

Um estudo sobre a técnica de compressão de imagens utilizando Wavelets e Árvores de Zeros

Luiz Carlos Bambilra Torres¹

Hilton de Oliveira Mota²

luzbambilra@gmail.com; hmota@acad.unibh.br

Curso de Ciência da Computação – Uni-BH (www.unibh.br)

Resumo – Este artigo apresenta os resultados da avaliação de um algoritmo para compressão de imagens digitais baseado na transformada de wavelets associada à técnica da árvore de zeros. O algoritmo explora as correlações espaciais existentes entre os coeficientes dos diferentes níveis de decomposição como forma de estabelecer parâmetros de relevância. Uma codificação no nível de bits permite obter elevadas taxas de compressão associadas a baixas distorções. Resultados preliminares permitiram observar taxas de compressão da ordem de dezenas sem distorção visível na imagem reconstruída.

Palavras-chave – compressão de imagens, wavelets, árvore de zeros.

Abstract – This paper presents the results for a digital image compression algorithm based on the wavelets transform associated to the zerotrees technique. The algorithm explores the spatial correlations of coefficients located at adjacent decomposition levels as a way to establish relevance parameters. A bit-level coding scheme allows to obtain high compression levels and low distortions. Preliminary results showed that it is possible to achieve compression rates in the order of tens without visible distortion in the reconstructed data.

Keywords – image compression, wavelets, zerotrees.

1 INTRODUÇÃO

A crescente demanda pelo armazenamento e transporte de informações de multimídia, tais

como imagens, vídeos e áudio, tem exigido o desenvolvimento de técnicas para otimizar a sua representação de modo a diminuir o volume ocupado pelos dados e permitir o uso eficiente dos canais de comunicação e armazenamento disponíveis. Os algoritmos de compressão de dados desempenham um papel importante neste meio, porque somente através destes é possível lidar com as restrições tecnológicas e financeiras existentes nos sistemas computacionais atuais.

Nos últimos anos a transformada de wavelets (TW) [4]-[7] tem se apresentado como uma poderosa ferramenta para a compressão de dados. A TW oferece vantagens sobre outros tipos de transformações devido à característica de localidade das suas funções de análise, o que permite isolar e ressaltar os detalhes de uma imagem de forma extremamente eficiente. Uma vez que tais detalhes são as informações mais relevantes para obtenção de uma reconstrução fiel, a TW otimiza naturalmente o processo de representação.

Várias estratégias de compressão baseadas na TW foram propostas na literatura [8][9]. Estas diferem entre si, sobretudo pela forma como os parâmetros de relevância para os coeficientes são determinados. Neste trabalho uma estratégia de codificação baseada em correlações espaciais foi investigada. O algoritmo utiliza informações a respeito do posicionamento dos coeficientes de wavelets em diferentes níveis de decomposição como forma de estabelecer a relevância de cada um. Isto permite que coeficientes de baixa amplitude sejam eliminados da codificação, enquanto os de alta amplitude, normalmente associados aos detalhes da imagem, sejam codificados. O processo de seleção pode ser realizado dinamicamente sem que haja perda da informação, até que se obtenha um nível de

compressão pré-estabelecido ou uma distorção aceitável na reconstrução. Após o processo de seleção os coeficientes são codificados também dinamicamente por meio da geração de padrões binários, que permitem representar a informação de forma extremamente compacta. O algoritmo foi avaliado utilizando-se imagens-padrão normalmente empregadas para tal propósito. Durante o processo observou-se taxas de compressão da ordem de dezenas de vezes sem a introdução de distorções visíveis a olho nu.

2 A TRANSFORMADA DISCRETA DE WAVELETS

A transformada discreta de wavelets (TDW) é uma transformação ortogonal baseada em funções de suporte compacto denominadas wavelets. Devido às características das suas funções de análise, a TW possui a propriedade singular de realizar a decomposição de sinais em frequência enquanto mantém, até certo ponto, a sua informação temporal. Fazendo uma analogia com a transformada de Fourier, imagine uma música tocada de um violino em que cada nota representa uma frequência e as notas variam de acordo com tempo. A transformada de Fourier permite determinar com exatidão cada frequência contida na música, mas não a sequência em que as notas são geradas. Já a TDW permite determinar, além da frequência, a localização temporal de cada nota.

A TDW consiste na decomposição do sinal em níveis, cada um contendo uma versão de menor resolução e um conjunto de detalhes relativos ao nível anterior. Este procedimento é mostrado de forma esquemática na Fig.1, onde S representa o sinal a ser processado e utiliza-se a wavelet de Haar. A convolução de S com os coeficientes da wavelet geram duas saídas, uma contendo os detalhes, identificados como H , e a outra contendo a aproximação de menor resolução, identificada como L . Para gerar um segundo nível de decomposição aplica-se a mesma operação na aproximação e, assim, sucessivamente até que se atinja o nível máximo desejado.

Quando aplicada ao processamento de imagens, a TDW segue o mesmo princípio,

entretanto a decomposição resulta em uma componente de aproximação e três componentes de detalhes em cada nível, como mostrado na Fig.2. Primeiramente realiza-se a convolução entre a imagem e a wavelet, o que resulta em

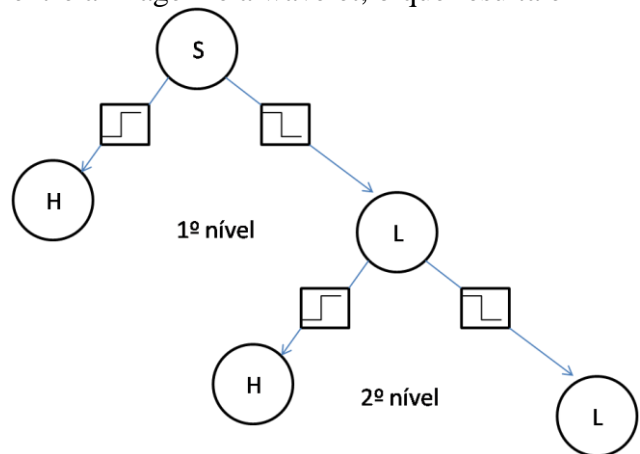


Figura 1. Representação esquemática de uma decomposição de TDW em dois níveis.

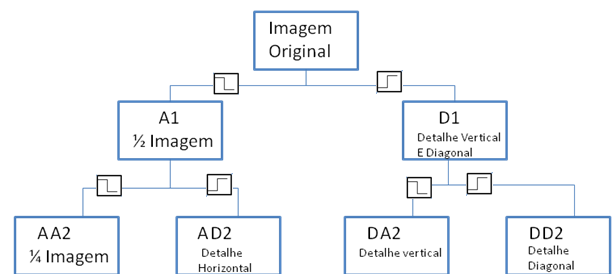


Figura 2. Representação esquemática da decomposição de uma imagem em um nível.

duas saídas, representadas por $A1$ e $D1$. Posteriormente, realiza-se a convolução entre $A1$ e a wavelet, o que resulta nas saídas $AA2$ e $AD2$, contendo a aproximação e os detalhes horizontais da imagem, respectivamente. Por fim, realiza-se a convolução entre $D1$ e a wavelet, gerando-se as matrizes $DA2$ e $DD2$ que contêm os detalhes verticais e diagonais. Para obter-se o segundo nível de decomposição basta aplicar o mesmo procedimento a $AA2$.

A Fig.3 apresenta um exemplo de decomposição de uma imagem digital em um nível utilizando a wavelet de Haar. A Fig.3 (a) mostra a imagem original e a Fig.3 (b) as quatro sub-imagens relativas à aproximação e aos detalhes do primeiro nível. A sub-imagem superior esquerda representa a aproximação, a superior direita os detalhes verticais, a inferior

esquerda os detalhes horizontais e a inferior direita os detalhes diagonais.



Figura 3. a) Imagem original. b) Sub-imagens resultantes da decomposição por transformada de wavelets.

3 ALGORITMO DA ÁRVORE DE ZEROS

O algoritmo Árvore de Zeros [1] utiliza-se de um sistema de mapeamento para codificar uma imagem através de símbolos de tal forma que o resultado ocupa menos bits do que outros métodos de compressão conhecidos. Essa característica o torna um poderoso algoritmo de compressão de imagens. Apoiando-se no fato de que a TDW produz uma sub-imagem de menor resolução a cada nível de decomposição, o algoritmo realiza uma varredura buscando estabelecer relações de significância entre os coeficientes que ocupam as mesmas posições relativas em cada nível. Por exemplo, se a região próxima ao ombro da moça mostrada na Fig. 3(a) resultar em coeficientes de baixa amplitude há uma alta probabilidade de que os coeficientes localizados na mesma região em níveis inferiores sejam também de baixa amplitude [2] [3]. Este relacionamento é representado esquematicamente na Fig. 4(a). Portanto, existe uma redundância entre estes coeficientes que pode ser explorada visando reduzir a quantidade de informação necessária para reconstruir a imagem.

As relações de significância entre coeficientes são estabelecidas com base em suas posições relativas em cada nível. Isto é representado na Fig. 4(b), onde é possível notar que um coeficiente localizado no nível H3 corresponde a quatro coeficientes em H2 e dezesseis coeficientes em H1. Através da

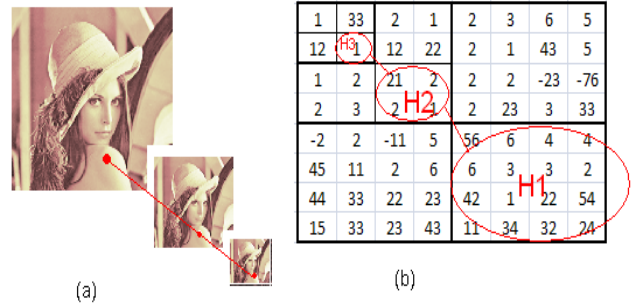


Figura 4. a) Três níveis diferentes de resolução da fig.3 (a). b) Representação de coeficientes de três níveis de decomposição.

similaridade e localização relativa entre coeficientes de diferentes níveis é possível agrupá-los em uma estrutura chamada árvore de zeros e codificar esse agrupamento com somente um símbolo.

A significância de um coeficiente é estabelecida comparando-o com um limiar determinado por

$$L(x) = 2^{\rho(10E_2(|MAX(f(x)|)))} \quad (1)$$

onde ρ é um operador de arredondamento para o inteiro mais próximo, $f(x)$ é o coeficiente da imagem e MAX o coeficiente de maior magnitude encontrado após a decomposição. Uma vez estabelecido o limiar inicial, o algoritmo realiza uma codificação em duas etapas. Na primeira, denominada dominante, faz-se uma varredura comparando-se cada coeficiente com o limiar e codificando-os de acordo com quatro símbolos:

- P = coeficiente significativo positivo, ocorre quando $|x| \geq L(x)$ e $x > 0$;
- N = coeficiente significativo negativo, ocorre quando $|x| \geq L(x)$ e $x < 0$;
- T = coeficiente insignificante, ou seja, $|x| < L(x)$, e cujos descendentes são também insignificantes. Estes são denominados raízes de árvores de zeros.
- Z = coeficiente insignificante, mas que possui descendentes significantes. Estes são denominados zeros isolados.

Na segunda etapa, denominada subordinada, o algoritmo refina os coeficientes gerados no primeiro passo de acordo com a seguinte regra:

para cada coeficiente considerado significativo calcula-se o seu módulo e verifica-se se o BMS (bit mais significativo) é igual ao de $L(x)$. Caso positivo, o coeficiente é codificado com o símbolo “1”, caso contrário com símbolo “0”.

Como exemplo, a Fig.5 apresenta uma matriz numérica representativa da decomposição de uma imagem de 8 por 8 pixels em três níveis. A varredura da imagem é feita a partir do último nível de decomposição em direção ao primeiro, no sentido mostrado pela seta na Fig.5 (a). O coeficiente de maior magnitude neste exemplo é igual a 70 o que, de acordo com a Eq.1, resulta em um limiar inicial $L(x) = 64$. Como 70 é maior que o limiar e é positivo, esse coeficiente é codificado com o símbolo P , como mostrado na Fig.5 (b). O próximo coeficiente, cujo valor é -68, é codificado com o símbolo N . O terceiro coeficiente possui valor 12, que é menor do que o limiar. Entretanto, ele deve ser considerado um zero isolado e codificado com o símbolo Z , uma vez que possui um descendente significativo de valor 67 em sua estrutura de árvore. O último coeficiente do primeiro nível possui valor igual a 16. Este coeficiente possui um valor inferior ao limiar e não possui nenhum descendente significativo, portanto é codificado com o símbolo T . Neste caso, toda sua estrutura de árvore é zerada, como é mostrado na Fig. 5(b).

Na etapa subordinada são verificados quais coeficientes possuem o BMS igual ao de $L(x)$. Como os coeficientes 70 e -68 possuem esse bit (em módulo), ambos são codificados com o símbolo “1”. O processo prossegue então para os coeficientes do próximo nível de decomposição, até que o penúltimo nível seja atingido.

Após a análise de todos os coeficientes o processo pode ser repetido quantas vezes necessário, sempre atualizando-se o limiar como $L_{i+1}(x) = L_i(x)/2$. Para cada iteração é guardada uma lista com os símbolos gerados nos passos dominante e subordinado. O algoritmo pode ser interrompido a qualquer momento, geralmente



Figura 5. a) Representação de coeficientes da TDW em três níveis de decomposição. b) Codificação inicial dos coeficientes do 3º nível juntamente com seus descendentes.

definido pelo usuário com base em parâmetros tais como a taxa de compressão desejada ou o nível de distorção aceitável. Com $L(x) = 1$ obtém-se uma imagem com um erro médio quadrático muito próximo de zero, relacionado basicamente aos truncamentos ocorridos na compressão.

A decodificação dos símbolos é feita em um processo reverso. O algoritmo lê a lista de símbolos gerados no passo “Subordinado” para saber se o coeficiente possui valor negativo ou positivo, se é um zero isolado ou se é insignificante junto com sua estrutura de árvore. Os símbolos do passo subordinado são acrescidos a cada iteração, fazendo com que o valor do coeficiente seja reconstruído. A Tabela I mostra a reconstrução do coeficiente 70 para cada limiar considerado.

Tabela I – Exemplo de reconstrução de um coeficiente codificado.

$L(x)$	Coeficiente	Coeficiente em binário	Símbolo Subordinado
64	64	1000000	1
32	64	1000000	0
16	64	1000000	0
8	64	1000000	0
4	68	1000100	1
2	70	1000110	1
1	70	1000110	0

4 RESULTADOS

O algoritmo da árvore de zeros foi implementado no ambiente MatLab®, sendo que quatro funções primordiais foram

implementadas na linguagem C como forma de obter-se um melhor desempenho. O compilador utilizado foi o LCC versão 2.4 integrado ao próprio MatLab®. A Fig.6 mostra a estrutura do algoritmo.

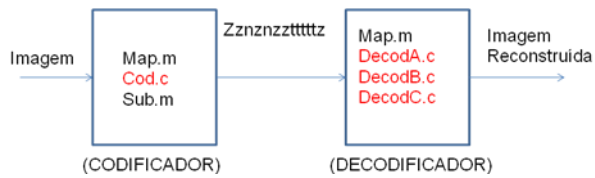


Figura 6. Estrutura do algoritmo da árvore de zeros.

A Fig.7 apresenta uma imagem utilizada durante os testes. A imagem possui 512 por 512 pontos com um mapa de cores de 256 níveis de cinza (8 bpp). O arquivo original possui um tamanho de $512 \times 512 \times 8 = 2097152$ bits.



Figura 7. Imagem com 512x512 pontos com codificada 256 níveis de cinza (8 bpp).

A Fig. 8 (a) mostra a reconstrução obtida a partir de uma taxa de compressão de 9:1. A reconstrução apresentou um erro médio quadrático igual a 11.19 dB. A Fig. 8(b) apresenta a reconstrução a partir de uma taxa de compressão de 33:1 com um erro médio quadrático de 18.69 dB.



Figura 8. (a) Imagem comprimida com taxa 9:1. (b) Imagem comprimida com taxa 33:1.

A Fig. 9 apresenta resultados da reconstrução a partir de taxas de compressão de 81:1 e 213:1. Os erros médios quadráticos são de 24.36 e 28.12 dB, respectivamente.



Figura 9. a) Imagem comprimida com taxa 88:1. b) Imagem comprimida com taxa 213:1.

O Gráfico 1 mostra a relação entre a taxa de compressão e o erro médio quadrático de oito imagens, é possível observar que taxa de compressão aumenta exponencialmente à medida que o erro médio quadrático aumenta.

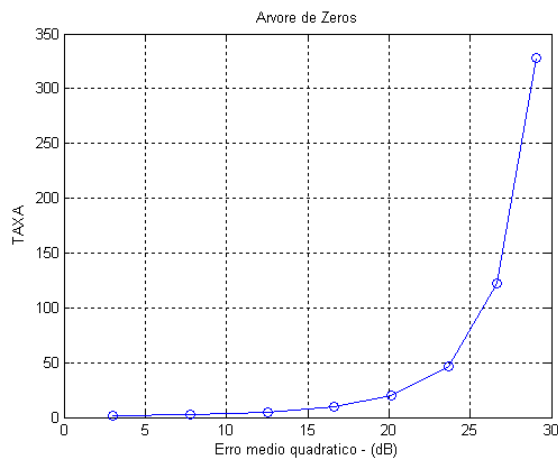


Gráfico 1. Compressão árvore de zeros, relação entre taxa e erro médio quadrático (dB).

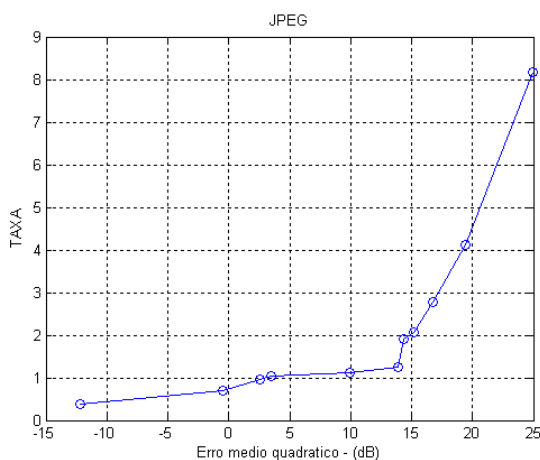


Gráfico 2. Compressão JPEG, relação entre taxa e erro médio quadrático (dB).

Comparamos o algoritmo Árvore de Zeros com o método de compressão de dados JPEG, o Gráfico 2 mostra a relação entre a taxa de compressão e o erro médio quadrático. Cabe salientar que a imagem da Fig.7 foi utilizada para a compressão tanto para a montagem do Gráfico 1 como do Gráfico 2. É possível perceber a grande diferença de compressão entre os dois métodos, sendo que o algoritmo Arvore de Zeros obteve taxas de compressão muito superiores do que o método JPEG.

5 CONCLUSÃO

Este trabalho apresentou os resultados da implementação de um método de compressão de

imagens baseado em árvores de zeros e correlação espacial dos coeficientes de wavelets entre vários níveis de decomposição. Os resultados permitiram compreender a técnica e obter uma implementação inicial, a partir da qual pretende-se estender a técnica para outras aplicações de compressão e eliminação de ruídos.

AGRADECIMENTOS

Os autores gostariam de agradecer ao Centro Universitário de Belo Horizonte - UNI-BH por divulgarem a pesquisa e criar a oportunidade de publicação dentro do ambiente universitário.

REFERÊNCIAS

- [1] Shapiro, J. M. "Embedded image coding using zerotrees of wavelet coefficients". In: IEEE Transactions on Signal Processing. Vol. 41, no. 12, pp. 3445-62 dezembro de 1993.
- [2] Mallat, S., Zhong, S. "Characterization of signals from multiscale edges". In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 48, no. 7, pp. 710-32, julho de 1992.
- [3] Abramovich, F., Bailey, T. C., Sapatinas, T. "Wavelet analysis and its statistical applications". The statistician. Vol. 49, part 1, pp. 1-29, 2000.
- [4] A Primer on Wavelets and their Scientific Applications. CRC Press, Boca Raton, 1999.
- [5] B. B. Hubbard, The world according to wavelets: the story of the mathematical technique in the making. 2nd ed., Natick, MA: A K Peters, Ltd., 1998.
- [6] S. Qian, Introduction to time-frequency and wavelet transforms. Upper Saddle River, NJ: Prentice Hall PTR, 2002.
- [7] S. Mallat, A wavelet tour of signal

processing. 2nd ed., San Diego, CA:
Academic Press, 1998.

[8] Bryan E. Usevitch, "A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000", IEEE signal processing magazine, September 2001.

[9] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, "The JPEG 2000 Still Image Compression Standard", IEEE signal processing magazine, September 2001.