 1/2PCA, as avaliações realizadas e os resultados obtidos mostraram que este novo método de codificação leva a uma economia de 50% em termos de número de multiplicações e subtrações quando comparado ao algoritmo convencional de busca total (BT) do vizinho mais próximo. À medida que o tamanho do dicionário

aumenta, o algoritmo 1/2PCA tende a levar a uma economia de 50% em termos de número de adições e comparações quando comparado ao algoritmo BT. Observou-se, no entanto, que o algoritmo 1/2PCA pode eventualmente levar a uma pequena queda de SNRseg do sinal reconstruído quando comparado com o algoritmo BT

No tocante à avaliação dos dicionários PCA para uso em transmissão de voz por canal ruidoso, os resultados obtidos mostraram que os dicionários PCA apresentam uma maior robustez aos erros de canal quando comparados aos dicionários LBG.

7.3 Contribuições Mais Relevantes

As seguintes contribuições podem ser destacadas:

1. Desenvolvimento do algoritmo SOA, a partir da introdução do parâmetro taxa de aprendizagem final no algoritmo apresentado em [25]. Em virtude de apresentar um parâmetro a mais, a versão do algoritmo SOA desenvolvida no presente trabalho apresenta uma maior flexibilidade (um maior número de graus de liberdade) para o projeto de dicionários;
2. Desenvolvimento do algoritmo SSC, resultante da introdução do parâmetro taxa de aprendizagem final no algoritmo apresentado em [24]. O algoritmo SSC desenvolvido no presente trabalho apresenta cinco parâmetros ajustáveis;
3. Descrição detalhada dos algoritmos SOA e SSC, destacando e discutindo as principais diferenças entre os algoritmos SOA e de Kohonen. Nesse sentido, o paradigma de vizinhança apresentado pelo algoritmo SOA foi devidamente abordado – mostrou-se que o algoritmo SOA, por definir a vizinhança no próprio espaço de padrões, como uma hiper-esfera centrada no vetor de treino, procura permitir que o conjunto de treino produza uma influência mais justa (em comparação ao algoritmo de Kohonen, cuja vizinhança é definida numa grade topológica de nós, sendo centrada no nó correspondente ao neurônio/vetor-código vencedor) dos vetores de treino no projeto dos vetores-código do dicionário. Mostrou-se também que a definição de vizinhança do algoritmo SOA representa uma evolução da definição de vizinhança apresentada por França e Aguiar Neto em [23], apresentando-se também como uma definição mais coerente que essa última;

4. Desenvolvimento do algoritmo FS-SSC, resultante da introdução do princípio de consciência de Grossberg no algoritmo SSC;
5. Obtenção de expressões analíticas (em função do tamanho do dicionário, da dimensão dos seus vetores-código, do número de vetores do conjunto de treino e do número de iterações realizadas durante o treinamento do dicionário) para o número de operações (divisões, multiplicações, comparações, adições e subtrações) realizadas pelos algoritmos SSC e LBG. Estas expressões permitiram estabelecer condições que devem ser obedecidas para que o algoritmo SSC seja mais eficiente que o algoritmo LBG quanto à complexidade computacional envolvida no projeto de dicionários;
6. Desenvolvimento do algoritmo PCA, resultante da modificação do algoritmo proposto por França e Aguiar Neto em [193]. A modificação consistiu na introdução de uma operação de normalização dos autovalores da matriz de covariância do sinal de voz. Mostrou-se que o método proposto apresenta um desempenho superior ao apresentado por sua versão original;
7. Concepção de um método para redução da complexidade computacional envolvida na etapa de codificação da QV. O método proposto, denominado 1/2PCA em virtude de o codificador utilizar apenas metade de um dicionário PCA previamente projetado, explora eficientemente a organização estruturada dos dicionários PCA.

7.4 Propostas de Trabalhos Futuros

Podem ser apontadas as seguintes propostas de continuação do trabalho:

1. Introdução de uma modificação na definição de vizinhança do algoritmo SOA, de modo a torná-lo adequado para projeto de dicionários que apresentem uma robustez a erros de canal – uma possível modificação consistiria em definir a vizinhança não mais no espaço euclidiano, mas sim no espaço de Hamming, isto é, na primeira fase do algoritmo SOA seriam atualizados todos os vetores-código cujas palavras-binárias fossem próximas (pequenas distâncias de Hamming) da palavra-binária do vetor-código vencedor;

2. Avaliação da complexidade computacional dos algoritmos FS-SSC, SOA e de Kohonen;
3. Estudo e desenvolvimento de técnicas para reconhecimento de locutor utilizando voz corrompida por ruído;
4. Aplicação de algoritmos de redes neurais para determinação de componentes principais;
5. Aplicação de técnicas mais elaboradas de alocação de bits em codificação de sinais utilizando *wavelets* e QV;
6. Implementação e avaliação de outra versão de aprendizagem competitiva sensível à frequência. Precisamente, incorporar no algoritmo FS-SSC a sistemática proposta em [186], em que é utilizada a distorção modificada

$$\hat{d}(\cdot) = \left(1 + \frac{f_i}{f_{\text{thd}}}\right) \times d(\cdot), \quad (7.1)$$

em que f_i é a frequência do vetor-código w_i (número de vezes que foi escolhido vencedor até o momento) e f_{thd} é um limiar de frequência. O termo $\frac{f_i}{f_{\text{thd}}}$ representa a sensibilidade à frequência da função distorção modificada. Segundo [186], a escolha de f_{thd} tem um papel importante para o bom desempenho do dicionário projetado.

Apêndice A

Entropia Normalizada dos Vetores-Código

Este apêndice descreve a entropia normalizada dos vetores-código, que serve como uma medida para avaliar o grau de homogeneidade da distribuição dos vetores da fonte ao longo das células de Voronoi [35].

Seja $W = \{w_i \mid i = 1, 2, \dots, N\}$ um dicionário de tamanho N , em que w_i denota o i -ésimo vetor-código K -dimensional. Seja p_i a probabilidade de que um dado vetor da fonte pertença à região ou célula de Voronoi correspondente a w_i . Em outras palavras, p_i representa a probabilidade de que w_i seja o vizinho mais próximo de um dado vetor de entrada x , isto é, o vetor-código mais semelhante a x dentre todos os vetores-código do dicionário.

Seja $S = \{x_s, s = 1, \dots, m\}$ um longo conjunto de treino (isto é, $m \gg N$) representativo, ou seja, S é um longo conjunto de vetores representativos que são utilizados no algoritmo de projeto de dicionário. O conjunto S também pode ser um longo conjunto representativo de vetores da fonte a ser codificada (quantizada vetorialmente) por meio da utilização de um dicionário previamente projetado. Devido ao mapeamento promovido pela quantização vetorial, o conjunto S é particionado em N conjuntos (disjuntos) de Voronoi $S_i, i = 1, \dots, N$, em que cada célula S_i coleciona todos os vetores da fonte que são mapeados no i -ésimo vetor-código: $S_i = \{x_s : Q(x_s) = w_i\}$. Em outras palavras, w_i é o vetor-código mais próximo de $x_s \in S_i$.

Seja m_i o tamanho do subconjunto S_i , isto é, m_i é o número de vetores de entrada mapeados no i -ésimo vetor-código. Uma estimativa para a probabilidade de que w_i seja o vetor-código mais próximo de qualquer x_s (que corresponde à probabilidade de

que S_i seja a célula de Voronoi de \mathbf{x}_s) pode ser obtida como

$$p_i = \frac{m_i}{m}. \quad (\text{A.1})$$

A entropia H dos vetores-código é definida como

$$H = \sum_{i=1}^N p_i \log_2 \left(\frac{1}{p_i} \right). \quad (\text{A.2})$$

A entropia normalizada \tilde{H} dos vetores-código é dada por

$$\tilde{H} = \frac{H}{\log_2 N}, \quad (\text{A.3})$$

isto é,

$$\tilde{H} = \frac{\sum_{i=1}^N p_i \log_2 \left(\frac{1}{p_i} \right)}{\log_2 N}. \quad (\text{A.4})$$

A máxima entropia normalizada ocorre para vetores-código equiprováveis. De fato, equiprobabilidade implica $H = \log_2 N$. Como consequência, $\tilde{H} = 1$. É importante observar que $\tilde{H} \rightarrow 1$ à medida que aumenta a homogeneidade da distribuição dos vetores de entrada ao longo das células de Voronoi, isto é, $\tilde{H} \rightarrow 1$ à medida que $H \rightarrow \log_2 N$. Por outro lado, para N determinado, a entropia normalizada decresce à medida que aumenta o número de células de Voronoi pequenas.

Bibliografia

- [1] Jayant, N. "Signal Compression: Technology Targets and Research Directions". *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 5, pp. 796-818, June 1992.
- [2] Jayant, N. S. and Noll, P. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.
- [3] Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, MA, 1992.
- [4] Gray, R. M. "Vector Quantization". *IEEE ASSP Magazine*, pp. 4-29, April 1984.
- [5] Cosman, P. C., Oehler, K. L., Riskin, E. A., and Gray, R. M. "Using Vector Quantization for Image Processing". *Proceedings of the IEEE*, Vol. 81, No. 9, pp. 1326-1341, September 1993.
- [6] Cosman, P. C., Gray, R. M., and Vetterli, M. "Vector Quantization of Image Subbands: A Survey". *IEEE Transactions on Image Processing*, Vol. 5, No. 2, pp. 202-225, February 1996.
- [7] Berger, T. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [8] Gray, R. M. and Neuhoff, D. L. "Quantization". *IEEE Transactions on Information Theory*, Vol. 44, No. 6, pp. 2325-2383, October 1998.
- [9] Kosko, B. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.

- [10] Linde, Y., Buzo, A., and Gray, R. M. "An Algorithm for Vector Quantizer Design". *IEEE Transactions on Communications*, Vol. COM - 28, No. 1, pp. 84-95, January 1980.
- [11] Ramamurthi, B. and Gersho, A. "Classified Vector Quantization of Images". *IEEE Transactions on Communications*, Vol. COM-34, No. 11, pp. 1105-1115, November 1986.
- [12] Moayeri, N., Neuhoff, D. L., and Stark, W. E. "Fine-Coarse Vector Quantization". *IEEE Transactions on Signal Processing*, Vol. 39, No. 7, pp. 1503-1515, July 1991.
- [13] Abut, H., Gray, R. M., and Rebolledo, G. "Vector Quantization of Speech and Speech-Like Waveforms". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 3, pp. 423-435, June 1982.
- [14] Gersho, A. and Cuperman, V. "Vector Quantization: A Pattern-Matching Technique for Speech Coding". *IEEE Communications Magazine*, pp. 15-20, December 1983.
- [15] Goldberg, M. and Sun, H. "Image Sequence Coding Using Vector Quantization". *IEEE Transactions on Communications*, Vol. COM-34, No. 7, pp. 703-710, July 1986.
- [16] Nasrabadi, N. M. and King, R. A. "Image Coding Using Vector Quantization: A Review". *IEEE Transactions on Communications*, Vol. 36, No. 8, pp. 957-971, August 1988.
- [17] Zeger, K., Vaisey, J., and Gersho, A. "Globally Optimal Vector Quantizer Design by Stochastic Relaxation". *IEEE Transactions on Signal Processing*, Vol. 40, No. 2, pp. 310-322, February 1992.
- [18] Kohonen, T. *Self-Organization and Associative Memory (3rd ed)*. Springer-Verlag, Berlin, 1989.
- [19] Kohonen, T. "The Self-Organizing Map". *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1464-1480, September 1990.

- [20] Haykin, S. *Neural Networks - A Comprehensive Foundation*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458, 1999.
- [21] França, R. M. V. "Quantização Vetorial de Forma de Onda de Voz Usando um Algoritmo Competitivo". *Relatório Técnico, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1993.
- [22] França, R. M. V. "Um Algoritmo Auto-Organizativo para Quantização Vetorial de Forma de Onda". *Relatório Técnico, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1993.
- [23] França, R. M. V. and Aguiar Neto, B. G. "A Self-Organizing Algorithm for Waveform Vector Quantization". *Records of SBT/IEEE International Telecommunication Symposium*, pp. 136-140, August 1994.
- [24] França, R. M. V. and Aguiar Neto, B. G. "Voice Waveform Vector Quantization Using a Competitive Algorithm". *Records of the IEEE GLOBECOM'94*, pp. 872-875, November 1994.
- [25] França, R. M. V. and Aguiar Neto, B. G. "A Modified Kohonen's Algorithm for Voice Waveform Vector Quantization". *Records of the XI Simpósio Brasileiro de Inteligencia Artificial*, pp. 351-366, October 1994.
- [26] França, R. M. V. and Aguiar Neto, B. G. "Comparing Self-organizing Algorithms for Vector Quantization". *Proceedings of the International Conference EANN - Engineering Applications of Neural Networks*, pp. 481-484, August 1995.
- [27] Karayiannis, N. B. and Pai, P.-I. "Fuzzy Vector Quantization Algorithms and Their Applications in Image Compression". *IEEE Transactions on Image Processing, Vol. 4, No. 9*, pp. 1193-1201, September 1995.
- [28] Karayiannis, N. B. and Pai, P.-I. "Fuzzy Algorithms for Learning Vector Quantization". *IEEE Transactions on Neural Networks, Vol. 7, No. 5*, pp. 1196-1211, September 1996.
- [29] Pan, J. S., McInnes, F. R., and Jack, M. A. "VQ Codebook Design Using Genetic Algorithms". *Electronics Letters, Vol. 31, No. 17*, pp. 1418-1419, 17th August 1995.

- [30] Equitz, W. H. "A New Vector Quantization Clustering Algorithm". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 10, pp. 1568-1575, October 1989.
- [31] Lee, D., Baek, S., and Sung, K. "Modified K-means Algorithm for Vector Quantizer Design". *IEEE Signal Processing Letters*, Vol. 4, No. 1, pp. 2-4, January 1997.
- [32] Baek, S., Jeon, B., and Sung, K.-M. "A Fast Encoding Algorithm for Vector Quantization". *IEEE Signal Processing Letters*, Vol. 4, No. 12, pp. 325-327, December 1997.
- [33] Katsavoudinis, I., Kuo, C.-C. J., and Zhang, Z. "Fast Tree-Structured Nearest Neighbor Encoding for Vector Quantization". *IEEE Transactions on Image Processing*, Vol. 5, No. 2, pp. 398-404, February 1996.
- [34] Bei, C.-D. and Gray, R. M. "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization". *IEEE Transactions on Communications*, Vol. COM-33, No. 10, pp. 1132-1133, October 1985.
- [35] Paliwal, K. K. and Ramasubramanian, V. "Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization". *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 538-540, May 1989.
- [36] Soleymani, M. R. and Morgera, S. D. "An Efficient Nearest Neighbor Search Method". *IEEE Transactions on Communications*, Vol. COM-35, No. 6, pp. 677-679, June 1987.
- [37] Soleymani, M. R. and Morgera, S. D. "A Fast MMSE Encoding Technique for Vector Quantization". *IEEE Transactions on Communications*, Vol. 37, No. 6, pp. 656-659, June 1989.
- [38] Huang, C.-M., Bi, Q., Stiles, G. S., and Harris, R. W. "Fast Full Search Equivalent Encoding Algorithms for Image Compression Using Vector Quantization". *IEEE Transactions on Image Processing*, Vol. 1, No. 3, pp. 413-416, July 1992.
- [39] Torres, L. and Huguet, J. "An Improvement on Codebook Search for Vector Quantization". *IEEE Transactions on Communications*, Vol. 42, No. 2/3/4, pp. 208-210, February/March/April 1994.

- [40] Tai, S. C., Lai, C. C., and Lin, Y. C. "Two Fast Nearest Neighbor Searching Algorithms for Image Vector Quantization". *IEEE Transactions on Communications*, Vol. 44, No. 12, pp. 1623-1628, December 1996.
- [41] Crochiere, R. E., Webber, S. M., and Flanagan, J. K. L. "Digital Coding of Speech in Sub-bands". *The Bell System Technical Journal*, Vol. 55, pp. 1069-1086, October 1976.
- [42] Vetterli, M. "Multi-dimensional Sub-band Coding: Some Theory and Algorithms". *Signal Processing*, Vol. 6, pp. 97-112, April 1984.
- [43] Woods, J. W. and O'Neil, S. D. "Subband Coding of Images". *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-34, pp. 1278-1288, October 1986.
- [44] Westerink, P. H., Boekee, D. E., Biemond, J., and Woods, J. W. "Sub-band Coding of Images Using Vector Quantization". *IEEE Transactions on Communications*, Vol. 36, No. 6, pp. 713-719, June 1988.
- [45] Misiti, M., Misiti, Y., Oppenheim, G., and Poggi, J.-M. *Wavelet Toolbox for Use with Matlab*. The Math Works, Inc., Notick, MA, 1996.
- [46] . "Special Issue on Wavelet Transforms and Multiresolution Signal Analysis". *IEEE Transactions on Information Theory*, Vol. 38, No. 2, April 1996.
- [47] . "Special Issue on Wavelets". *Proceedings of the IEEE*, Vol. 84, No. 4, March 1992.
- [48] Strang, G. and Nguyen, T. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [49] Resnikoff, H. L. and Wells-Jr., R. O. *Wavelet Analysis*. Springer-Verlag New York, Inc., 1998.
- [50] Ferreira, G. M. "Codificação de Canal Usando Wavelets em Sistemas de Comunicação Sujeitos ao Desvanecimento Rayleigh". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 2000.

- [51] Ferreira, G. M., Lopes, W. T. A. e Assis, F. M. "Codificação Wavelet Aplicada a Sistemas com Diversidade Espacial sobre Canais Sujeitos ao Desvanecimento Rayleigh". *CD-ROM do XVIII Simpósio Brasileiro de Telecomunicações (SBrT'2000)*, Gramado - RS, Brasil, Setembro 2000.
- [52] Burrus, C. S., Gopinath, R. A., and Guo, H. *Introduction to Wavelets and Wavelet Transforms*. Prentice-Hall, New Jersey, 1998.
- [53] Daubechies, I. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.
- [54] Meyer, Y. *Wavelets: Algorithms and Applications*. Society of Industrial and Applied Mathematics, Philadelphia, 1993.
- [55] Mallat, S. G. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-696, July 1989.
- [56] Vetterli, M. and Herley, C. "Wavelets and Filter Banks: Theory and Design". *IEEE Transactions on Signal Processing*, Vol. 40, No. 9, pp. 2207-2232, September 1992.
- [57] Desarte, P., Macq, B., and Slock, D. T. M. "Signal-Adapted Multiresolution Transform for Image Coding". *IEEE Transactions on Information Theory*, Vol. 38, No. 2, pp. 897-904, March 1992.
- [58] Xiong, Z., Ramchandran, K., and Orchard, M. T. "Space-Frequency Quantization for Wavelet Image Coding". *IEEE Transactions on Image Processing*, Vol. 6, No. 5, pp. 677-693, May 1997.
- [59] Wallace, G. K. "The JPEG Still Picture Compression Standard". *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, April 1991.
- [60] Davis, G. M. and Nosratinia, A. "Wavelet-based Image Coding: An Overview". *Applied and Computational Control, Signals and Circuits*, 1999.
- [61] Villasenor, J. D., Belzer, B., and Liao, J. "Wavelet Filter Evaluation for Image Compression". *IEEE Transactions on Image Processing*, Vol. 4, No. 8, pp. 1053-1060, August 1995.

- [62] Mandal, N. K., Panchanathan, S., and Aboulnasr, T. "Choice of Wavelets for Image Compression". *Lecture Notes in Computer Science, Vol. 1133*, pp. 239–249, 1996.
- [63] Unser, M. "Ten Good Reasons for Using Spline Wavelets". *Proc. SPIE Vol. 3169, Wavelets Applications in Signal and Image Processing V*", pp. 422–431, 1997.
- [64] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. "Image Coding Using Wavelet Transform". *IEEE Transactions on Image Processing, Vol. 1, No. 2*, pp. 205–220, April 1992.
- [65] Averbuch, A., Lazar, D., and Israeli, M. "Image Compression Using Wavelet Transform and Multiresolution Decomposition". *IEEE Transactions on Image Processing, Vol. 5, No. 1*, pp. 4–15, January 1996.
- [66] Liang, J. and Parks, T. W. "Image Coding Using Translation Invariant Wavelet Transforms with Symmetric Extensions". *IEEE Transactions on Image Processing, Vol. 7, No. 5*, pp. 762–769, May 1998.
- [67] DeVore, R. A., Jawerth, B., and Lucier, B. J. "Image Compression through Wavelet Transform Coding". *IEEE Transactions on Information Theory, Vol. 38, No. 2*, pp. 719–746, March 1992.
- [68] Lewis, A. S. and Knowles, G. "Image Compression Using the 2-D Wavelet Transform". *IEEE Transactions on Image Processing, Vol. 1, No. 2*, pp. 244–250, April 1996.
- [69] Shapiro, J. M. "Embedded Image Coding Using Zerotrees of Wavelet Coefficients". *IEEE Transactions on Signal Processing, Vol. 41, No. 12*, pp. 3445–3462, December 1993.
- [70] Said, A. and Pearlman, W. A. "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees". *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6*, pp. 243–250, June 1996.
- [71] Karayiannis, N. B., Pai, P.-I., and Zervos, N. "Image Compression Based on Fuzzy Algorithms for Learning Vector Quantization and Wavelet Image Decom-

- position". *IEEE Transactions on Image Processing*, Vol. 7, No. 8, pp. 1223-1230, August 1998.
- [72] Sampson, D. G., da Silva, E. A. B., and Ghanbari, M. "Wavelet Transform Image Coding Using Lattice Vector Quantisation". *Electronics Letters*, Vol. 30, No. 18, pp. 1477-1478, 1st September 1994.
- [73] Kjoelen, A., Umbaugh, S. E., and Zuke, M. "Compression of Skin Tumor Images - Wavelet/Vector Quantization Methods for Reducing the Time, Cost and Bandwidth of Storing and Transmitting Data". *IEEE Engineering in Medicine and Biology*, pp. 73-80, May/June 1998.
- [74] Cosman, P. C., Perlmutter, S. M., and Perlmutter, K. O. "Tree-structured Vector Quantization with Significance Map for Wavelet Image Coding". *Proceedings of Data Compression Conference (DCC)*, pp. 33-41, April 1995.
- [75] Karlekar, J., Poonacha, P. G., and Desai, U. B. "Image Compression Using Zero-tree and Multistage Vector Quantization". *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, Vol. 2, pp. 610-613, 1997.
- [76] Perlmutter, S. M., Perlmutter, K. O., and Cosman, P. C. "Vector Quantization with Zerotree Significance Map for Wavelet Image Coding". *Proceedings of 29th Asilomar Conference on Signal, Systems and Computers, Pacific Grove, CA*, November 1995.
- [77] Chang, S. G., Yu, B., and Vetterli, M. "Image Denoising Via Lossy Compression and Wavelet Thresholding". *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, pp. 604-607, 1997.
- [78] Donoho, D. L. "Denoising by Soft-Thresholding". *IEEE Transactions on Information Theory*, Vol. 41, No. 3, pp. 613-627, 1995; also *Stanford Statistics Department Technical Report TR-409* (November 1992).
- [79] Donoho, D. L. and Johnstone, I. M. "Adapting to Unknown Smoothness via Wavelet Shrinkage". *Stanford Statistics Department Technical Report TR-425*, June 1993.

- [80] Shohan, Y. and Gersho, A. "Efficient Bit Allocation for an Arbitrary Set of Quantizers". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1445–1453, September 1988.
- [81] Moulin, P. "A Multiscale Relaxation Algorithm for SNR Maximization in Nonorthogonal Subband Coding". *IEEE Transactions on Image Processing*, Vol. 4, pp. 1269–1281, September 1995.
- [82] Jayant, N., Johnston, J., and Safranek, R. "Signal Compression Based on Models of Human Perception". *Proceedings of the IEEE*, Vol. 81, No. 10, pp. 1385–1422, October 1993.
- [83] Osberger, W., Maeder, A. J., and McLean, D. "An Objective Quality Assessment Technique for Digital Image Sequences". *Proceedings of the IEEE International Conference on Image Processing (ICIP'96)*, Vol. I, pp. 897–900, 1997.
- [84] Lu, J., Algazi, V. R., and Estes, R. R. "Comparison of Wavelet Image Coders Using the Picture Quality Scale (PQS)". *Optical Engineering*, Vol. 35, No. 9 (disponível no endereço www.mathsoft.com/wavelets.html), pp. 2605–2619, 1996.
- [85] Eskicioglu, A. M. and Fischer, P. S. "Image Quality Measures and Their Performance". *IEEE Transactions on Communications*, Vol. 43, No. 12, pp. 2959–2965, December 1995.
- [86] Madeiro, F. "Avaliação de Desempenho de Algoritmos para Projeto de Quantizadores Vetoriais Aplicados à Compressão de Imagens". *Relatório Técnico RT00177/98*, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica, 1998.
- [87] Madeiro, F. "Análise da Correlação entre Medidas Objetivas e Subjetivas em Quantização Vetorial de Imagens". *Relatório Técnico RT00207/99*, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica, 1999.
- [88] Madeiro, F., Fachine, J. M., Aguiar Neto, B. G., and Alencar, M. S. "On the Performance of Objective Quality Measures for Evaluating Vector-Quantized Images". *Proceedings of the XVII Brazilian Telecommunications Symposium (SBT'99)*, Vila Velha – ES, Brazil, pp. 616–621, September 1999.

- [89] Kunt, M., Ikonomopoulos, A., and Kocher, M. "Second-Generation Image-Coding Techniques". *Proceedings of the IEEE, Vol. 73, No. 4*, pp. 549-574, April 1985.
- [90] Reid, M. M., Millar, R. J., and Black, N. D. "Second-Generation Image-Coding: An Overview". *ACM Computing Surveys, Vol. 29, No. 1*, pp. 3-29, March 1997.
- [91] Ran, X. and Farvardin, N. "A Perceptually Motivated Three-Component Image Model - Part I: Description of the Model". *IEEE Transactions on Image Processing, Vol. 4, No. 4*, pp. 401-415, April 1995.
- [92] Ramos, M. G., Hemani, S. S., and Tamburro, M. A. "Psychovisually-Based Multiresolution Image Segmentation". *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, pp. 66-69, 1997.
- [93] Höntsch, I., Karan, L. J., and Safraanek, R. J. "A Perceptually Tuned Embedded Zerotree Image Coder". *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, pp. 41-44, 1997.
- [94] Ran, X. and Farvardin, N. "A Perceptually Motivated Three-Component Image Model - Part II: Applications to Image Compression". *IEEE Transactions on Image Processing, Vol. 4, No. 4*, pp. 430-447, April 1995.
- [95] Miyahara, M., Kotani, K., and Algazi, V. R. "Objective Picture Quality Scale (PQS) for Image Coding". *IEEE Transactions on Communications, Vol. 46, No. 9*, pp. 1215-1226, September 1998.
- [96] Lopes, W. T. A., Madeiro, F., Alencar, M. S. e Aguiar Neto, B. G. "Uso de Codificação na Transmissão de Imagens Quantizadas Vetorialmente em Canal Gaussiano com Desvanecimento". *Anais do XVII Simpósio Brasileiro de Telecomunicações (SBT'99), Vila Velha - ES, Brasil*, pp. 748-753, Setembro, 1999.
- [97] Lopes, W. T. A., Madeiro, F., Alencar, M. S., and Aguiar Neto, B. G. "Space-Time Codes Applied to the Transmission of Vector-Quantized Images over a Rayleigh Fading Channel". *Proceedings of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000), Orlando, USA*, July 2000.
- [98] Tarokh, V., Seshadri, N., and Calderbank, A. R. "Space-time codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction".

IEEE Transactions on Information Theory, Vol. 44, No. 2, pp. 744–765, March 1998.

- [99] Lopes, W. T. A., Madeiro, F., Alencar, M. S. e Aguiar Neto, B. G. “Transmissão de Sinais de Voz Quantizados Vetorialmente Através de Canais com Desvanecimento Utilizando Múltiplas Antenas”. *Anais e CD-ROM do IX Simpósio Brasileiro de Microondas e Optoeletrônica (SBMO'2000)*, João Pessoa – PB, Brasil, pp. 511–515, Agosto, 2000.
- [100] Farvardin, N. “A Study of Vector Quantization for Noisy Channels”. *IEEE Transactions on Information Theory*, Vol. 36, No. 4, pp. 799–809, July 1990.
- [101] Knagenhjelm, P. and Agrell, E. “The Hadamard Transform – A Tool for Index Assignment”. *IEEE Transactions on Information Theory*, Vol. 42, No. 4, pp. 1139–1151, July 1996.
- [102] Zeger, K. and Gersho, A. “Pseudo-Gray Coding”. *IEEE Transactions on Communications*, Vol. 38, No. 12, pp. 2147–2158, December 1990.
- [103] Farvardin, N. and Vaishampayan, V. “On the Performance and Complexity of Channel-Optimized Vector Quantizers”. *IEEE Transactions on Information Theory*, Vol. 37, No. 1, pp. 155–160, January 1991.
- [104] Jafarkhani, H. and Farvardin, N. “Design of Channel-Optimized Vector Quantizers in the Presence of Channel Mismatch”. *IEEE Transactions on Communications*, Vol. 48, No. 1, pp. 118–124, January 2000.
- [105] Lopes, W. T. A., Madeiro, F., Alencar, M. S., and Aguiar Neto, B. G. “Simulated Annealing for Robust VQ: Improving Image Transmission through a Fading Channel”. *Proceedings of the VIth Brazilian Symposium on Neural Networks (IEEE SBRN'2000)*, Rio de Janeiro – RJ, Brazil, pp. 243–248, November 2000.
- [106] Kirkpatrick S., Gelatt, C. D. and M. P. Vecchi. “Optimization by Simulated Annealing”. *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.
- [107] Freeman, J. A. and Skapura, D. M. *Neural Networks - Algorithms, Applications and Programming Techniques*. Addison-Wesley, Reading, MA, 1991.

- [108] Lopes, W. T. A., Madeiro, F., Alencar, M. S., and Aguiar Neto, B. G. "An index assignment algorithm for improving the transmission of vector-quantized images over a Rayleigh fading channel". *CD-ROM of the XVIII Brazilian Telecommunications Symposium (SBrT'2000)*, Gramado - RS, Brazil, September 2000.
- [109] Lopes, W. T. A., Madeiro, F., Alencar, M. S. e Aguiar Neto, B. G. "Quantização Vetorial Robusta Utilizando *Simulated Annealing*". *CD-ROM do V Congresso Brasileiro de Redes Neurais (CBRN'2001)*, Rio de Janeiro - RJ, Brasil, pp. 601-606, Abril 2001.
- [110] Deller Jr., J. R., Proakis, J. G., and Hansen, J. H. L. *Discrete-time Processing of Speech Signals*. Macmillan Publishing Co., 1993.
- [111] Dimolitsas, S. "Objective Speech Distortion Measures and Their Telelevance to Speech Quality Assesments". *IEE Proceedings*, Vol. 136, Pt. I, No. 5, pp. 317-324, October 1989.
- [112] Dimolitsas, S. "Subjective Quality Quantification of Digital Voice Communication Systems". *IEE Proceedings-I*, Vol. 138, No. 6, pp. 585-595, December 1991.
- [113] Dimolitsas, S., Corcoran, F. L., and Ravishankar, C. "Dependence of Opinion Scores on Listening Sets Used in Degradation Category Rating Assessments". *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 5, pp. 421-424, September 1995.
- [114] Goodman, D. J. and Nash, R. D. "Subjective Quality of the Same Speech Transmission Conditions in Seven Different Countries". *IEEE Transactions on Communications*, Vol. COM-30, No. 4, pp. 642-654, April 1982.
- [115] Daumer, W. R. "Subjective Evaluation of Several Efficient Speech Coders". *IEEE Transactions on Communications*, Vol. COM-30, No. 4, pp. 655-662, April 1982.
- [116] Voran, S. "Objective Estimation of Perceived Speech Quality - Part I: Development of the Measuring Normalizing Block Technique". *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 4, pp. 371-382, July 1999.

- [117] Voran, S. "Objective Estimation of Perceived Speech Quality – Part II: Evaluation of the Measuring Normalizing Block Technique". *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 4, pp. 383–390, July 1999.
- [118] Karanusekera, S. A. and Kingsbury, N. G. "A Distortion Measure for Blocking Artifacts in Images Based on Human Visual Sensitivity". *IEEE Transactions on Image Processing*, Vol. 4, No. 6, pp. 713–724, June 1995.
- [119] Aguiar Neto, B. G. *Processamento e Transmissão Digital de Voz (Apostila)*. Curso de Pós Graduação em Engenharia Elétrica, Universidade Federal da Paraíba, Campina Grande - PB, 1995.
- [120] da Silva, L. M. "Contribuições para a Melhoria da Codificação CELP a Baixas Taxas de Bits". *Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica*, 1996.
- [121] Paliwal, K. K. and Atal, B. S. "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame". *IEEE Transactions on Speech and Audio Processing*, Vol. 1, No. 1, pp. 3–14, January 1993.
- [122] Dall'Agnol, S. L. Q., Alcaim, A., and de Marca, J. R. B. "Performance of LSF Vector Quantizers for VSELP Coders in Noisy Channels". *European Transactions on Telecommunications*, Vol. 5, No. 5, pp. 19/553–29/563, September - October 1994.
- [123] de Marca, J. R. B. "An LSF Quantizer for the North-American Half-Rate Speech Coder". *IEEE Transactions on Vehicular Technology*, Vol. 43, No. 3, pp. 413–419, August 1994.
- [124] LeBlanc, W. P., Bhattacharya, B., Mahmoud, S. A., and Cuperman, V. "Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 kb/s Speech Coding". *IEEE Transactions on Speech and Audio Processing*, Vol. 1, No. 4, pp. 373–385, October 1993.
- [125] Ramachandran, R. P., Sondhi, M. M., Seshadri, N., and Atal, B. S. "A Two Codebook Format for Robust Quantization of Line Spectral Frequencies". *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 3, pp. 157–168, May 1995.

- [126] Collura, J. S. and Tremain, T. E. "Vector Quantizer Design for the Coding of LSF Parameters". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'93)*, pp. II.29-II.32, 1993.
- [127] Hagen, R., Paksoy, E., and Gersho, A. "Voicing-Specific LPC Quantization for Variable-Rate Speech Coding". *IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 5*, pp. 485-494, September 1999.
- [128] Laroia, R., Phamdo, N., and Farvardin, N. "Robust and Efficient Quantization of Speech LSP Parameters Using Structured Vector Quantizers". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'91)*, pp. 641-644, 1991.
- [129] Sugamura, N. and Farvardin, N. "Quantizer Design in LSP Speech Analysis-Synthesis". *IEEE Journal on Selected Areas in Communications, Vol. 6, No. 2*, pp. 432-440, February 1988.
- [130] Hagen, R. "Robust LPC Spectrum Quantization - Vector Quantization by a Linear Mapping of a Block Code". *IEEE Transactions on Speech and Audio Processing, Vol. 4, No. 4*, pp. 256-280, July 1996.
- [131] Eriksson, T., Lindén, J., and Skoglund, J. "Interframe LSF Quantization for Noisy Channels". *IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 5*, pp. 495-509, September 1999.
- [132] da Silva, L. M. and Alcaim, A. "Differential Coding of Speech LSF Parameters Using Hybrid Vector Quantization and Bidirectional Prediction". *IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 2*, pp. 208-211, March 2000.
- [133] de Moura, J. E. A. e Alcaim, A. "Compressão de Imagens Através de Quantização Vetorial Classificada e Predição de Médias". *Anais do VIII SIBGRAPI*, pp. 73-78, 1995.
- [134] Kubrick, A. and Ellis, T. "Classified Vector Quantisation of Images: Codebook Design Algorithm". *IEE Proceedings, Vol. 137, Pt. I, No. 6*, pp. 379-386, December 1990.

- [135] Rabiner, L. R. and Schafer, R. W. *Digital Processing of Speech Signals*. Prentice-Hall, Upper Saddle River, New Jersey, 1978.
- [136] Costa, W. C. de A. "Reconhecimento de Fala Utilizando Modelos de Markov Escondidos (HMM's) de Densidades Contínuas". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, Junho, 1994.
- [137] Martim, T. B. "Practical Applications of Voice Input to Machines". *Proceedings of the IEEE, Vol. 64, No. 4*, pp. 487-501, April 1976.
- [138] Soong, F. K., Rosenberg, A. E., Juang, B.-H., and Rabiner, L. R. "A Vector Quantization Approach to Speaker Recognition". *AT&T Technical Journal, Vol. 66, Issue 2*, pp. 14-26, March/April 1987.
- [139] Doddington, G. R. "Speaker Recognition - Identifying People by their Voices". *Proceedings of the IEEE, Vol. 73, No. 11*, pp. 1651-1664, November 1985.
- [140] O'Shaughnessy, D. "Speaker Recognition". *IEEE ASSP Magazine*, pp. 4-17, October 1986.
- [141] Fachine, J. M. "Verificação de Locutor Utilizando Modelos de Markov Escondidos (HMMs) de Densidades Discretas". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, Abril, 1994.
- [142] Atal, B. S. "Automatic Recognition of Speakers from Their Voices". *Proceedings of the IEEE, Vol. 64, No. 4*, pp. 460-475, April 1976.
- [143] Rosenberg, A. E. "Automatic Speaker Verification: A Review". *Proceedings of the IEEE, Vol. 64, No. 4*, pp. 475-487, April 1976.
- [144] Campbell, J. P. "Speaker Recognition: A Tutorial". *Proceedings of the IEEE, Vol. 85, No. 9*, pp. 1437-1462, September 1997.
- [145] Furui, S. "Cepstral Analysis Technique for Automatic Speaker Verification". *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 29, No. 2*, pp. 254-272, April 1981.
- [146] Mammone, R. J., Zhang, X., and Ramachandran, R. P. "Robust Speaker Recognition - A Feature-Based Approach". *IEEE Signal Processing Magazine, Vol. 13, No. 5*, pp. 58-71, September 1996.

- [147] Gopalan, K., Anderson, T. R., and Cupples, E. J. "A Comparison of Speaker Identification Results Using Features Based on Cepstrum and Fourier-Bessel Expansion". *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 3, pp. 289–294, May 1999.
- [148] Ramachandran, R. P., Zilovic, M. S., and Mammone, R. J. "A Comparative Study of Robust Linear Predictive Analysis Methods with Applications to Speaker Identification". *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 2, pp. 117–125, March 1995.
- [149] Fechine, J. M. "Reconhecimento da Identidade Vocal Utilizando Modelagem Híbrida: Paramétrica e Estatística". *Tese de Doutorado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, Dezembro, 2000.
- [150] Rabiner, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, February 1989.
- [151] Rabiner, L. R. and Juang, B. H. "An Introduction to Hidden Markov Models". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 3, No. 1, pp. 4–16, February 1986.
- [152] Soong, F. K. and Rosenberg, A. E. "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 6, pp. 871–879, June 1988.
- [153] Xu, L., Oglesby, J., and Mason, J. S. "The Optimization of Perceptually-based Features for Speaker Identification". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'89)*, pp. 520–523, 1989.
- [154] He, J., Liu, L., and Palm, G. "A Discriminative Training Algorithm for VQ-based Speaker Identification". *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 3, pp. 353–356, May 1999.
- [155] Zilovic, M. S., Ramachandran, R. P., and Mammone, R. J. "Speaker Identification Based on the Use of Robust Cepstral Features Obtained from Pole-zero

- Transfer Functions". *IEEE Transactions on Speech and Audio Processing*, Vol. 6, No. 3, pp. 260–267, May 1998.
- [156] Madeiro, F. "Avaliação de Desempenho de Quantizadores Vetoriais Paramétricos Aplicados ao Reconhecimento de Locutor". *Relatório Técnico RT00178/98, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1998.
- [157] Savic, M. and Gupta, S. K. "Variable Parameter Speaker Verification System Based on Hidden Markov Modeling". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'90)*, pp. 281–284, 1990.
- [158] Matsui, T. and Furui, S. "Comparison of Text-independent Recognition Methods Using VQ-distortion and Discrete/continuous HMM's". *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 3, pp. 456–459, July 1994.
- [159] Bennani, Y., Soulie, F. F., and Gallinari, P. "A Connectionist Approach for Automatic Speaker Identification". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'90)*, pp. 265–268, 1990.
- [160] Oglesby, J. and Mason, J. S. "Optimisation of Neural Models for Speaker Identification". *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'90)*, pp. 261–264, 1990.
- [161] Farrell, K. R., Mammone, R. J., and Assaleh, K. T. "Speaker Recognition Using Neural Networks and Conventional Classifiers". *IEEE Transactions on Speech and Audio Processing (Special Issue on Neural Networks for Speech)*, Vol. 2, pp. 194–205, January 1994.
- [162] Violaro, F., Kaspar, B., and Martins, J. A. "Isolated Word Recognition Using Hidden Markov Models". *Proceedings of the VII Brazilian Symposium on Microwaves and Optoelectronics (TELEMO'96) and the XIV Brazilian Telecommunications Symposium (SBT'96), Curitiba - PR, Brazil*, pp. 533–538, July 1996.
- [163] Tolba, H. and O'Shaughnessy, D. "Voiced-Unvoiced Classification Using the First Mel Frequency Cepstral Coefficient". *Proceedings of the IEEE International*

- Conference on Speech Processing (ICSP'97), Seoul, Korea, pp. 137-142, August 1997.*
- [164] Simpson, P. K. *Neural Networks Applications - IEEE Technology Update Series*. II series, Technical Activities Board, Inc., New York, 1996.
- [165] Simpson, P. K. *Neural Networks Theory, Technology and Applications - IEEE Technology Update Series*. III series, Technical Activities Board, Inc., New York, 1996.
- [166] Veloso, L. R. "Reconhecimento de Caracteres Numéricos Manuscritos". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica, 1998.*
- [167] Correia, S. E. N. "Reconhecimento de Caracteres Numéricos Manuscritos Usando a Transformada Wavelet". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica, 2000.*
- [168] Beale, R. and Jackson, T. *Neural Computing: An Introduction*. Institute of Physics Publishing, Bristol and Philadelphia, 1990.
- [169] Hertz, J., Krogh, A., and Palmer, R. G. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA, 1992.
- [170] Haykin, S. "Neural Networks Expand SP's Horizons". *IEEE Signal Processing Magazine*, pp. 24-49, March 1996.
- [171] Grossberg, S. "Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors". *Biological Cybernetics, Vol. 23*, pp. 121-134, 1976.
- [172] Bauer, H.-U. and Pawelzik, K. R. "Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps". *IEEE Transactions on Neural Networks, Vol. 3, No. 4*, pp. 570-579, July 1992.
- [173] Lo, Z.-P., Yu, Y., and Bavarian, B. "Analysis of the Convergence Properties of Topology Preserving Neural Networks". *IEEE Transactions on Neural Networks, Vol. 4, No. 2*, pp. 207-220, March 1993.

- [174] Ritter, H. and Schulten, K. "Convergence Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability, and Dimension Selection". *Biological Cybernetics*, Vol. 60, pp. 59-71, 1988.
- [175] Yair, E., Zeger, K., and Gersho, A. "Competitive Learning and Soft Competition for Vector Quantizer Design". *IEEE Transactions on Signal Processing*, Vol. 40, No. 2, pp. 294-309, February 1992.
- [176] França, R. M. V. "Algoritmo de Kohonen Modificado para Quantização Vetorial". *Relatório Técnico, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1993.
- [177] Madeiro, F., Fachine, J. M., Aguiar Neto, B. G., and Alencar, M. S. "Building Acoustic Patterns for Speaker Recognition through a Self-Organizing Algorithm". *Proceedings of the 2nd Conference on Telecommunications (Conftele'99), Sesimbra, Portugal*, pp. 385-389, April 1999.
- [178] Madeiro, F., Vilar, R. M., Aguiar Neto, B. G., and de Assis, F. "Designing Codebooks for Speech Compression through a Neural Network Algorithm". *Proceedings of the 2nd Conference on Telecommunications (Conftele'99), Sesimbra, Portugal*, pp. 675-679, April 1999.
- [179] França, R. M. V. "Abordagem Neural da Quantização Vetorial de Sinais de Voz". *Proposta de Tese, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1996.
- [180] Madeiro, F. "Quantização Vetorial Aplicada à Compressão de Sinais de Voz e Imagem". *Dissertação de Mestrado, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, 1998.
- [181] Madeiro, F., Vilar, R. M., and Aguiar Neto, B. G. "A Self-Organizing Algorithm for Image Compression". *Proceedings of the Vth Brazilian Symposium on Neural Networks (IEEE SBRN'98), Belo Horizonte - MG, Brazil*, pp. 146-150, December 1998.
- [182] Madeiro, F., Fachine, J. M. e Aguiar Neto, B. G. "Algoritmo Modificado de Kohonen Aplicado ao Projeto de Dicionários de Padrões Acústicos para Re-

- conhecimento de Locutor”. *Anais do V Simpósio Brasileiro de Redes Neurais (SBRN'98)*, Belo Horizonte – MG, Brasil, pp. 22–26, Dezembro, 1998.
- [183] Madeiro, F., Vilar, R. M. e Aguiar Neto, B. G. “Avaliação de Desempenho de um Algoritmo Modificado de Kohonen em Quantização Vetorial”. *Anais do V Simpósio Brasileiro de Redes Neurais (SBRN'98)*, Belo Horizonte – MG, Brasil, pp. 41–46, Dezembro, 1998.
- [184] Fachine, J. M., Madeiro, F., Vilar, R. M., Aguiar Neto, B. G. e Alencar, M. S. “Algoritmo Competitivo Aplicado ao Reconhecimento da Identidade Vocal de Locutores”. *CD-ROM do V Congresso Brasileiro de Redes Neurais (CBRN'2001)*, Rio de Janeiro – RJ, Brasil, pp. 619–624, Abril 2001.
- [185] Krishnamurthy, A. K., Ahalt, S. C., Melton, D. E., and Chen, P. “Neural Networks for Vector Quantization of Speech and Images”. *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 8, pp. 1449–1457, October 1990.
- [186] Chen, O. T.-C., Sheu, B. J., and Fang, W.-C. “Image Compression Using Self-Organization Networks”. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 5, pp. 480–489, October 1994.
- [187] Costa Neto, P. L. de O. *Estatística*. Editora Edgard Blucher LTDA., São Paulo, SP, 1977.
- [188] Levine, D. M., Berenson, M. L. e Stephan, D. *Estatística: Teoria e Aplicações*. LTC – Livros Técnicos e Científicos Editora S. A., 2000.
- [189] Madeiro, F., Vajapeyam, M. S., Aguiar Neto, B. G. e Alencar, M. S. “Codificação de voz baseada em wavelets e quantização vetorial”. *CD-ROM do XVIII Simpósio Brasileiro de Telecomunicações (SBrT'2000)*, Gramado – RS, Brasil, Setembro, 2000.
- [190] Alcaim, A., Solewicz, J. A. e Moraes, J. A. “Frequência de Ocorrência dos Fones e Listas de Frases Foneticamente Balanceadas no Português Falado no Rio de Janeiro”. *Revista da Sociedade Brasileira de Telecomunicações*, Vol. 7, No. 1, pp. 23–41, Dezembro, 1992.

- [191] Xiong, Z. "Representation and Coding of Images Using Wavelets". *Ph.D. Thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign*, 1996.
- [192] de Lamare, R. C. e Alcaim, A. "Uma comparação de Quantizadores Vetoriais para Codificação dos Parâmetros LSF da Voz". *CD-ROM do XIX Simpósio Brasileiro de Telecomunicações (SBrT'2001), Fortaleza - CE, Brasil, Setembro 2001*.
- [193] França, R. M. V. and Aguiar Neto, B. G. "Designing Codebooks for Voice Waveform Vector Quantization Based on the Karhunen-Loève Transform". *Proceedings of the IEEE International Telecommunications Symposium (ITS'96)*, pp. 44-48, October 1996.
- [194] Madeiro, F., Vilar, R. M., Aguiar Neto, B. G., and Alencar, M. S. "An Algorithm for Voice Waveform VQ Codebook Design Based on PCA". *Journal of the Brazilian Telecommunications Society, Vol. 15, No. 2*, pp. 59-69, December 2000.
- [195] Gray, R. M. and Linde, Y. "Vector Quantizers and Predictive Quantizers for Gauss-Markov Sources". *IEEE Transactions on Communications, Vol. COM-30, No. 2*, pp. 381-389, February 1982.
- [196] Wu, L. and Fallside, F. "Source Coding and Vector Quantization with Codebook Excited Neural Networks". *Computer Speech and Language*, pp. 243-276, 1992.
- [197] Kaouri, H. A. and McCanny, J. V. "Reduced-Memory Multirate Vector Quantization". *Electronics Letters, Vol. 25, No. 7*, pp. 471-473, 30th March 1989.
- [198] Makur, A. and Subbalakshmi, K. P. "Variable Dimension VQ Encoding and Codebook Design". *IEEE Transactions on Communications, Vol. 45, No. 8*, pp. 897-899, August 1997.
- [199] CREATIVE - Audio Products. "Sound Blaster 16 Pro CSP (CT2290)". <http://support.creativehelp.com/specs/audio/sb16>.
- [200] Madeiro, F. "Projeto e Avaliação de Dicionários para Quantização Vetorial de Voz e Imagem". *Proposta de Tese, Universidade Federal da Paraíba, Departamento de Engenharia Elétrica*, Março, 2001.

- [201] Diamantsaras, K. J. and Kung, J. S. *Principal Component Neural Networks - Theory and Applications*. John Wiley & Sons, New York - NY, 1996.
- [202] Xu, L. and Yuille, A. L. "Robust Principal Component Analysis by Self-Organizing Rules Based on Statistical Physics Approach". *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 131-143, January 1995.
- [203] Fukunaga, J. *Statistical Pattern Recognition*. Academic Press, New York, 1990.
- [204] Chen, C. S. and Huo, K.-S. "Karhunen-Loeve Method for Data Compression and Speech Synthesis". *IEE Proceedings - I*, Vol. 138, No. 5, pp. 377-380, October 1991.
- [205] Sondhy, M. M., Schmidt, C. E. and Rabiner, L. R. "Improving the Quality of a Noise Speech Signal". *Bell System Technical Journal*, Vol. 60, No. 6, pp. 1847-1859, October 1981.
- [206] Dendrinou, M., Bakamidis, S. and Carayannis, G. "Speech Enhancement from Noise: A Regenerative Approach". *Speech Communication*, Vol. 10, No. 1, pp. 45-57, February 1991.
- [207] Aguiar Neto, B. G. "Signalaufbereitung in digitalen Sprachsignal Übertragungssystemen". *Doctor thesis. Technische Universität Berlin - Germany*, November 1987.
- [208] Vary, P., Heute, U. and Hess, W. *Digitale Sprachsignalverarbeitung*. Teubner Verlag, Stuttgart, 1998.