

Avaliação de Desempenho de um Algoritmo Competitivo com Consciência Aplicado ao Projeto de Quantizadores Vetoriais

P.H. ESPÍRITO SANTO e F. MADEIRO

Resumo—Neste trabalho é avaliado o desempenho de um algoritmo competitivo com consciência aplicado ao projeto de dicionários para quantização vetorial de imagens. A avaliação contempla a complexidade computacional, a entropia normalizada dos vetores-código e a qualidade dos dicionários projetados, avaliada via relação sinal-ruído de pico das imagens reconstruídas. É apresentada uma comparação de desempenho com o algoritmo LBG (Linde-Buzo-Gray) e com o algoritmo competitivo simples.

Palavras-Chave—Quantização vetorial, projeto de dicionários, algoritmo competitivo com consciência.

Abstract—This work presents a performance evaluation of a frequency sensitive competitive learning algorithm applied to codebook design for image vector quantization. The evaluation concerns computational complexity, normalized entropy of the codevectors and quality of the designed codebooks, evaluated via peak signal to noise ratio of the reconstructed images. A comparative analysis against LBG (Linde-Buzo-Gray) algorithm and simple competitive learning algorithm is presented.

Keywords—Vector quantization, codebook design, frequency sensitive competitive algorithm.

I. INTRODUÇÃO

Sistemas multimídia, redes digitais de serviços integrados, videoconferência, telefonia móvel, sistemas de armazenamento de imagens médicas e de impressões digitais são exemplos de aplicações das técnicas de compressão de sinais, cujo objetivo fundamental é reduzir o número de bits de forma a representar adequadamente os sinais.

A quantização vetorial (QV), técnica bastante utilizada em compressão de sinais, consiste em mapear vetores do sinal de entrada em um conjunto finito de vetores representativos w_i , denominados vetores-código ou vetores de reconstrução.

O conjunto $W = \{w_i; i = 1, 2, \dots, N\}$ de vetores-código é denominado dicionário, o número (N) de vetores-código do dicionário é denominado tamanho do dicionário, e o número (K) de componentes de cada vetor w_i é denominado dimensão do quantizador. A correspondente taxa de codificação, R , medida em bits por pixel (bpp), em se tratando de codificação de imagens, pode ser expressa por $R = \frac{1}{K} \log_2 N$, caso sejam usadas palavras-binárias de mesmo comprimento para representar os vetores-código.

Dentre as técnicas usadas para projeto de dicionários, o algoritmo LBG [1] destaca-se por seu amplo uso. Para o projeto de quantizadores vetoriais, outras técnicas podem ser utilizadas, como, por exemplo, algoritmos de aprendizagem de redes neurais [2].

II. ALGORITMO FSC

Na aprendizagem competitiva simples, a cada apresentação de um vetor de entrada apenas o vetor-código vencedor, w_i^* ,

(ou seja, o mais próximo do vetor de entrada), tem suas componentes atualizadas. Seja n_{tot} o número total de iterações do algoritmo e M o número de vetores da entrada x . O algoritmo competitivo simples (denotado a partir de agora simplesmente por C) consiste em:

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

Para $1 \leq m \leq M$

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

$$i^* = \arg \min d[x(m), w_i(n, m)]$$

Atualize o vencedor:

$$w_{i^*j}(n, m + 1) = w_{i^*j}(n, m) + \Delta w_{i^*j}(n, m), \text{ com}$$

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

Na descrição acima, $d[.,.]$ denota uma medida de distância. Na expressão que descreve a atualização do vencedor, $x_j(m)$ é a j -ésima componente do vetor $x(m)$, enquanto w_{i^*j} é a j -ésima componente do vencedor w_{i^*} na n -ésima iteração. Temos ainda que Δw_{i^*j} é a modificação introduzida na j -ésima componente do vencedor, $\eta(n)$ é a taxa de aprendizagem na n -ésima iteração, dada por

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

em que $\eta(1)$ e $\eta(n_{\text{tot}})$ denotam dois parâmetros do algoritmo: a taxa de aprendizagem inicial e a taxa de aprendizagem final respectivamente.

Um problema encontrado no algoritmo C é que alguns vetores-código podem ter pouca ou nenhuma chance de vencer uma competição, podendo vir a resultar em um dicionário que contenha vetores-código que sejam sub-treinados. Uma técnica para minimizar esse problema foi proposta por Krishnamurthy et al. [4], a qual consiste em introduzir “consciência” no treinamento do dicionário: um neurônio (vetor-código) que está vencendo muito deve “dar uma chance” para que outros neurônios também possam vencer. Neste trabalho, a idéia de consciência é introduzida no algoritmo C, dando origem ao algoritmo designado por FSC (*frequency sensitive C*). Na abordagem FSC, a medida de distância, \hat{d} , utilizada para determinar o vencedor é dada por

$$\hat{d}[x(m), w_i(n, m)] = f_i \times d[x(m), w_i(n, m)], \quad (2)$$

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

A equação (2) mostra que se um vetor-código é frequentemente escolhido vencedor, sua \hat{d} aumentará. Conseqüentemente, sua chance de vencer a(s) próxima(s) competição(ões) diminuirá, dando a outros vetores-código com pequeno valor de f_i a oportunidade de serem vencedores na(s) próxima(s) apresentação(ões) do(s) vetor(es) de entrada. A técnica FSC procura fazer com que todos os vetores-código sejam treinados (tenham suas componentes ajustadas) aproximadamente o mesmo número de vezes.

A seção a seguir aborda a complexidade computacional do algoritmo competitivo com consciência.

III. COMPLEXIDADE COMPUTACIONAL

A Tabela I apresenta o número de operações (multiplicações, adições, subtrações, divisões e comparações) realizadas pelo algoritmo competitivo com consciência aplicado ao projeto de dicionários, em função de K , N , M e do número de iterações realizadas pelo algoritmo FSC (n_{FSC}). Tendo em vista os resultados apresentados em [3] para a aprendizagem competitiva simples, observa-se que a complexidade computacional do algoritmo competitivo com consciência é ligeiramente superior à do algoritmo competitivo simples. De fato, fixados K , N , e o número de iterações de ambos algoritmos, o incremento percentual de multiplicações do primeiro em relação ao segundo é de $\frac{1}{K}$. Para $K=16$, por exemplo, o incremento é de 6,25%. Ambos os algoritmos realizam o mesmo número de adições, subtrações, divisões e comparações.

TABELA I
Complexidade Computacional do Algoritmo do FSC.

Op.	FSC
Mult.	$[1 + (K + (K + 1)N)M]n_{FSC} - 1$
Adi.	$[1 + (K - 1)NM + KM]n_{FSC} - 1$
Sub.	$[1 + (1 + N)KM]n_{FSC} + 1$
Div.	1
Comp.	$(N - 1)Mn_{FSC}$

IV. RESULTADOS

Foram projetados dicionários de dimensão $K=16$ (correspondendo a blocos de imagens de 4x4 pixels) e tamanho $N=32, 64, 128, 256, 512$. A imagem utilizada nas simulações foi Peppers, 256x256 pixels, 8 bpp. A Tabela II apresenta a entropia normalizada dos vetores-código [4] [5]: quanto mais próximo de um/zero, mais/menos uniforme o número de vezes que os vetores-código são atualizados ao longo do treinamento do dicionário. A Tabela mostra que o algoritmo FSC leva a valores de entropia normalizada superiores aos obtidos com o algoritmo C, o que ratifica a “consciência” introduzida pela equação 2. Um aspecto desfavorável ao algoritmo FSC em relação ao algoritmo C diz respeito à atribuição de palavras-binárias de comprimento variável, usando, por exemplo, o código de Huffman. Como a entropia normalizada para o primeiro é maior, o comprimento médio obtido para as palavras-código também é maior. Como exemplo, para $N=64$, é possível obter um comprimento médio de 0,3486 bpp e 0,3127 bpp para os vetores-código FSC e C respectivamente. Vale mencionar que o uso de palavras-binárias de mesmo comprimento implica taxa de 0,375 bpp.

A Figura 1 apresenta o desempenho dos dicionários LBG, C e FSC em termos de PSNR (dB) das imagens reconstruídas versus a taxa de codificação para Peppers. O melhor desempenho está associado ao algoritmo C. Para quase todas as taxas consideradas, os menores valores de PSNR são apontados pelos dicionários FSC.

V. CONCLUSÕES

Neste trabalho observou-se que a introdução de “consciência” no processo de aprendizagem competitiva

TABELA II
Entropia Normalizada dos vetores-código. Imagem Peppers.

N	C	FSC
32	0,878	0,934
64	0,846	0,944
128	0,861	0,945
256	0,837	0,951
512	0,818	0,943

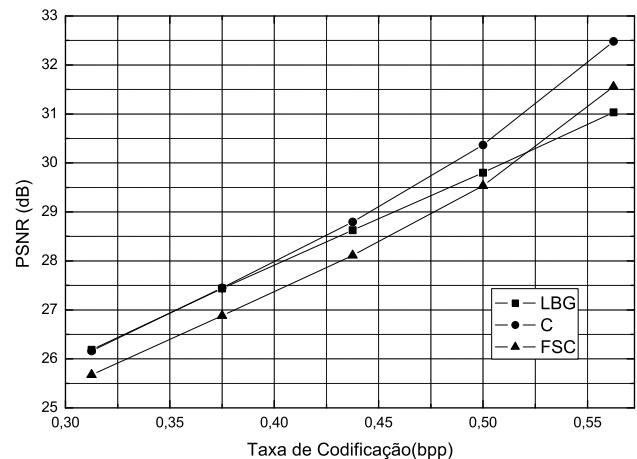


Fig. 1. PSNR (dB) versus taxa de codificação (bpp) para os dicionários LBG, C e FSC. Imagem Peppers.

aplicado ao projeto de dicionários leva a um aumento da complexidade computacional. De fato, o algoritmo competitivo com consciência (FSC) requer um maior número de operações de multiplicação, quando comparado ao algoritmo competitivo simples (C). Este aumento, no entanto, não é correspondido por uma melhora na qualidade dos dicionários projetados (avaliado em termos de PSNR das imagens reconstruídas). Os dicionários LBG em geral também levam a imagens reconstruídas com valor de PSNR superior aos obtidos pelas imagens reconstruídas com dicionários FSC. Em geral, é possível obter um menor comprimento médio para as palavras-código atribuídas aos vetores de reconstrução quando são usados dicionários C em vez de FSC.

REFERÊNCIAS

- [1] 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.
- [2] Yair, E., Zeger, K., and Gersho, A., “Competitive Learning and Soft Competition for Vector Quantization Design”, *IEEE Transactions on Signal Processing*, vol. 40, No. 2, pp 294–309, February 1980.
- [3] Madeiro, F., Lopes, W. T. A., Aguiar Neto, B. G., Alencar, M. S., “Complexidade Computacional de um Algoritmo Competitivo Aplicado ao Projeto de Quantizadores Vetoriais”, *Learning and Nonlinear Models*, vol. 2, No. 1, pp 172–186, 2004.
- [4] 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.
- [5] 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.