

SOM Hierárquico Aplicado à Compressão de Imagens

José A. F. Costa¹, José M. Barbalho², A. Duarte D. Neto², Márcio L. A. Netto¹

¹DCA - Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas - 13083-970 - Campinas - SP

²Laboratório de Engenharia de Computação e Automação - Departamento de Engenharia Elétrica
Universidade Federal do Rio Grande do Norte - 59.072-970 - Natal - RN

E-mails: {costa, marcio}@dca.fee.unicamp.br, {marinho, adriao}@leca.ufrn.br

Abstract

The increase of the need for image storage and transmission in computer systems has grown the importance of signal and image compression algorithms. The approach involving vector quantization (VQ) relies on designing a finite set of codes which will substitute the original signal during transmission with a minimal of distortion. Algorithms such as LGB and SOM work in an unsupervised manner toward finding a good codebook for a given training data. However, the number of code vectors (N) needed for VQ increases with the vector dimension, and full-search algorithms such as LGB and SOM can lead to large training and coding times. An alternative for reducing the computational complexity is the use of a tree-structured vector quantization algorithm. This paper presents an application of an hierarchical SOM for image compression in which reduces the search complexity from $O(N)$ to $O(\log N)$, enabling a faster training and image coding. Results are given for conventional SOM, LBG and HSOM, showing the advantage of the proposed method.

1. Introdução

A importância de algoritmos de compressão de imagens é crescente, devido a vários fatores como: aumento do número de equipamentos de transmissão e armazenamento de imagens, redução de custos na transmissão da informação, massiva utilização na internet; geração de imagens a partir de softwares de simulação numérica, etc. Compressão de imagens é essencial para aplicações como vídeo conferência, multimídia, armazenamento de imagens e transmissão ao longo de canais limitados em banda. O principal objetivo da compressão de imagens é reduzir a taxa de bits para transmissão ou armazenamento mantendo um nível aceitável na qualidade da imagem [1-2]. São diversas as técnicas de compressão de imagens. Essencialmente podemos classificar as técnicas de compressão sem perdas e as técnicas de compressão com perdas. As técnicas de compressão livre de perdas por não possibilitarem elevadas taxas de compressão são adequadas a aplicações onde a informação tenha que ser mantida em toda a sua fidelidade. Já as técnicas

de compressão com perdas possibilitam elevadas taxas de compressão mas a distorção resultante deve ser minimizada [1-2].

Dentre os diversos métodos de compressão com perdas a etapa de quantização é de fundamental importância. Técnicas de quantização vetorial são bem mais eficientes do que quantização escalar, quando aplicadas a problemas de dimensões elevadas como é o caso da compressão de imagens [3-5,10]. O processo de quantização vetorial requer o projeto de um conjunto finito de vetores referência (codebook) que será utilizado para substituir partes da imagem, durante a transmissão, com o mínimo de erro ou distorção em relação à imagem original.

A aplicação do algoritmo SOM (self-organizing map) convencional na compressão de imagens já é uma técnica bastante conhecida na literatura [8, 10, 12], com uma boa performance no que diz respeito a taxa de compressão, mas com uma complexidade de busca na ordem $O(N)$ e elevado tempo de treinamento, comprometendo em algumas aplicações o seu uso. Neste trabalho utiliza-se um SOM hierárquico (HSOM) [7,13], com o intuito de reduzir a complexidade de busca para ordem de $O(\log N)$, possibilitar um treinamento mais rápido, permitir uma abordagem de multi-resolução, e assim disponibilizar uma efetiva técnica de compressão vetorial aplicada a compressão de imagens.

A figura 1 ilustra todo o processo de compressão / descompressão com quantização vetorial tratado neste artigo, onde o algoritmo HSOM dá início o processo de compressão gerando o "codebook" da imagem a ser comprimida. Em seguida, toda imagem original é quantizada por um conjunto finito de vetores referência do "codebook". Após a quantização vetorial pode-se aplicar métodos de codificação como o algoritmo de Huffman, que possibilita compressão de acordo com informações estatísticas da mensagem. Com isso, uma representação comprimida da imagem será obtida, sendo armazenada ou transmitida. O processo de reconstrução inicia-se com a decodificação e, posteriormente, obtenção da imagem reconstituída a partir do conjunto de vetores referência (codebook).

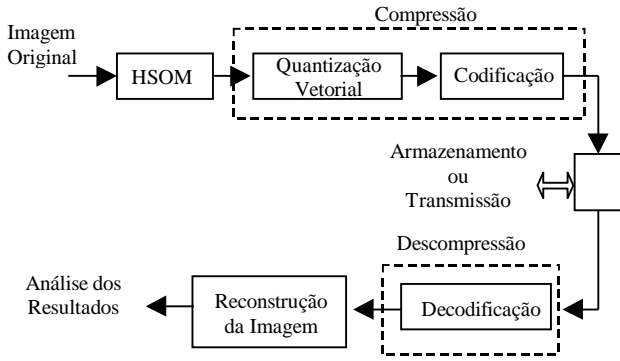


Figura 1. Descrição do processo de compressão / descompressão de imagens.

Este artigo está organizado da seguinte forma: a seção 2 descreve sobre a quantização vetorial, enfocando o algoritmo básico do SOM, enquanto que a seção 3 descreve o algoritmo do HSOM. A seção 4 apresenta os critérios de qualidade que serão utilizados para avaliar quantitativamente as imagens reconstituídas. Resultados são apresentados na seção 5. Análises e conclusões finais são apresentadas na seção 6.

2. Quantização Vetorial e o Algoritmo SOM

O processo de quantização vetorial Q é um mapeamento de vetores $\mathbf{x} \in \mathfrak{R}^n$ para vetores $\mathbf{m}_i (i = 1, 2, \dots, N) \in W$, um subconjunto finito de \mathfrak{R}^n [3-5].

$$\mathbf{x} \in \mathfrak{R}^n \rightarrow \mathbf{m}_i \in W \quad (1)$$

O projeto do conjunto de vetores referência, ou codebook, é feito utilizando um algoritmo de treinamento, como por exemplo LBG [5] ou SOM [14]. Na quantização vetorial, cada amostra do sinal de entrada é comparado com um conjunto de N vetores referência \mathbf{m}_i . O padrão \mathbf{x} é mapeado para o vetor referência com menor distância, ou maior similaridade. O espaço de atributos n -dimensional é particionado em N células convexas e não sobrepostas, onde cada uma é representada por um único vetor \mathbf{m}_i , como ilustra a Figura 2.

O algoritmo mais conhecido para quantização vetorial é o LBG (Linde-Buzo-Gray), ou k-means, que iterativamente re-aloca os centros dos agrupamentos de acordo com a pertinência das amostras, dada pela distorção mínima, geralmente a distância Euclidiana [5]. Porém, o desempenho do LBG depende de fatores como a escolha dos vetores iniciais e a solução pode não ser a ótima, devido a problemas de mínimos locais.

O mapa de Kohonen (SOM) efetua a quantização vetorial de um espaço \mathfrak{R}^n em um espaço discreto de vetores referência, ou neurônios, geralmente dispostos na forma de em um vetor ou uma matriz [13]. O SOM

provê redução de dimensionalidade ao mesmo tempo em que tenta preservar ao máximo a topologia do espaço de entrada.

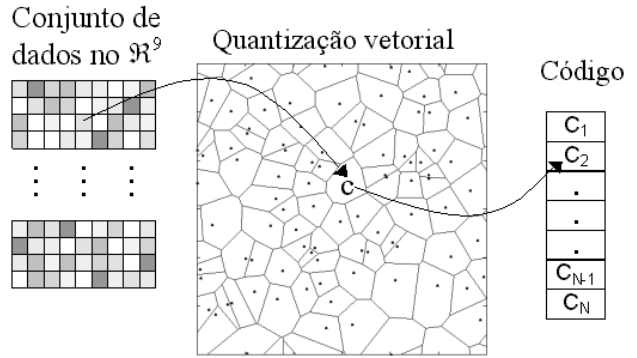


Figura 2. Quantização Vetorial.

Cada neurônio i , ou vetor referência, é representado por um vetor de pesos n -dimensional, $\mathbf{m}_i = [m_{i1}, m_{i2}, \dots, m_{in}]^T$ onde n é igual a dimensão dos vetores de entrada. Na fase de treinamento, para cada padrão de entrada é encontrado um neurônio vencedor usando o critério de maior similaridade,

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \quad (2)$$

onde $\|\cdot\|$ é a medida de distância. Os pesos do neurônio vencedor, bem como os pesos dos neurônios compreendidos em sua vizinhança, são atualizados de acordo com a regra,

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \quad (3)$$

onde t indica a iteração, $\mathbf{x}(t)$ é o padrão de entrada fornecido de forma aleatória na iteração t e $h_{ci}(t)$ é o núcleo de vizinhança em torno da unidade c vencedora [14].

Uma variação recente do SOM convencional é o algoritmo SOM em lote (batch), que torna insensível a seqüência de apresentação dos dados em cada época. As contribuições de cada padrão são acumuladas e ao final de cada época que é feita a atualização dos pesos [14]. Este algoritmo tem provado ser bastante eficiente [9, 14] e será utilizado nas simulações neste trabalho.

3. Algoritmo HSOM

A principal motivação do uso de uma estrutura hierárquica para quantização vetorial é a redução do esforço computacional que se faz necessário para encontrar o neurônio vencedor em cada iteração, no treinamento, e também na fase de execução, quando na codificação da imagem. O modelo utilizado neste trabalho é similar ao descrito em Lampinen e Oja [13], e denominado aqui de HSOM (SOM Hierárquico).

O HSOM é uma extensão do SOM convencional. Define-se uma estrutura de árvore, onde cada nó é um SOM, treinado com um determinado conjunto de dados. O mapa no nível 1 é treinado com o conjunto completo de dados, e de acordo com a quantização de cada neurônio, os mapas filhos são treinados com subconjuntos deste. Geometricamente, as células do diagrama de Voronoi do mapa em um nível k são particionadas em um nível $k + 1$, havendo um aumento da resolução (detalhamento) a cada nível em que caminhamos na direção do mapa raiz aos neurônios folhas, i.e., que estão na base da árvore. A figura 3 ilustra a configuração (simplificada) do HSOM.

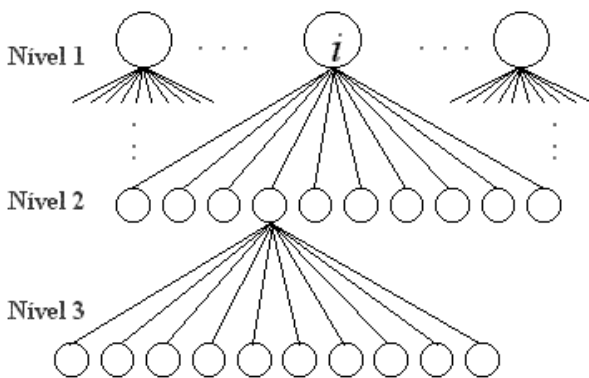


Figura 3. Configuração do SOM Hierárquico.

O treinamento do HSOM é executado sequencialmente, i.e., do nível mais elevado da árvore até o nível mais baixo. O algoritmo básico de treinamento proposto é descrito a seguir:

1. Define-se a estrutura da árvore: número máximo de níveis (nm), número de neurônios em cada nível ou regra de produção da árvore, tamanho da vizinhança, e método de inicialização de pesos em cada sub-mapa.
2. Nível atual = 1.
3. Apresenta-se um padrão de entrada x , $x \in R^n$, selecionado aleatoriamente de todo o conjunto de padrões.
4. Uma função de ativação é usada para calcular o estado de cada neurônio em relação ao padrão x no nível hierárquico mais elevado.
5. Seleciona-se o neurônio vencedor, \bar{w}_i^* .
6. Os pesos sinápticos do neurônio vencedor, \bar{w}_i^* , com também dos neurônios que estão dentro da vizinhança de \bar{w}_i^* são atualizados de acordo com a regra básica do SOM.
7. Repita os passos 3-6 até completar o fase de organização do mapa no nível atual.
8. Particione o conjunto de dados utilizado para treinar o mapa no nível atual gerando sub-conjuntos. Cada sub-conjunto S_i é relativo aos padrões que foram mapeados no neurônio i do nível atual.

9. Para cada neurônio que foi mapeado com mais de um padrão, crie uma sub-rede e treine, de acordo com o algoritmo do SOM, utilizando o sub-conjunto de dados que foi mapeado no neurônio pai, do mapa com nível hierárquico imediatamente inferior, como descrito nos passos 3 a 6.
10. Execute o passo 9 até atingir o nível máximo da rede, definido no início do algoritmo.

Nota-se, pela figura 3, que há uma redução significativa no esforço computacional. Por exemplo, caso tenhamos mapas unidimensionais com 10 neurônios e 3 níveis, teremos na base da árvore um total de 1000 neurônios. Para um dado padrão x , a busca pelo neurônio vencedor que será utilizado para representar o sinal, inicia-se no nível 1, i.e., 10 neurônios. A busca continua no nível 2, no mapa filho do neurônio vencedor i do mapa raiz, i.e., mais 10 neurônios, e assim sucessivamente. Desta forma, o espaço de busca, ou número de neurônios a serem computadas as distâncias e detecção do vencedor caíram de 1000 para 30, o que faz aumentar significativamente a eficiência do método em problemas que necessitem minimizar o tempo de codificação de sinais, como no caso de transmissão de imagens. Em relação ao tempo de treinamento também teremos uma grande redução, em relação a utilização de um mapa 1×1000 (por exemplo), pois cada mapas são menores e são treinados a partir de subconjuntos de dados. Assim, com esta estrutura obteremos tempos de treinamento próximos a de uma rede SOM unidimensional com poucos neurônios (ex. 10) e qualidade na imagem reconstruída equivalente a uma rede SOM unidimensional com muitos neurônios (ex. 100).

4. Medida da Qualidade da Imagem

A presença de distorção na imagem reconstituída será inevitável, tendo em vista que foi realizado um processo que acarreta em perda de elementos constituintes da imagem original, como é o caso em qualquer processo de compressão envolvendo quantização. Duas classes gerais de critérios são utilizados para avaliar esta distorção: 1) critérios de fidelidade objetivos, isto é, medidas quantitativas, e 2) os critérios de fidelidade subjetivos baseados em análise qualitativa [2].

A relação sinal-ruído média quadrática (SNR) e a raiz do erro média quadrática (E_{rms}), são definidos, respectivamente, por [2]:

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{I}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{I}(x, y) - I(x, y)]^2} \quad (4)$$

$$E_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{I}(x,y) - I(x,y)]^2 \right]^{1/2} \quad (5)$$

onde $I(x, y)$ representa a imagem original e $\hat{I}(x,y)$ a imagem restaurada. Estes parâmetros são comumente utilizados como critério de fidelidade objetivo.

A análise qualitativa pode ser obtida a partir das avaliações subjetivas de um ou mais observadores humanos com relação a uma ou mais imagens típicas [2].

5. Resultados

A plataforma de implementação utilizada foi um computador pessoal com processador Intel Pentium III com 667 MHz. Os algoritmos HSOM, SOM e LBG foram implementados em Matlab, utilizando-se também rotinas do SOM Toolbox [15]. Utilizamos a imagem da Lena, com tamanho de 512 x 512 pixels e com 256 níveis de cinza, ou 8 bits/pixel (*bpp*). Utilizamos um HSOM com dois níveis, e cada mapa possuía estrutura unidimensional com 10 neurônios. A inicialização dos mapas foi linear [14] e o número máximo de épocas foi 500. O algoritmo utilizado foi o SOM em lote, e a vizinhança inicial correspondeu a 80% do tamanho do mapa, enquanto que o raio de vizinhança final foi 1. A imagem foi escalonada em níveis de cinza para a faixa de valores entre 0 e 1 e para treinamento foram coletadas 3000 amostras obtidas aleatoriamente a partir da imagem Lena. O tamanho das amostras utilizado tanto no treinamento quanto na codificação foi 3 x 3, obtendo-se um espaço de entrada 9-dimensional.

A figura 4 mostra a imagem original de Lena usada nas simulações, enquanto que a figura 5 e 6 ilustram os resultados obtidos para o algoritmo LBG com 10 e 100 vetores referência.



Figura 4: Imagem Original.



Figura 5: Resultado para LBG com 10 vetores referência.



Figura 6: Resultado para LBG com 100 vetores referência.



Figura 7: SOM com 10 neurônios

Uma análise qualitativa das imagens nos mostra que tanto o LBG como o SOM com 10 neurônios, figuras 5 e 7 apresentam significativas perdas com relação a imagem original. Com 100 neurônios (ou vetores referência) todos os métodos apresentam distorções menores, a diferença como veremos pela análise quantitativa, mostra que uma melhor performance do

HSOM, principalmente no referente ao tempo de processamento.



Figura 8: SOM com 100 neurônios



Figura 9: SOM hierárquico com 2 níveis, com 10 neurônios em cada mapa.

A tabela 1 mostra uma análise quantitativa das imagens reconstituídas para as diversas configurações. As medidas são realizadas através da razão sinal ruído, SNR, e a raiz do erro médio quadrático, E_{rms} .

Tabela 1: Medidas de Qualidade.

	SNR	E_{rms}
LBG (10)	55,11	0,0673
LBG (100)	66,63	0,0612
SOM (10)	56,90	0,0657
SOM (100)	68,06	0,0605
HSOM	70,72	0,0593

Nota-se pela tabela 1, como esperado, que tanto o SOM convencional e do LBG, ambos com 10 vetores referência, são inferiores aos outros métodos em estudo. Enquanto isso, o método do HSOM é o que apresenta as melhores medidas, de tal forma que sua utilização na obtenção de “codebook” irá minimizar as distorções inerentes ao processo de quantização vetorial.

A grande vantagem do método hierárquico é em relação ao tempo de treinamento e de codificação. A tabela 2 mostra o tempo de treinamento, em segundos, obtido na execução no ambiente Matlab, para os métodos. O tempo de treinamento de um software desenvolvido em uma linguagem, por exemplo C, é cerca de 10 vezes menor.

Tabela 2: Tempo de Treinamento (segundos).

Métodos	Tempo (s)
LBG (10 neur.)	9,12
LBG (100 neur.)	171,20
SOM (10 neur.)	34,38
SOM (100 neur.)	464,34
HSOM	69,09

Verifica-se que o tempo total de treinamento do HSOM, uma estrutura com 11 mapas, 1 raiz e 10 submapas, todos unidimensionais com 10 neurônios, totalizando 100 neurônios na base, i.e., para efetuar quantização, é apenas cerca de 2 vezes o tempo de treinamento de um mapa unidimensional com 10 neurônios, e é também bastante inferior ao LBG com 100 vetores referência. O ganho de tempo estende-se também na fase de codificação, onde busca-se o vencedor em um espaço com 20 neurônios (10 em cada nível) ao invés de 100 (SOM convencional).

Outro aspecto importante nesta configuração hierárquica é a possibilidade de se trabalhar com uma abordagem de multi-resolução. Ou seja, em aplicações onde a presença de distorção é aceitável pode-se trabalhar com estruturas hierárquicas com poucos níveis e poucos neurônios. Já em aplicações que demandem excelentes imagens resultantes do processo de compressão, pode-se utilizar muitos níveis hierárquicos e muitos neurônios por nível.

Com relação à capacidade de compressão pode-se constatar que o sistema pode atingir taxas elevadas de compressão. No exemplo apresentado, obtivemos uma taxa de 0,66 *bits por pixel*, o que corresponde a cerca de 8,25% do tamanho original da imagem, considerando que foi aplicado a codificação de Huffman sobre a imagem quantizada e não considerando o espaço necessário para armazenar o codebook.

6. Conclusões

Neste trabalho foi proposto um método de compressão tendo como base o algoritmo HSOM. A finalidade principal desta configuração é reduzir o esforço computacional necessário ao treinar uma rede de Kohonen convencional com um número de neurônios elevado, devido à busca, a cada iteração, pelo neurônio vencedor.

Métodos como o LBG e o SOM convencional também foram avaliados quantitativamente, e os

resultados mostram que o tempo de treinamento, bem como o tempo de processamento de todo processo de compressão / descompressão da imagem de Lena, são bastante reduzidos no modelo hierárquico proposto. Além disso, não houve perda de qualidade do HSOM considerando o SOM convencional com 100 neurônios.

Trabalhos futuros incluirão a análise da distorção em conjuntos de dados sintéticos e de imagens para uma variedade de mapas, com ênfase especial na implicação da escolha dos parâmetros no resultado final.

A estrutura rígida da árvore, como apresentada neste artigo, pode ser flexibilizada de várias formas, como por exemplo em Costa [9]. Pode-se utilizar informações estatísticas dos subconjuntos de dados para escolher parâmetros dos sub-mapas a serem criados em níveis mais elevados, e estratégias de poda podem também ser utilizados para eliminar ramos da árvore com importância menor.

Em termos de processamento de imagens, pode-se ainda desenvolver um sistema de restauração de imagens visando melhorar a qualidade da imagem resultante do processo de compressão. Assim, espera-se obter melhores resultados na qualidade final da imagem reconstruída aumentando ao máximo a taxa de compressão e reduzindo o esforço computacional para codificação, com o objetivo de melhorar o desempenho de aplicações que necessitam de transmissão de imagens em canais limitados em banda.

Agradecimentos: J. Costa e M. Netto agradecem à Fapesp (proc. 99/09334-0).

Referências

- [1] J. D. Gibson et al. *Digital Compression for Multimedia Principles & Standards*. Morgan Kaufmann Publishers. 1998.
- [2] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Reading, MA, Addison-Wesley, 1992.
- [3] G. Burel and I. Pottier. Vetor Quantization of Images using Kohonen Algorithm. Theory and Implementation. *Revue Technique Thomson-CSF*, vol. 23, No. 1, March, 1991.
- [4] C. Chan and L. Po. A Complexity Reduction Technique for Image Vector Quantization. *IEEE Trans. on Image Processing*, vol. 1, no. 3, 1992.
- [5] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. on Information Theory*, vol. 44, pp. 2325-2383, 1998.
- [6] P. Koikkalainen. Progress with the Tree-Structured Self-Organizing Map. In: *Proc. of the 11 th European Conference on Artificial Intelligence*, pp. 211-215, 1994.
- [7] Miikkulainen, R. Script Recognition with Hierarchical Feature Maps. *Connection Science*, pp. 83-101, 1990.
- [8] J. Kangas, and T. Kohonen. Developments and applications of the self-organizing map and related algorithms. *Mathematics and Computers in Simulation*, vol. 41, pp. 3-12, 1996.
- [9] J. A. F. Costa. *Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis*. Tese de Doutorado, Campinas, SP, 1999.
- [10] R. D. Dony and S. Haykin. Neural Network Approaches to Image Compression. *Proc. of the IEEE*, Vol. 83, No. 2, 1995.
- [11] S. Haykin. *Neural Networks – A Comprehensive Foundation*. IEEE Press, Englewood Cliffs –NJ, 1994.
- [12] F. Madeiro, R. Vilar and B. G. Aguiar Neto. A self-organizing algorithm for image compression. In: *Proc. of the V Brazilian Symp. on Neural Networks*, IEEE Computer Society, pp. 146-150. 1998.
- [13] Lampinen, J. and Oja, E. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, v. 2, pp. 261-272, 1992.
- [14] T. Kohonen. *Self-organizing maps*, 2nd ed. Springer-Verlag: Berlim.
- [15] SOM Toolbox Team. *SOM Toolbox*, URL: <http://www.cis.hut.fi/projects/somtoolbox/>. 2000.