

# Aceleração de Projeto de Dicionários para Quantização Vetorial de Parâmetros LSF

Francisco Madeiro  
Escola Politécnica de Pernambuco  
Universidade de Pernambuco  
Pernambuco, Brasil  
madeiro@poli.br

Waslon T. A. Lopes  
Departamento de Engenharia Elétrica  
Universidade Federal da Paraíba  
Paraíba, Brasil  
waslon@ieee.org

**Resumo**—Este trabalho contempla o projeto de dicionários para codificação de voz, precisamente, para codificação dos parâmetros LSF (*line spectral frequencies*). É investigada uma alternativa para aceleração do projeto de dicionários LSF, que consiste na redução do número de iterações do algoritmo LBG (Linde-Buzo-Gray). Mostra-se que podem ser obtidas reduções em torno de 10% a 20% no número de iterações realizadas pelo algoritmo LBG, no cenário de projeto de dicionários para *split vector quantization*.

**Palavras-chaves**—quantização vetorial; projeto de dicionários; codificação de voz; algoritmo LBG; *k-means*.

## I. INTRODUÇÃO

O controle eficiente de erros de canal, a aplicação de técnicas criptográficas e a possibilidade de integração de sinais e dados podem ser apontados como benefícios decorrentes da representação digital de sinais.

Para aplicações que demandam redução de requisitos de largura de banda, para transmissão, e/ou de memória, para armazenamento, a codificação eficiente de sinais (como voz, imagem, áudio, vídeo) desempenha papel essencial. Exemplos das aplicações supracitadas incluem comunicação multimídia, navegação *web*, armazenamento de imagens médicas, comunicações móveis, arquivamento de impressões digitais e transmissão de imagens de sensoriamento remoto obtidas de satélites. Nessas aplicações, o propósito fundamental da compressão de sinais é reduzir o número de bits para representá-los adequadamente, de forma a minimizar os requisitos de memória e/ou de largura de banda.

A quantização vetorial (QV) [1,2], que se constitui em uma extensão da quantização escalar em um espaço multidimensional, encontra-se fundamentada na Teoria da Distorção Versus Taxa, segundo a qual um melhor desempenho é obtido codificando blocos de amostras (isto é, vetores) em vez de amostras individuais (isto é, escalares). Quando aplicada a sistemas de codificação de voz [3,4] e imagem [5,6], a QV tem sido utilizada com sucesso, permitindo elevadas taxas de compressão. A quantização vetorial também tem sido utilizada em outras aplicações, tais como, identificação vocal [7,8], segurança da informação (a exemplo de esteganografia e marca d'água digital) [9,10,11] e reconhecimento de palavras manuscritas [12].

Neste trabalho é avaliada uma alternativa para a redução do tempo gasto para o projeto de dicionários (quantizadores vetoriais) aplicados à transmissão de sinais de voz baseada em quantização vetorial. Em sistemas de codificação de voz baseados em QV, a qualidade dos sinais reconstruídos depende da qualidade dos dicionários projetados. No caso particular de projeto de dicionários para quantização vetorial de parâmetros LSF (*line spectral frequencies*), e.g. [3], um problema relevante é o tempo gasto no projeto de dicionários. Neste trabalho é investigada uma alternativa para a redução do tempo supramencionado, no âmbito do algoritmo Linde-Buzo-Gray (LBG) [13], o qual é a técnica mais amplamente utilizada para projeto de quantizadores vetoriais.

## II. QUANTIZAÇÃO VETORIAL

Fundamentada na Teoria da Distorção Versus Taxa, a QV consiste no mapeamento  $Q$  de um vetor de entrada  $x$  pertencente ao espaço euclidiano  $k$ -dimensional,  $\mathbb{R}^k$ , em um vetor pertencente a um subconjunto finito  $W$  de  $\mathbb{R}^k$ , ou seja

$$Q: \mathbb{R}^k \rightarrow W.$$

O dicionário  $W = \{w_i; i = 1, 2, \dots, N\}$  é o conjunto de vetores de reprodução (também chamados de vetores-código ou vetores de reconstrução),  $k$  é a dimensão do quantizador e  $N$  é o tamanho do dicionário. A taxa de codificação do quantizador vetorial, que mede o número de bits por componente do vetor, é dada por  $(1/k)\log_2 N$ . Em se tratando de codificação de imagens, a taxa é expressa em bits por pixel (bpp). Em codificação de forma de onda de voz, é expressa em bits/amostra.

O mapeamento  $Q$  dos vetores de entrada no dicionário introduz um particionamento de  $\mathbb{R}^k$  em  $N$  células, denominadas células de Voronoi,  $S_i$ ,  $i = 1, 2, \dots, N$ , tais que

$$S_i = \mathbb{R}^k$$

e

$$S_i \cap S_j = \emptyset, i \neq j,$$

em que cada célula  $S_i$  é definida como

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

Assim, o vetor código  $w_i$  é o vetor que representa todos os vetores de entrada  $x$  pertencentes à célula  $S_i$ .

Para codificar um vetor de entrada, é necessário calcular sua distância para cada um dos vetores pertencentes ao dicionário. A versão quantizada do vetor de entrada  $x$  é dada por  $w_i$ , ou seja,  $Q(x)=w_i$ , caso a distância do vetor  $x$  para o vetor  $w_i$  seja a menor distância encontrada, ou seja, se  $d(x, w_i) < d(x, w_j), \forall j \neq i$ . Neste caso, o codificador produz como saída a representação binária do índice  $i$ .

Convém mencionar que a quantização vetorial pode ser vista como uma forma de reconhecimento de padrões, em que cada vetor (padrão) de entrada é “aproximado por” ou mapeado em um dos vetores-código (padrões de referência) do dicionário.

Nas subseções a seguir é abordado o projeto de dicionários, que se constitui em um problema relevante no âmbito da QV. O desempenho de sistemas de processamento de sinais baseados em QV depende fortemente da qualidade dos dicionários projetados. Dentre as técnicas de projeto de dicionários, o algoritmo LBG (Linde-Buzo-Gray) [13] destaca-se por sua ampla utilização. O presente trabalho contempla a obtenção de dicionários para transmissão de voz baseada em quantização vetorial de parâmetros LSF (*line spectral frequencies*) [3]. No cenário de SVQ (*split vector quantization*), é avaliada uma alternativa para reduzir o tempo gasto no projeto dos dicionários para SVQ de parâmetros LSF. A alternativa, originalmente proposta para QV de imagens, consiste em reduzir o número de iterações do algoritmo LBG.

#### A. Algoritmo LBG

O algoritmo LBG, também conhecido como K-means ou GLA (Generalized Lloyd Algorithm), é a técnica mais utilizada para o projeto de dicionários.

Seja a iteração do algoritmo LBG denotada por  $n$ . Dados  $k$  (dimensão do quantizador),  $N$  (tamanho do dicionário) e um limiar de distorção  $\varepsilon \geq 0$ , o algoritmo LBG segue os seguintes passos:

1. Inicialização: dado um dicionário inicial  $W_0$  e um conjunto de treino  $X = \{x(m); m = 1, 2, \dots, M\}$ , faça  $n = 0$  e  $D_{-1} = \infty$ ;

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 distorção

$$D_n = \sum_{i=1}^N \sum_{x(m) \in S_i} d(x(m), w_i),$$

3. Teste de convergência (critério de parada do algoritmo): se  $(D_{n-1} - D_n) / D_n \leq \varepsilon$  pare, com  $W_n$  representando o dicionário final (dicionário projetado); caso contrário, continue;

4. Atualização do dicionário: compute os novos vetores-código como os centroides das classes de vetores; faça  $W_{n+1} \leftarrow W_n$ ; faça  $n \leftarrow n + 1$  e vá para o passo 2.

No algoritmo LBG a função distorção decresce monotonicamente, uma vez que o dicionário é iterativamente atualizado visando satisfazer as condições de centroide e de vizinho mais próximo [1]. Nesse algoritmo a distorção introduzida ao se representarem os vetores do conjunto de treinamento pelos vetores-código (centroides) é monitorada a cada iteração. A regra de parada (teste de convergência) do algoritmo baseia-se na distorção monitorada – o treinamento do dicionário é encerrado quando  $(D_{n-1} - D_n) / D_n \leq \varepsilon$ .

#### B. Algoritmo de Lee et al.

O algoritmo proposto por Lee et al. [14] é semelhante ao K-means convencional. Ele iterativamente atualiza o dicionário como o K-means convencional com uma pequena modificação que será apresentada a seguir. O tempo de computação para cada iteração é igual ao do K-means convencional, porém o algoritmo de Lee et al. geralmente necessita menos iterações para convergir.

A ideia do algoritmo de Lee et al. está centrada no ajuste do dicionário, quando, diferentemente do K-means convencional, não é feito o ajuste simplesmente pela média dos vetores pertencentes à região de Voronoi do dado vetor-código, e sim por uma estimativa de onde o vetor-código irá chegar com o decorrer das iterações. O algoritmo de Lee et al. escolhe como novo vetor-código um ponto entre o novo centroide para a região de Voronoi, calculado pelo K-means convencional, e o ponto de reflexão do vetor-código corrente em relação ao novo centroide. Para determinar o ponto exato que será escolhido, o algoritmo de Lee et al. utiliza um valor real de escala entre 1,0 e 2,0 e calcula o novo vetor-código como *novo vetor-código* = *vetor-código atual* + *escala* \* (*novo centroide* – *vetor-código atual*). Quando selecionamos o valor 1,0 para a escala estamos dando ao algoritmo de Lee et al. o comportamento idêntico ao K-means convencional.

#### C. Algoritmo de Paliwal e Ramasubramanian

O algoritmo proposto por Paliwal e Ramasubramanian [15] é semelhante ao algoritmo proposto por Lee et al. [14], porém, no lugar de uma escala de valor fixo ele utiliza uma escala variável em função da iteração. O algoritmo de Lee et al., por apresentar um valor fixo de escala, causa perturbações bruscas no dicionário, inclusive nas últimas iterações, onde ele já está próximo da convergência. Tal comportamento do algoritmo não é desejável, pois pode levar a um ótimo local mais pobre.

A alternativa de aceleração proposta por Paliwal e Ramasubramanian [15] busca solucionar o problema da escala fixa do algoritmo de Lee et al. [14]. No algoritmo de Paliwal e Ramasubramanian [15], a escala é função da iteração  $n$ ,

$$s = 1 + \frac{b}{b + n},$$

em que  $s$  é definido como o fator da escala e  $b > 0$ . Segundo Paliwal e Ramasubramanian, para QV de imagem no domínio espacial, uma boa escolha é  $b = 9$ . Assim como acontece no algoritmo de Lee *et al.*,  $1 < s < 2$ .

### III. SVQ DOS PARÂMETROS LSF

A SVQ dos parâmetros LSF consiste em dividir o vetor de parâmetros em subvetores. Em se tratando de vetores de dimensão 10, uma alternativa que tem sido utilizada consiste em utilizar dois subvetores, um de dimensão 4 e outro de dimensão 6. As técnicas de projeto de dicionários têm como alvo a redução da distorção espectral média obtida ao se representarem os vetores de treino pelos respectivos vetores-código.

Para o projeto de dicionários, é comum a utilização de versões ponderadas da distância Euclidiana. Lamare e Alcaim [16] utilizam a seguinte métrica:

$$d(f, f) = \sum_{i=1}^p \alpha_i(f) (f_i - f_i')^2,$$

em que

$$\alpha_i(f) = \frac{1}{f_i - f_{i-1}} + \frac{1}{f_{i+1} - f_i},$$

com  $i = 1, \dots, p$  e  $f_0 = 0$  e  $f_{p+1} = 0,5$ .

Em se tratando de codificação dos parâmetros LSF, a qualidade da quantização é avaliada por meio da distorção espectral, definida como

$$SD = \frac{1}{F_s} \int_0^{F_s} 10 \log_{10} S(f) - 10 \log_{10} S(f')^2 df^{\frac{1}{2}},$$

em que  $S(f)$  e  $S(f')$  denotam respectivamente a envoltória espectral original e quantizada e  $F_s$  é a frequência de amostragem.

Na seção a seguir são apresentados resultados referentes ao projeto de dicionário, no âmbito de SVQ (*split vector quantization*) dos parâmetros LSF.

### IV. RESULTADOS

Foram utilizados como conjuntos de treino e de teste sinais de voz com cerca de 2400 s e 30 s de duração respectivamente. Com uso de análise LPC (*linear predictive coding*) de ordem 10, considerando o método de autocorrelação a cada 20 ms, o conjunto de treino corresponde a 120.061 vetores LSF e o de teste a 1.505 vetores LSF. Foi utilizada a técnica de SVQ. Para a avaliação de desempenho dos dicionários projetados, foram determinadas: distorção espectral média; percentagem de *outliers* com distorção espectral entre 2 dB e 4 dB e percentagem de *outliers* com distorção espectral superior a 4 dB.

Nº. bits	LBG		P&R, $b=5$		P&R, $b=9$	
	1º. sub	2º. sub	1º. sub	2º. sub	1º. sub	2º. sub
20	66,65	54,95	58,75	48,10	50,50	45,55
22	46,35	43,00	42,50	39,45	40,30	35,70
24	33,45	31,35	27,35	26,45	26,05	25,90

Considerou-se SVQ correspondente ao uso de dois subvetores, um de dimensão 4 e outro de dimensão 6, modalidade de quantização denotada por SVQ (4,6). Foram projetados dicionários de tal maneira que o número de bits por quadro seja igual a 20, 22 ou 24. Para o caso de 20 bits por quadro, foram projetados dicionários de tamanho 1024 (10 bits por vetor-código) para o subvetor 1 (de dimensão 4) e dicionários de tamanho 1024 (10 bits por vetor-código) para o subvetor 2 (de dimensão 6); para o caso de 22 bits por quadro, foram projetados dicionários de tamanho 2048 (11 bits por vetor-código) para o subvetor 1 (de dimensão 4) e dicionários de tamanho 2048 (11 bits por vetor-código) para o subvetor 2 (de dimensão 6); para o caso de 24 bits por quadro, foram projetados dicionários de tamanho 4096 (12 bits por vetor-código) para o subvetor 1 (de dimensão 4) e dicionários de tamanho 4096 (12 bits por vetor-código) para o subvetor 2 (de dimensão 6).

São apresentados resultados referentes ao uso de 20 dicionários iniciais diferentes para cada cenário de simulações considerado.

Utilizou-se, nas simulações, uma máquina com a seguinte configuração: Processador Intel Core (TM) I5, CPU M 520 2,4 GHz, com 4 GB de memória RAM e sistema operacional Linux Fedora 20.

Conforme se pode observar na Tabela I, o fator de escala proposto por Paliwal e Ramasubramanian (2000) leva a um aumento da velocidade de convergência do algoritmo LBG. Os dicionários projetados, tanto para o subvetor 1 (de dimensão 4) quanto para o subvetor 2 (de dimensão 6), são obtidos em um menor número de iterações quando se usa o algoritmo de Paliwal e Ramasubramanian (2000) em substituição ao algoritmo LBG convencional. Além disso, um menor número de iterações é obtido com  $b=9$  em vez de  $b=5$ . Como exemplo, considere um número de bits/quadro correspondente a 22. Para o primeiro subvetor, o algoritmo LBG necessitou de um número médio de 46,35 iterações, ao passo que o algoritmo de Paliwal e Ramasubramanian necessitou de um número médio de 42,50 iterações com a escolha de  $b=5$  e de 40,30 iterações em média com  $b=9$ .

É possível observar, na Tabela I, que o uso da abordagem *look ahead* do algoritmo de Paliwal e Ramasubramanian contribui para a obtenção de uma economia, em geral, em torno de 10% a 20% no número de iterações, quando comparado com o algoritmo LBG. As Tabelas II e III mostram que essa economia não ocorre ao custo de comprometimento da qualidade dos dicionários projetados. De fato, no que diz respeito à distorção espectral (SD), considerando o uso de  $b=5$  ou  $b=9$ , a aceleração levou a um comprometimento de 0,01 dB em termos de distorção espectral em comparação aos valores

de SD obtidos com o uso de dicionários LBG, para todos os valores de número de bits por quadro considerados. No que diz respeito ao percentual de *outliers* com SD na faixa 2-4 dB, a maior diferença percentual observada foi de 0,72%. Quanto ao percentual de *outliers* com SD maior que 4 dB, a maior diferença percentual observada foi de 0,02%.

TABLE I. NÚMERO MÉDIO DE ITERAÇÕES PARA 20 DICIONÁRIOS INICIAIS DIFERENTES. NOTAÇÃO: 1°. SUB DENOTA O PRIMEIRO SUBVETOR E 2°. SUB DENOTA O SEGUNDO SUBVETOR

TABLE II. DIFERENÇA DE DESEMPENHO ENTRE ALGORITMO DE PALIWAL E RAMASUBRAMANIAN, COM  $B=5$ , E O ALGORITMO LBG EM TERMOS DE DISTORÇÃO ESPECTRAL (SD), PERCENTAGEM DE OUTLIERS COM SD NO INTERVALO 2-4 DB E PERCENTAGEM DE OUTLIERS COM  $SD > 4$  DB, AO SER RECONSTRUÍDO O CONJUNTO DE AVALIAÇÃO POR MEIO DE SVQ (4,6)

No. de bits	SD	Outliers	
		2-4 dB	> 4 dB
20	+0,01	+0,65%	-0,01%
22	+0,01	+0,54%	-0,02%
24	+0,01	+0,43%	0,00%

TABLE III. DIFERENÇA DE DESEMPENHO ENTRE O ALGORITMO DE PALIWAL E RAMASUBRAMANIAN, COM  $B=9$ , E O ALGORITMO LBG EM TERMOS DE DISTORÇÃO ESPECTRAL (SD), PERCENTAGEM DE OUTLIERS COM SD NO INTERVALO 2-4 DB E PERCENTAGEM DE OUTLIERS COM  $SD > 4$  DB, AO SER RECONSTRUÍDO O CONJUNTO DE AVALIAÇÃO POR MEIO DE SVQ (4,6)

No. de bits	SD	Outliers	
		2-4 dB	> 4 dB
20	+0,01	+0,72%	+0,01%
22	+0,01	+0,55%	-0,01%
24	+0,01	+0,30%	-0,01%

## V. CONCLUSÃO

O projeto de dicionários constitui-se em um problema relevante no âmbito de sistemas de processamento de sinais baseados em quantização vetorial, como por exemplo: sistemas de compressão de sinais (voz, imagem, áudio, vídeo), sistemas de identificação vocal e sistemas de esteganografia (ocultação da informação) e marca d'água digital.

Em se tratando de compressão de voz, o projeto de dicionários para codificação dos parâmetros LSF (*line spectral frequencies*), em geral, envolve o uso de grandes conjuntos de treino.

Este trabalho apresentou uma avaliação de desempenho do algoritmo de Paliwal e Ramasubramanian (P&R), quando aplicado ao projeto de dicionários para quantização vetorial, na modalidade *split vector quantization* (SVQ), dos parâmetros LSF. Cumpre mencionar que o algoritmo P&R foi proposto originalmente na literatura como alternativa de aceleração do

algoritmo LBG (Linde-Buzo-Gray), sendo o método P&R tradicionalmente investigado para projeto de dicionários voltados para quantização vetorial de imagens no domínio espacial, cenário em que se observa que o algoritmo P&R leva ao projeto de dicionários com um menor número de iterações que o obtido pelo algoritmo LBG. Além de uma velocidade de convergência maior que a do algoritmo LBG, o P&R em geral leva a dicionários de melhor qualidade, em termos de relação sinal-ruído de pico das imagens reconstruídas.

No presente trabalho, o algoritmo P&R, avaliado no cenário de projeto de dicionários para SVQ, também apresentou uma velocidade de convergência maior que a do algoritmo LBG, entretanto, não levou a dicionários de melhor qualidade, avaliada por meio da distorção espectral. Precisamente, os dicionários projetados com o método P&R apresentaram praticamente o mesmo desempenho, em termos de distorção espectral do sinal de voz reconstruído, que o apresentado pelos dicionários LBG.

Como trabalhos futuros, pretende-se incorporar técnicas eficientes de busca do vizinho mais próximo (como por exemplo o algoritmo *Equal-average Nearest Neighbor Search* (ENNS) [17]) na etapa de particionamento do algoritmo LBG, tendo em vista que esta etapa responde por parcela substancial do número de operações lógicas e aritméticas [18] gastas pelo algoritmo.

## AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

## REFERÊNCIAS

- [1] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.
- [2] R. M. Gray. "Vector Quantization". *IEEE ASSP Magazine*, pp. 4–29, April 1984.
- [3] K. K. Paliwal and B. S. Atal. "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.
- [4] S. Chatterjee and T. Sreenivas. "Conditional PDF-Based Split Vector Quantization of Wideband LSF Parameters". *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 641–644, September 2007.
- [5] K. Sasazaki, S. Saga, J. Maeda and Y. Suzuki. "Vector Quantization of Images with Variable Block Size". *Applied Soft Computing*, no. 8, pp. 634–645, 2008.
- [6] D. Tsolakis, G. Tsekouras and J. Tsimikas, "Fuzzy Vector Quantization for Image Compression Based on Competitive Agglomeration and a Novel Codeword Migration Strategy," *Engineering Applications of Artificial Intelligence*, pp. 1212-1225, vol 25, 2012.
- [7] J. He, L. Liu and G. Palm. "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.
- [8] F. Madeiro, J. M. Fachine, W. T. A. Lopes, B. G. Aguiar Neto and M. S. Alencar. "Identificação Vocal por Frequência Fundamental, QV e HMMS". In *Em-TOM-Ação: A Prosódia em Perspectiva*, edited by M. A. M. Aguiar and F. Madeiro, pp. 91–120. Editora Universitária da UFPE, Recife, PE, 2007.
- [9] Y. K. Chiang and P. Tsai. "Steganography Using Overlapping Codebook Partition". *Signal Processing*, vol. 88, no. 5, pp. 1203–1215, May 2008.

- [10] W.-J. Wang, C.-T. Huang, C.-M. Liu, P.-C. Su and S.-J. Wang. "Data Embedding for Vector Quantization Image Processing on the Basis of Adjoining State-codebook Mapping". *Information Sciences*, vol. 246, pp. 69–82, 2013.
- [11] J.-J. Shen and J.-M. Ren. "A Robust Associative Watermarking Technique Based on Vector Quantization". *Digital Signal Processing*, vol. 20, pp. 1408–1423, 2010.
- [12] J. Li, H. Mouchère and C. Viard-Gaudin. "An Annotation Assistance System Using an Unsupervised Codebook Composed of Handwritten Graphical Multi-stroke Symbols". *Pattern Recognition Letters*, vol. 35, pp. 46-57, 2014. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
- [13] Y. Linde, A. Buzo and R. M. Gray. "An Algorithm for Vector Quantizer Design". *IEEE Transactions on Communications*, 28, no. 1, pp. 84-95, January 1980.
- [14] D. Lee, S. Baek and K. Sung. "Modified K-means Algorithm for Vector Quantizer Design". *IEEE Signal Processing Letters*, vol. 4, no. 1, pp. 2-4, 1997.
- [15] K. K. Paliwal and V. Ramasubramanian. Comments on "Modified K-means Algorithm for Vector Quantizer Design". *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1964-1967, 2000.
- [16] R. C. Lamare and A. Alcáim. "Uma Comparação de Quantizadores Vetoriais Preditivos para Codificação dos Parâmetros LSF". *Simpósio Brasileiro de Telecomunicações*, 2001.
- [17] L. Guan and M. Kamel, "Equal-average Hyperplane Partitioning Method for Vector Quantization of Image Data", *Pattern Recognition Letters*, vol. 13, no. 10, pp. 693-699, October 1992.
- [18] F. Madeiro, W. T. A. Lopes, B. G. Aguiar Neto and M. S. Alencar. "Complexidade Computacional de um Algoritmo Competitivo Aplicado ao Projeto de Quantizadores Vetoriais". *Learning and Nonlinear Models*, vol. 2, no. 1, pp. 34-48, 2004.