

UTILIZAÇÃO DO MÉTODO WAVELETS EZW – ABORDAGEM EM COMPRESSÃO DE IMAGENS

Aldo Ventura da Silva*

RESUMO

O presente trabalho visou à elaboração de um compressor de imagens baseado em métodos wavelets, para isso foi utilizado algoritmos e códigos existentes. O algoritmo utilizado foi o *Embedded Zerotree Wavelet (EZW)*, desenvolvido por (SHAPIRO, 1993), que utiliza a transformada wavelet para seu propósito.

Buscamos entender a compressão de imagens de forma concisa, para isso buscando alguns exemplos simples e visando o contexto geral. Houve a análise do algoritmo EZW, que é a forma de codificar uma imagem utilizando wavelet, deixando-a comprimida; por meio de seqüências de zeros imposta pelo algoritmo EZW, temos a árvore de zeros que é a substituição de um pixel com algum valor válido para zero, isso visando uma regra, o que corresponderia a um valor nulo, assim obtemos a compressão.

E por ultimo temos a análise do código, utilizando duas imagens a Lena e a Bárbara, que são imagens conhecidas no campo de análise de imagens. Com elas utilizamos o Erro Médio Quadrático (EMQ), que é a medida estatística que escolhemos para

* Bacharel em Sistemas de Informação – Universidade de São Paulo (USP)

verificar a qualidade da compressão, sendo que se o EMQ aproxima-se do zero, melhor é a qualidade da compressão se comparada à imagem original. Observando a quantidade de Kilobytes e o EMQ, pudemos inferir opiniões sobre a qualidade da compressão, que se mostrou promissora em imagens em que a quantidade de Kilobytes era grande, pois manteve o EMQ muito próximo de zero em até metade dos Kilobytes da imagem Original. Portanto, o algoritmo EZW, apesar de ser o pioneiro em compressão de imagens utilizando wavelet, é ainda muito promissor no campo de compressão de imagens, mostrando ser uma ferramenta essencial para quem necessitar dessa funcionalidade.

Palavras-chave: Compressão de Imagens, Transformada Wavelets, Análise de Sinais.

1 INTRODUÇÃO

Com o desenvolvimento da tecnologia que vem crescendo vertiginosamente, temos observado uma ampla revolução no panorama mundial das telecomunicações, criando uma série de novos serviços e permitindo a melhoria dos já existentes. Uma área que tem evoluído significativamente é a de processamento digital de imagens, temos observado que a utilização de imagens digitais em serviços de telecomunicações aumentou consideravelmente, exigindo armazenamento e velocidade de processamento cada vez maiores. E com o advento da Rede Digital de Serviços Integrados de Faixa Larga (RDSI – FL) abre-se a motivação de desenvolvimentos de aplicativos e serviços para a transmissão de sinais de TV digital, HDTV (High Definition TV), vídeo-fone, vídeo-conferencia, banco de dados de imagens, etc.

Justamente por termos vantagens, como maior robustez a erros de transmissão e facilidade de manuseio, a imagem no formato digital implica em um maior volume de dados, que exige uma maior capacidade de armazenamento e de transmissão. Logo, técnicas de compressão de imagens têm sido de fundamental importância na área de processamento digital de imagens.

O objetivo deste trabalho é desenvolver um algoritmo de compressão de imagem, enfatizando os pontos importantes da teoria wavelet que ajudam no desenvolvimento deste trabalho, principalmente o método de compressão de imagens Embedded Zerotree Wavelet (EZW).

2 ABORDAGEM BASICA DE COMPRESSÃO DE IMAGENS

Imagens digitais possuem grande vantagem, mas o desafio é minimizar a quantidade de bits necessários para sua representação. Em diversos casos, elas possuem em sua representação original, uma grande quantidade de informação redundante, possibilitando a compressão dos dados, esse é o caso de algumas técnicas de compressão de imagens que visam a compressão por meio de eliminar a redundância existente na imagem original.

2.1 Objetivo Geral

Compressão é a arte ou ciência de representar dados com a menor quantidade de informação possível a uma qualidade aceitável (OLIVEIRA, 2008), aproveitando-se para isso das próprias características dos dados. Sendo assim, a compressão de imagens nada mais é do que a aplicação das técnicas de compressão a imagens digitais.

Podemos representar uma imagem como uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais, e o valor da função em cada ponto coordenado é chamado intensidade ou nível de cinza da imagem. Quando as coordenadas x e y assumem finitos valores, ou seja, quantidade discretas, diz-se que a imagem é digital. Logo, podemos notar que uma imagem digital é composta por um número finito de elementos, que possuem uma localização particular em um plano e um valor associado. Estes elementos são conhecidos como *pixels* (do inglês, *picture element*)(OLIVEIRA, 2008).

O Processo de digitalização das coordenadas é conhecido como amostragem, nesse caso, estaremos considerando uma imagem como seqüência de valores amostrados(OLIVEIRA, 2008) (2.1):

$$x[n_1, n_2], 0 \leq n_1 < N_1, 0 \leq n_2 < N_2 \quad (2.1)$$

A primeira coordenada n_1 é entendida como índice para as linhas da imagem, enquanto n_2 , como índice para as colunas. Isto é ilustrado na figura 1:



FIGURA 1 - (extraída de OLIVEIRA, 2008) – Ilustração de uma imagem digitalizada e a interpretação das coordenadas das amostras

2.2 Técnicas de Compressão de Imagens

Compressão de imagens é o processo de redução da quantidade de bits necessária para representar uma imagem. A compressão, em geral, é possível porque o número de bits realmente necessário para a representação da imagem pode ser reduzido, devido à redundância natural existente na imagem. Em geral temos os seguintes tipos de redundância (FARIAS, 2008):

- Redundância espacial → Existe uma grande correlação entre valores de pixels vizinhos.
- Redundância espectral → É a correlação entre planos de cores ou entre diferentes bandas espectrais. Este

tipo de redundância é mais facilmente identificável em imagens coloridas.

- Redundância temporal → A correlação entre os diferentes quadros de uma seqüência de imagens é denominada redundância temporal. Geralmente as imagens de uma seqüência apresentam uma grande quantidade de informação que permanece invariável de um quadro para outro. Os algoritmos de compressão de seqüências de imagens, ou seja, vídeo, tiram proveito deste tipo de redundância para obter uma compressão mais eficiente.

- Redundância de codificação → Uma codificação é considerada ótima se o número de símbolos resultante for mínimo. Este limitante fornece o número mínimo de bits por símbolo necessários para codificar uma fonte (FARIAS, 2008) citando (AMBRANSON, 1963). Uma imagem codificada apresenta uma redundância de codificação se o número de símbolos utilizados para codificá-la for maior que este limitante.

- Redundância psico-visual → O olho humano não reage com igual intensidade a todas as informações visuais contidas em uma imagem. Algumas informações têm maior importância relativa, ou seja, são psico-visualmente redundantes, podendo ser eliminadas, mesmo assim a qualidade da imagem permanece inalterada em uma percepção humana.

A redundância é uma definição matemática quantitativa (FARIAS, 2008). Sejam n_1 e n_2 os números de unidades de informação contidas em dois conjuntos de dados, que representam a mesma informação. Para aplicações em compressão de imagens, o primeiro conjunto (n_1) representa a imagem original e o segundo (n_2) representa a imagem comprimida.

A redundância de dados relativa ao primeiro conjunto é definida como (FARIAS, 2008) (2.2):

$$R_D = 1 - \frac{1}{C_R} \quad (2.2)$$

onde C_R , denominada de taxa de compressão relativa, é definida por (2.3):

$$C_R = \frac{n_1}{n_2} \quad (2.3)$$

Para o caso em que $n_1 = n_2$, $C_R = 1$ e $R_D = 0$, indicando que a primeira representação não contém redundância em relação à segunda. Quando $n_2 \ll n_1$, C_R tende a ∞ e R_D tende a 1, significando que houve uma compressão significativa. Para $n_2 \gg n_1$, C_R tende a ∞ e R_D tende a $-\infty$, nos é indicado que o segundo conjunto de dados contém um número de unidades de informação muito maior que o primeiro conjunto. Houve, portanto, uma expansão de dados, que é uma situação indesejável para a compressão. Para os casos de interesse prático, C_R e R_D estão dentro dos intervalos $(1, \infty)$ e $(0, 1)$ respectivamente (FARIAS, 2008).

2.3 Padrões de Compressão

Diversas aplicações de imagens são desenvolvidas seguindo a adoção de padrões, essa adoção ajuda a reduzir sensivelmente a custo do *hardware* dos sistemas de compressão de imagens. Os padrões relacionados à codificação e transmissão de sinais através de canais de telecomunicações são desenvolvidos com o apoio de setor de padronização em telecomunicações da União Internacional de Telecomunicações (“International Telecommunications Union” - ITU). Este setor é conhecido como CCITT. Nas três seguintes áreas, são concentrados os esforços de padronização de esquemas de compressão de imagens (FARIAS, 2008):

- Imagens de dois níveis:
Um comitê conhecido como “Joint Bilivel Imaging Group”

(JBIG) foi formado em 1988 com o objetivo de trabalhar no desenvolvimento de uma padrão de compressão e descompressão de imagens de dois níveis. A intenção era desenvolver um algoritmo que satisfizesse o padrão existente e estendesse a sua utilidade a outras aplicações.

- Imagens de tons contínuos, estáticas, monocromáticas ou coloridas.

Um comitê conhecido como “Joint Photographic Experts Group” (JPEG) foi formado no final do ano de 1986 com o propósito de desenvolver um padrão internacional para a compressão e descompressão de imagens de tons contínuos, estáticas, monocromáticas ou coloridas. O objetivo foi criar um padrão onde suas aplicações fossem as mais diversas possíveis como foto-videotexto, artes gráficas, sistemas médicos, etc (FARIAS, 2008).

- Imagens seqüenciais de tons contínuos
Desde 1988, um grupo de padronização conhecido como “Moving Picture Experts Group” (MPEG) vem trabalhando para desenvolver um padrão para armazenamento e recuperação de imagens em movimento e sons, usando meios digitais de armazenamento com uma taxa combinada de 1,0 – 1,5 Mbits/s. Atualmente, o padrão MPEG é utilizado em diversas aplicações como vídeo-texto, jogos, vídeo-mail, etc.

2.4 Elemento básico de um sistema de compressão

O sistema de compressão é a relação de um compressor e um descompressor, vejamos a figura W onde temos a concatenação dos sistemas de compressão e descompressão em termos de dois mapeamentos, que representam uma matriz de imagem, M_1 e M_2^{-1} , respectivamente. Para que a compressão seja sem perdas, exige-se que $M_2^{-1} = M_1^{-1}$. Existem, porém, casos em que M_1 não é irreversível, caso de compressão com perdas, em que

será utilizada a notação M_2^{-1} para reforçar a idéia de que o sistema descompressor é apenas uma aproximação de inversa do sistema compressor.

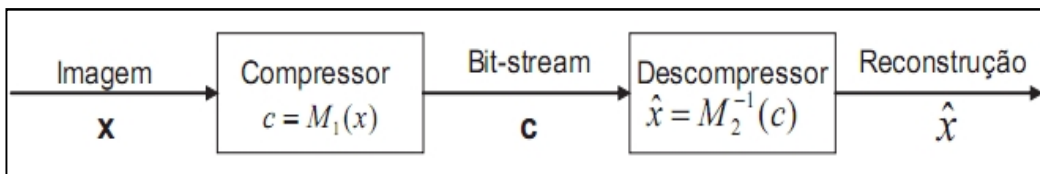


FIGURA 2 - (extraído de OLIVEIRA, 2008) – Visão global de um sistema de compressão

Detalhemos um pouco mais o sistema da figura 2. Como mostrado na Figura 3, o primeiro passo é o que transforma as amostras originais, componentes da imagem, em um novo conjunto, com características mais adequadas à compressão. Formalizando-se este passo, tem-se: $y = (T)x$ em que $y = \{y(k_1, k_2)\}$ é uma outra seqüência bidimensional e finita, tendo $K_1 K_2$ elementos, normalmente, esta operação é inversível. Assim, o descompressor é capaz de utilizar uma transformada inversa, T^{-1} , sem que qualquer distorção seja inserida nesta etapa. O segundo passo é o que permite representar, de uma maneira aproximada, a seqüência de amostras transformadas por uma seqüência de índices quantizados. Assim, obtém-se $q = Qy$, e $q \equiv q(p_1, p_2)$ representa uma seqüência bidimensional e finita com $P_1 P_2$ elementos. Geralmente, a quantidade de amostras $P_1 P_2$ não é maior e é, provavelmente, até menor que o número de amostras transformadas $K_1 K_2$. Essa etapa de quantização, sim, insere distorção no sistema de tal forma que o descompressor utiliza apenas uma inversa aproximada Q_2^{-1} . Finalmente, os índices de quantização são codificados entropicamente para formar o *bit-stream* final, que será transmitido e armazenado, por exemplo.

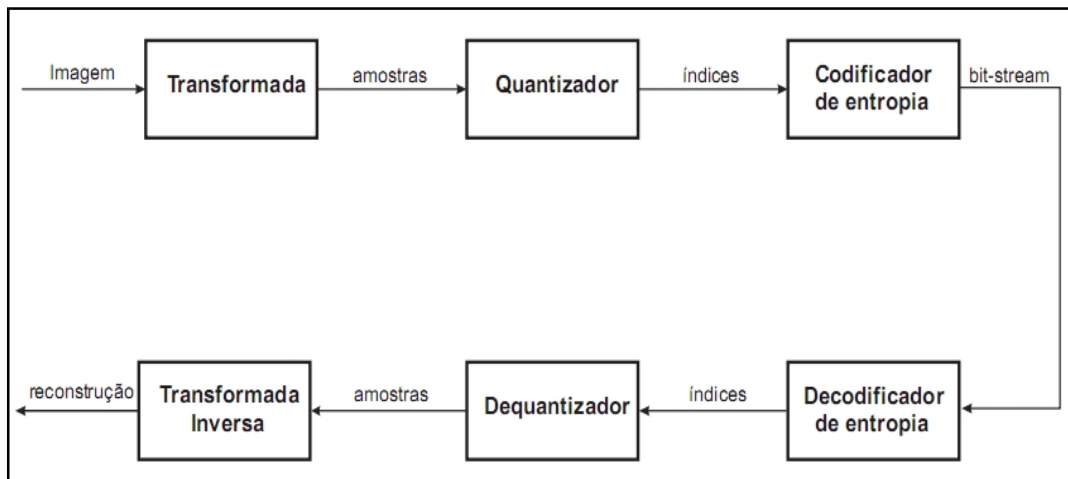


FIGURA 3 - (extraído de OLIVEIRA, 2008) – Evidenciação dos elementos de um sistema de compressão

3 O Algoritmo EZW

O algoritmo EZW implementado por (SHAPIRO, 1993), se divide em 3 processos básicos:

1. Uma Transformada Wavelet Discreta ou uma decomposição em sub-bandas hierárquicas;
2. Predição da ausência de informação significativa através das escalas pela exploração das similaridades próprias da imagem;
3. Compressão sem perda da seqüência resultante, que é obtida via codificação aritmética adaptativa (GUSMAO, 2002) citando (WITTEN et, al. 1987).

A estrutura geral do algoritmo está ilustrada na Figura 4.

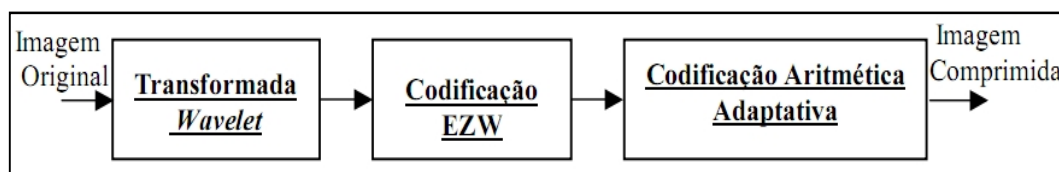


FIGURA 4 - (extraída de GUSMÃO, 2002) – Diagrama para uma codificação de imagem sem perda, baseada na EZW.

No primeiro estágio os dados da imagem original são decorrelacionados pelos bancos de filtros da transformada wavelet, para produzirem sub-bandas de correlação reduzida. Idealmente, se a transformada wavelet removesse toda correlação entre as amostras, não seria necessário codificar as sub-bandas usando *zerotrees*(GUSMAO, 2002). Depois da transformada wavelet ainda existe uma alta correlação dentro das sub-bandas e entre as sub-bandas. A correlação entre as sub-bandas é exemplificada na Figura 5.

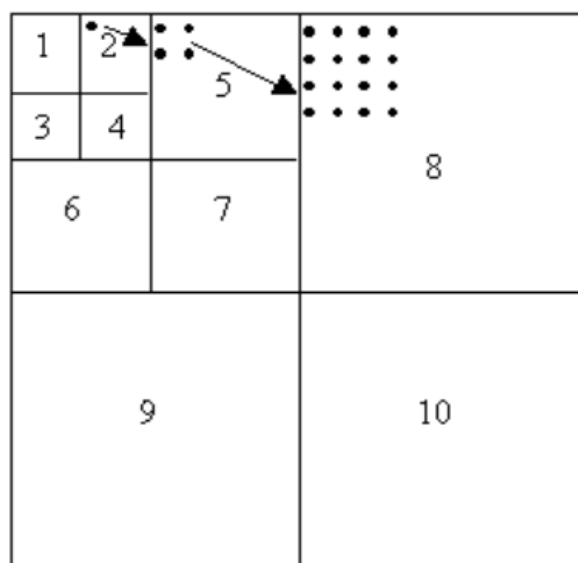


FIGURA 5 - (extraída de GUSMAO, 2002) – Correlação dos coeficientes entre as sub-bandas para 3 níveis de decomposição.

As sub-bandas de detalhamento 2, 5 e 8 são bastante correlacionadas entre si, uma vez que a sub-banda 2 é uma aproximação grosseira da sub-banda 5, e esta por sua vez, é uma aproximação grosseira da sub-banda 8. O mesmo é válido para as outras sub-bandas 3, 6 e 9; e 4, 7 e 10.

O segundo estágio da estrutura geral do algoritmo de compressão EZW explora a correlação existente entre as sub-bandas. O papel principal da codificação EZW é efetuar a quantização dos coeficientes das wavelets e reordená-los de forma que possam ser codificados com uma maior eficiência no estágio seguinte. E o terceiro estágio, consiste na codificação aritmética do algoritmo proposto por (SHAPIRO, 1993), onde é explorada a correlação residual usando técnicas de modelagem adaptativas (GUSMAO, 2002) citando (WITTEN et, al. 1987).

3.1 O Codificador EZW

O codificador EZW foi projetado para trabalhar com a transformada wavelet. Aplicando-se em imagens, a transformada wavelet transforma um sinal do domínio espacial para um domínio conjunto espaço-escala (GUSMAO, 2002). O codificador EZW realiza uma codificação progressiva, comprimindo uma imagem

dentro de um vetor de bits (*bitstream*), com um aumento progressivo de resolução. Isto significa que quanto mais bits são adicionados ao vetor, mais detalhes a imagem decodificada poderá conter. Codificação progressiva é também conhecida como codificação embutida (*embedded encoding*) (SHAPIRO, 1993). O termo embutido é usado para indicar que os dados comprimidos são ordenados na ordem de importância visual (GURMAO, 2002).

3.2 A Estrutura do Codificador EZW

Baseando-se nas seguintes observações, o codificador EZW foi projetado:

1. Imagens naturais, em geral, têm um espectro passa-baixa, ou seja, a energia se concentra nas baixas frequências. Logo, os coeficientes wavelets, em média, são menores nas sub-bandas de baixas frequências que nas de altas frequências. Dessa forma, podemos descartar as frequências mais altas, pois elas somente adicionam detalhes à imagem.
2. Coeficientes wavelet maiores são mais importantes que coeficientes menores.

Estas duas observações são levadas em consideração e exploradas pelo codificador EZW, que codifica os coeficientes na ordem decrescente de magnitude. Os coeficientes da imagem são comparados com um limiar previamente estabelecido. Se o coeficiente for maior que o limiar, ele é codificado e removido da imagem. Se for menor, é deixado na imagem para ser codificado nas iterações futuras. O artifício é usar a dependência entre os coeficientes wavelets através das diferentes escalas para eficientemente codificar grandes partes da imagem que se encontram abaixo do limiar em questão (GUSMAO, 2002). Após todos os coeficientes serem visitados (*scanned*) o limiar é reduzido e os coeficientes que estavam abaixo do limiar anterior são visitados novamente. Se agora forem maiores que o limiar atual,

eles são codificados. Caso contrário, são deixados novamente na imagem para as próximas iterações. Este processo se repete até que todos os coeficientes sejam codificados ou outro critério seja satisfeito, como por exemplo, a máxima taxa de bits seja atingida.

No sistema de sub-banda hierárquico, cada coeficiente de uma dada escala pode ser relacionado a um conjunto de coeficientes da próxima escala mais fina de orientação similar. Os coeficientes localizados na escala mais grossa, ou seja, escalas de baixa frequência são denominadas “pais”, e todos os coeficientes correspondentes à mesma localização espacial, porém, estando na escala mais fina, ou seja, de alta frequência, são chamados de “filhos”. Para um dado pai, o conjunto de todos os coeficientes em todas as escalas mais finas de orientação similar correspondendo à mesma localização espacial são denominados descendentes (GUSMAO, 2002).

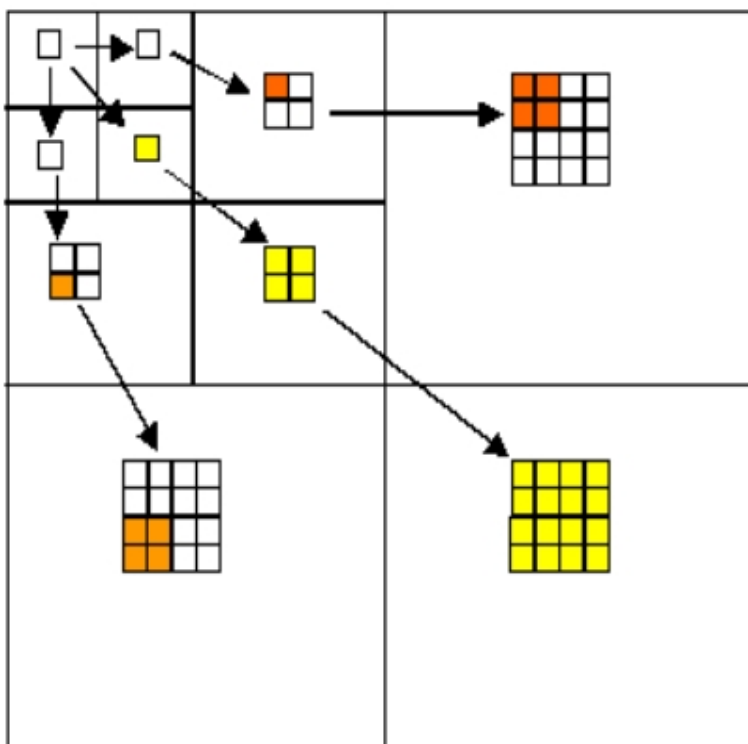


FIGURA 6 - (extraído de GUSMAO, 2002) – Relação dos Coeficientes em codificador EZW

Na Figura 6, temos a exemplificação de dependências entre pai-filho. Cada coeficiente pai tem quatro filhos, com exceção da sub-banda de baixas frequências, onde cada pai tem três filhos, um em cada sub-banda na mesma escala mas de diferente orientação.

Os coeficientes são visitados em uma ordem pré estabelecida de forma que nenhum coeficiente filho seja visitado antes do seu pai, mas somente depois que seu pai e todos os pais vizinhos foram visitados (FIGURA 7). Para uma transformada de 3 escalas, a varredura começa na sub-banda de frequência mais baixa, denotada por LL_3 , e segue para as sub-bandas HL_3, LH_3, HH_3 , como mostrado na Figura J, deste ponto ela segue para a escala dois, e assim por diante.

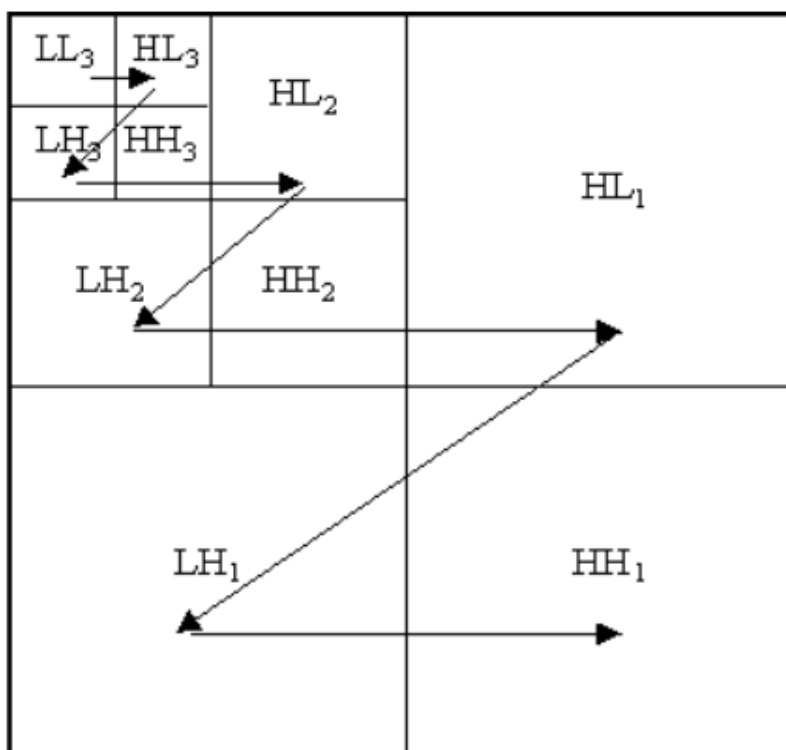


FIGURA 7 – Baseado em (SHAPIRO, 1993) – Ordem de varredura dos coeficientes wavelets nas sub-bandas

Se obtivermos um coeficiente wavelet x na escala mais grossa, e esse coeficiente seja insignificante com respeito a um

limiar T , isto é, se $|x| < T$, então todos os coeficientes wavelets descendentes deste nó serão, possivelmente, insignificantes com respeito a T (GUSMAO, 2002). Assim, todos os coeficientes de mesma localização espacial, como exemplificado na Figura 10, serão codificados com um único símbolo denominado *Zerotree* (árvore de zeros) (SHAPIRO, 1993). *Zerotree* é uma árvore na qual todos os seus nós representam coeficientes iguais ou menores que o limiar.

O codificador EZW explora a *zerotree* com base na observação que os coeficientes tendem a decrescer com a escala.

Considera-se que haverá maior probabilidade de que um coeficiente na árvore seja menor que um certo limiar se sua raiz for menor que este limiar. Se este for o caso, então toda a árvore pode ser codificada com um único símbolo *zerotree*, alcançando assim alta compressão. Se a imagem é visitada em uma ordem predefinida, indo da escala mais alta para a mais baixa (Figura 11), implicitamente muitas posições serão codificadas através do uso de *zerotree*. Na prática, em muitos casos não se obtêm *zerotree*, mas a probabilidade de que ela ocorra é geralmente alta (GUSMAO, 2002).

Na codificação é realizada uma quantização, que está relacionada com a codificação das magnitudes dos coeficientes, chamada quantização de aproximação sucessiva (QAS). O primeiro passo é determinar o limiar inicial e, aos poucos, a seqüência de limiares (T_0, \dots, T_{N-1}) para que a significância dos coeficientes seja determinada. A seqüência de limiares usada é de potências de dois, chamada codificação *bitplane* (GUSMAO, 2002) citando [36]. Sendo assim, o limiar inicial T_0 é escolhido tal que (3.1)

$$T_0 = 2^{\lceil \log_2 x_{\max} \rceil}, \quad (3.1)$$

onde, x_{\max} corresponde ao coeficiente de valor máximo na imagem, e $\lceil \log_2 x_{\max} \rceil$ significa o menor inteiro maior que o argumento. Os limiares das interações seguintes são reduzido consecutivamente por um fator de 2, ou seja (3.2),

$$T_i = T_{i-1}/2 \quad (3.2)$$

Durante a codificação (e decodificação) duas listas separadas de coeficientes wavelets são formadas, a lista dominante e a lista subordinada. A lista dominante contém as coordenadas daqueles coeficientes que ainda não foram considerados significantes com respeito ao limiar, ou seja, que ainda não foram codificados. Esta lista é formada na mesma ordem em que os coeficientes estão sendo visitados. A lista subordinada contém as magnitudes dos coeficientes que foram classificados como significantes, ou seja, aqueles valores encontrados superiores ao limiar em módulo. Para cada limiar, cada lista é incrementada uma vez durante o processo de codificação.

Na codificação (e decodificação) há também duas passagens: uma passagem dominante e uma passagem subordinada. A passagem dominante encontra os valores dos pixels que estão acima do limiar, enquanto a passagem subordinada quantiza todos os valores dos pixels significantes encontrados nas passagens dominantes anteriores.

Durante uma passagem dominante, os coeficientes com coordenadas na lista dominante, que ainda não foram tidos como significantes, são comparados com o limiar “ T_i ” para determinar suas significâncias. Se significantes, os sinais (positivo ou negativo) destes coeficientes são codificados. Valores de pixels tidos como significantes na passagem dominante são codificados com o símbolo positivo para um valor maior que zero, ou um símbolo negativo, para um valor menor que zero (SHAPIRO, 1993). Estes pixels são então adicionados à lista subordinada para serem quantizados, e as posições deles na imagem são preenchidas com zeros, evitando assim, que sejam codificados novamente na próxima passagem dominante. Valores de pixels encontrados sendo insignificantes na passagem dominante, mas com filhos significantes, são codificados como Zeros Isolados, e seus descendentes são codificados individualmente.

Quando o valor de um pixel é visto como insignificante, e todos os seus descendentes também o são, é possível codificar aquele pixel e toda sua geração com um único símbolo *zerotree*, como visto anteriormente. Assim, a passagem dominante mapeia os valores dos pixels em quatro símbolos do alfabeto os quais podem ser codificados com o uso de um codificador aritmético adaptativo (GUSMAO, 2002) citando (WITTEN et, al. 1987). Os quatro símbolos usados são:

1. POS (Positivo Significante): o coeficiente é maior ou igual ao limiar atual, com sinal positivo, e não foi codificado nas iterações anteriores;
2. NEG (Negativo Significante): o coeficiente é maior ou igual ao limiar atual, com sinal negativo, e não foi codificado anteriormente;
3. IZ (Zero Isolado): o coeficiente é menor que o limiar atual (i. e., é insignificante), mas um ou mais dos seus descendentes não o são;
4. ZTR (Raiz *Zerotree*): o coeficiente (que no caso é a raiz da árvore) e todos os seus descendentes são insignificantes com respeito ao limiar, ou somente ele é insignificante, se este estiver localizado em uma das sub-bandas de mais alta frequência (LH_1, HL_1 ou HH_1 , FIGURA 7).

Cada passagem dominante é seguida por uma passagem subordinada, que refina a magnitude de todos os coeficientes que se encontram na lista subordinada, alcançando assim uma maior precisão. Durante uma passagem subordinada, a largura do degrau do quantizador, que define o intervalo de incerteza para a magnitude real do coeficiente, é dividida pela metade. Para cada magnitude encontrada na lista subordinada, este refinamento pode ser codificado usando um alfabeto binário com o símbolo “1” indicando que o valor verdadeiro do coeficiente está na metade

superior do intervalo de incerteza ou um símbolo “0” indicando que o valor verdadeiro deste coeficiente se encontra na metade inferior(GUSMAO, 2002).

O processo continua alternando entre uma passagem dominante e uma passagem subordinada, enquanto o limiar vai sendo reduzido pela metade. A codificação pára quando uma determinada condição é satisfeita, tal como quando a taxa de bits planejada é alcançada.

3.2 A Wavelet de Haar

A wavelet de Haar é considerada a mais simples das wavelets. Ela foi proposta em 1909 pelo matemático Alfred Haar. A transformada de Haar é uma caso particular de transformada wavelet discreta, onde sua representação é definida por um pulso quadrado (3.3) (LIMA, 2003):

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 0.5 \\ -1, & 0.5 \leq t < 1. \\ 0, & \text{para outros valores de } t \end{cases} \quad (3.3)$$

Essa wavelet foi proposta muito antes do termo wavelet ser proposto. A transformada de Haar pode ser usada para representar as funções $f(t)$ como sendo o somatório (3.4)(LIMA, 2003):

$$fI(t) = \sum_{k=-\infty}^{\infty} C_k \phi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \psi(2^j t - k), \quad (3.4)$$

onde $\phi(t)$ é a função de escala definida por (3.5):

A imagem não pode ser exibida. Talvez o computador não tenha recebido software para abrir a imagem ou talvez ela esteja corrompida. Reinicie o computador e abra o arquivo novamente. Se ainda assim aparecer o e a mensagem, poderá ser necessário excluir a imagem e inserir novamente.

$$\quad (3.5)$$

e c_k e $d_{j,k}$ são parâmetros a serem calculados (LIMA, 2003).

Uma representação da wavelet de Haar encontra-se na figura 8:

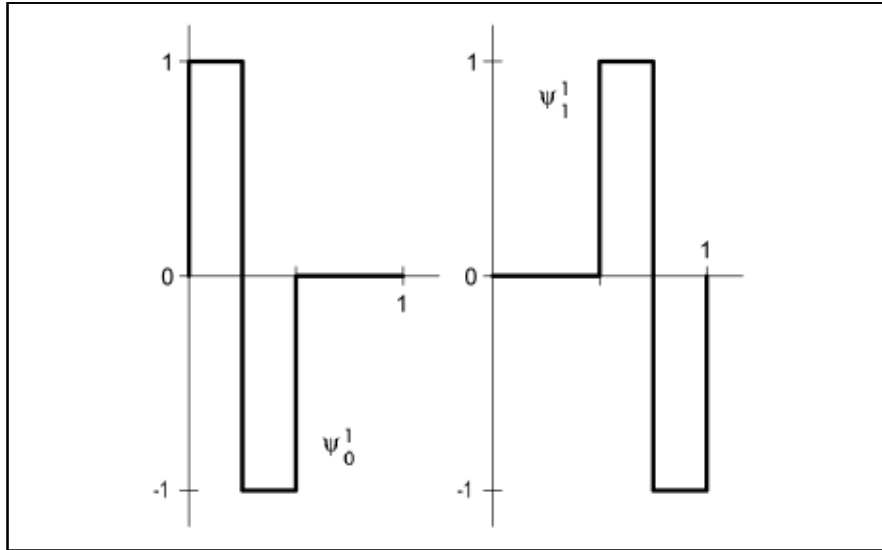


FIGURA 8 – Exemplos de wavelet de Haar.

4 Análise de Algoritmo e Imagens

Primeiramente, para o desenvolvimento do software de compressão de imagens, foram necessários códigos existentes que possuíssem a implementação do EZW, entretanto, o desenvolvimento fora feito em Java e os Códigos implementaram o EZW em Pascal e C. Também foram feitas diversas análises das diversas dissertações referentes ao tema desse trabalho, para que o mesmo prosseguisse, sendo necessário o entendimento sobre o tema.

4.1 Análise do código desenvolvido

Para o desenvolvimento do código para compressão de imagens baseado em EZW, os seguintes códigos foram tomados como base:

- Do autor Mow-Song, Ng em 20-07-2002 (MOW, 2002);
- Do autor C. Valente em 07-09-1999 (VALENTE, 1999), existe duas implementações uma em Pascal e a outra em C.

A base do algoritmos para compressão de imagem baseados em EZW consiste em:

1. Transformar uma imagem (quadrada) em um fluxo de dados para um arquivo de bit;
2. Com os dados em bits, aplicar a transformada wavelet discreta;
3. Aplicar o algoritmo EZW nos dados transformados;
4. Reconstruir a imagem através de um fluxo de dados, para formar a imagem comprimida.

Necessariamente neste trabalho existe a necessidade da imagem ser quadrada, pois, o algoritmo EZW não trabalha imagens em dimensões não quadradas, causando um erro do tipo estouro de array (*ArrayIndexOutOfBoundsException*) que é um exceção comum, essa exceção refere-se a quando tentamos colocar mais

dados do que nossa aplicação suporta. No caso de uma imagem não quadrada, ele criará uma matriz quadrada com base na primeira linha da imagem, caso a imagem possua 10 colunas na primeira linha, ele criará uma matriz 10x10, mas se a imagem possuir 20 linhas a matriz não suportará, pois ela está aguardando apenas 10 linhas.

Temos também a restrição de imagens coloridas, pois como o algoritmo explora os tons de cinza, qualquer outro tipo de valor que não esteja na escala de tons de cinza provoca grandes distorções na imagem, deixando-a ilegível, ou seja, não identificável.

A seguir será explicado o Erro Médio Quadrático (EMQ) que será nossa medida de variância entre a imagem Original e a imagem Comprimida.

4.2 Erro Médio Quadrático (EMQ)

O Erro Médio Quadrático é uma estimática de σ^2 que mede a variância entre as imagens, no nosso caso, visando mostrar o quão diferente a imagem Original, referindo-se a imagem base, ficou da imagem Comprimida. Quanto mais o resultado dessa variância se aproxima de zero, menor a diferença entre os pixels dessas imagens.

O calculo simples para o EMQ é ilustrado na fórmula abaixo (4.1):

$$EMQ = \sum_{i=j=1}^n ((s_i - s_j)^2 / N) \quad (4.1)$$

Onde s_i é o pixel da imagem original, s_j é o pixel da imagem comprimida, N é o número total de pixels, n é o número de amostras no grupo.

4.2 Análise de Imagens Comprimidas

Nesta sessão apresentam-se os resultados objetivos e subjetivos da utilização do algoritmo de compressão de imagens.

As imagens de teste utilizadas foram Lena e Bárbara, essas imagens são famosas e utilizadas em diversos trabalhos que envolvem imagens. Ambas com dimensões $V = 512$ e $H=512$, onde o V é o número de pixel por coluna e H é o número de pixels por linha. Esta duas imagens encontram-se na figura 9.



FIGURA 9 – Imagens de Teste

Para simulações utilizou-se o filtro wavelet haar, com 15 níveis de decomposição, pois, por ser a wavelet de haar a imagem fica quadriculada, dessa forma, os 15 níveis permitem uma melhoria na imagem, ficamos assim, atentos apenas à quantidade de bits nas compressões. O coeficiente da wavelet de haar são mostrados logo abaixo (Tabela 1). somente é mostrado o coeficiente da direta, visto que o filtro é simétrico (Livro).

Wavelet Haar
0.707106781186547524400844362104849

Tabela 1 – Coeficiente do filtro Haar

Este filtro foi escolhido por ser conhecido como o mais simples, e o primeiro a ser desenvolvido.

As figuras a seguir mostram os resultados das simulações obtidos segundo critérios objetivos. Para cada uma destas

condições apresenta-se o Erro Médio Quadrático (EMQ), que será nossa medida de variância. O EMQ foi explicado na pequena sessão anterior.

Temos primeiramente, a análise da imagem de Lena, com a compressão tendo o mesmo número de bytes da imagem original (FIGURA 14). Nesse caso, para verificarmos se o EMQ calculado é verdadeiro codificamos a imagem (a) resultando na imagem (b), obtemos o $EMQ = 1$, significando que a imagem (a) é muito semelhante à imagem (b).



(a) Lena Original

(b) Lena Comprimida

FIGURA 10 – Imagem Lena codificada com 152 Kb, e $EMQ = 1$.

Nossa próxima imagem possui metade do tamanho da imagem Original, resultando em um $EMQ = 9$, aparentemente houve diferenças de nitidez entre as imagens, mas analisando o primeiro pixel, vemos uma pequena alteração, que se reflete por toda a imagem. No caso da imagem Original o valor do primeiro pixel é 162 e da imagem comprimida 161, diferença que subjetivamente, ou seja, visualmente é imperceptível (FIGURA 11).



FIGURA 11 – Imagem Lena codificada com 76 Kb, EMQ = 9

Agora com uma compressão de 75% observamos uma EMQ = 30, onde percebemos uma quadriculação da imagem, que é uma característica da wavelet de haar (FIGURA 12).



FIGURA 12 – Imagem Lena codificada com 38 Kb, EMQ = 30

Iremos comprimir agora a imagem de Bárbara (FIGURA 13), porém já iniciando a compressão com metade de Kilobytes (21,5 Kb), no caso 50%. Podemos observar que o EMQ = 175, significando que as imagens possuem uma grande variância entre elas. Do ponto de vista subjetivo, vemos que a imagem comprimida possui uma grande perda da qualidade, sendo clara suas diferenças.

**(a) Bárbara Original****(b) Bárbara Comprimida (50%)**

FIGURA 13 – Imagem Bárbara codificada com 21,5 Kb, EMQ = 175

Com uma compressão em 75%, em relação à imagem Original, temos um EMQ = 283 (FIGURA 14). Observando a compressão percebemos grandes divergências entre as imagens, o que já seria esperado pois, quanto maior o EMQ maior será as divergências entre as imagens.

**(a) Bárbara Original****(b) Bárbara Comprimida (75%)**

FIGURA 14 – Imagem Bárbara codificada com 10,75 Kb, EMQ = 283

5 Conclusão

Neste trabalho, visamos à compressão de imagens utilizando wavelets, para isso, utilizamos a base de compressão de imagens wavelets a EZW, desenvolvida por (SHAPIRO,.) e exemplificada em diversas dissertações e em raras implementações. Também, utilizamos algoritmos do EZW implementados em C e Pascal, servindo de guia para o desenvolvimento, já que não se trata de um algoritmos trivial.

O codificador EZW possui algumas particularidades, no nosso caso, a imagem a ser codificada precisa ser quadrada, pois o algoritmo EZW é fundamentado para trabalhar com imagens de lados iguais; e com escala de cinza, pois no nosso caso, imagens coloridas podem representar uma variedade grande de frequências e isso aumentaria a dificuldade e não-trivialidade do trabalho.

Na sessão de análise das imagens comprimidas, pudemos observar que quanto maior a quantidade de Kilobytes uma imagem possuir, melhor é sua compressão, acreditamos que seja pela quantidade de detalhes ser maior, assim o algoritmo pode captar de descartar pixels que não fazem tanta diferença na representação daquela imagem. A imagem Original de Lena possui 152 Kb, enquanto que a imagem de Bárbara possui 43 Kb. Quando comprimimos ambas imagens à metade de Kb, ou seja, 50% da imagem Original, vemos que Lena possui uma qualidade grande se comparada à imagem Original resultando em um EMQ = 9, enquanto que Bárbara resulta em um EMQ = 175, com uma qualidade bem inferior se comparada à imagem Original.

Com isso, podemos concluir que a compressão de imagens utilizando técnicas wavelet mostra-se muito promissora, diversos outros algoritmos de compressão foram desenvolvido tendo como base o EZW. Com isso se evidencia que wavelets para compressão de imagens pode ser uma forma de otimizar a compressão de imagens, aumentando sua robustez e confiabilidade.

REFERÊNCIAS

AMBRANSON, N. *Information Theory and Coding*. McGraw-Hill Book Company, 1963.

FARIAS, M. C. Q. **Aplicação da Transformada Wavelet na Compressão de Imagens**. 1998. 162 p. Dissertação (Mestrado) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1998.

GUSMÃO, A. A. **Método Robusto e Simples de Compressão de Imagens Baseado no Algoritmo EZW**. 2002. 89 p. Dissertação (Mestrado) – Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 2002.

LIMA, P. C. Wavelets: uma introdução. Departamento de Matemática – ICEX – UFMG, 2003.

MOW, S. Home Page, que possui diversos algoritmos e códigos de programas. Disponível em: <http://read.pudn.com/downloads38/sourcecode/graph/130053/ewz/EZW2/EZW.C_.htm>. Acesso em: 01 jun 2010.

OLIVEIRA, K. F. **ANÁLISE DA TRANSFORMADA WAVELET DIRECIONAL ADAPTATIVA NA CODIFICAÇÃO DE IMAGENS**. 2008. 137 p. Dissertação (Mestrado) – Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, 2008.

SHAPIRO, J. M. “**Embedded Image Coding Using Zerotrees of Wavelet Coefficients**”. *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.

WITTEN, J. H; NEAL, R. M; CLEARY, J. G. “*Arithmetic Coding for Data Compression*”, *Comm. of ACM*, vol.30, no 6, pag. 520-540, 1987.

VALENTE, C. Home Page francesa, que possui diversas informações sobre wavelets. Disponível em: <<http://pagesperso-orange.fr/polyvalens/clemens/ezw/ezw.html>>. Acesso em: 01 jun 2010.