Revista Eletrônica Paulista de Matemática

ISSN 2316-9664
Volume 14, fev. 2019
Edição Ermac

Natalia Caroline Lopes Travassos
UNESP- Universidade Estadual Paulista "Júlio de Mesquita Filho".
nataliacaroline2006@gmail.com

Marco Aparecido Queiroz Duarte
UEMS - Universidade Estadual de Mato Grosso do Sul, Cassilândia.
marco@uems.br

Francisco Villarreal Alvarado
UNESP- Universidade Estadual Paulista "Júlio de Mesquita Filho".
villa@mat.feis.unesp.br

Estudo de revisão e proposta de aprimoramento para o armazenamento de imagens comprimidas via transformada wavelet

Review study and improvement proposal for compressed images storage via wavelet transform

Resumo

Este trabalho tem por objetivo fazer um estudo de revisão de quatro artigos, dos quais, três abordam métodos de armazenamento e reconstrução de imagens e o quarto vai além, apresentando alguns progressos em relação aos três primeiros. Os trabalhos estudados podem ser aplicados apenas em imagens de dimensões $m \times n$, sendo $m = n = 2^k$, e utilizam a função wavelet de Daubechies de ordem 6 (db6), ao passo que este artigo estende a aplicação para imagens onde m e n são quaisquer, além de substituir a função db6 pela função biortogonal 4.4 (bior4.4) que permite obter uma imagem com menor degradação visual em relação ao processamento feito com a db6 e que retorna menos coeficientes na decomposição wavelet, fazendo com que o vetor de armazenamento ocupe menos espaço. Outro diferencial deste trabalho em relação aos artigos revisados está no uso de mais duas métricas de avaliação da qualidade das imagens reconstruídas, sendo elas o erro quadrático médio e a similaridade estrutural.

Palavras-chave: Compressão. Armazenamento. Reconstrução. Transformada wavelet.

Abstract

The objective of this work is to do a review of four articles, of which three deal with methods of storing and reconstructing images, and the fourth goes further, presenting some progress in relation to previous ones. The studied works can be applied only to images of $m \times n$ dimensions, where $m = n = 2^k$, and use the db6, whereas this article extends the application to images where m and n are any, in addition to replace the db6 function by bior4.4 function that allows to obtain an image with less visual degradation in relation to the processing done with db6 and that returns less coefficients in the wavelet decomposition, making the storage vector to occupy less space. Another difference of this work in relation to the reviewed articles is the use of two more metrics to evaluate the quality of the reconstructed images, which were the root mean square error and the structural similarity.

Keywords: Compression. Storage. Reconstruction. Wavelet transform.

Artigo recebido em set. 2018 e aceito em fev. 2019.

1 Introdução

A grande quantidade de informações trocadas nos dias atuais faz com que a compressão de sinais seja algo altamente desejado em diversos ambientes. Neste trabalho será tratado um sinal em particular, a saber, uma imagem. Uma ferramenta matemática que se destaca no processamento de imagens é a transformada wavelet (DAUBECHIES, 1992). Devido às propriedades da análise de multirresolução, a transformada wavelet possibilita que uma imagem seja decomposta em vários níveis de resolução de forma que suas características sejam totalmente explicitadas (DAUBECHIES, 1992; MALLAT, 1989; STOLLNITZ; DEROSE; SALESIN, 1995). Isto possibilita uma análise detalhada da mesma e uma compressão muito eficaz (COIFMAN, 1990).

Um aspecto de grande interesse é o armazenamento de uma imagem na memória do computador. Um exemplo que apresenta essa necessidade é o problema com imagens médicas, já que elas são grandes em tamanho, e assim, laboratórios e hospitais necessitam de uma grande quantidade de largura de banda para enviar e receber imagens. “Considerando um hospital que faça ressonância magnética, tomografia computadorizada, radiografia computorizada, etc., pode-se estimar que serão produzidos de 5 a 15 GB de dados por dia” (BAIRAGI, 2015, p.718). Neste caso, e em tantos outros, é necessário comprimir estes dados utilizando uma técnica de compressão, isso porque uma imagem, depois de discretizada e codificada, carrega consigo uma quantidade muito grande de informação, e eventualmente ocupará muito espaço para o seu armazenamento.

Existem dois tipos de compressão de imagem: com perdas e sem perdas. Na compressão sem perdas, a imagem original é recuperada exatamente após a descompressão, ao contrário da compressão com perdas, que será abordada neste trabalho (WALKER; NGUYEN, 2001). A codificação baseada em wavelets fornece melhorias substanciais na qualidade da imagem para altas taxas de compressão. Ao longo dos últimos anos, uma variedade de métodos poderosos e sofisticados de compressão com perdas, como o esquema de codificação EZW (embedded zerotree wavelet), SPIHT (set partitioning in hierarchical trees), WDR (wavelet difference reduction) e ASWDR (adaptively scanned wavelet difference reduction) foram desenvolvidos (PAN; SIU; LAW, 2008). Os métodos típicos de compressão sem perdas são: algoritmo Huffman, LZW, compressão aritmética ou compressão run-length (WU; OTOO; SHOSHANI, 2006).

Este estudo faz uma revisão dos trabalhos de Silva (2008), Silva, Duarte e Villarreal (2011), Travassos, Duarte e Villarreal (2016) e Travassos, Duarte e Villarreal (2018), além de propor uma adaptação a este último trabalho, a fim de tornar possível a compressão, codificação e reconstrução de imagens coloridas de dimensões $m \times n$, não sendo mais necessário ter $m = n = 2^k$. Outro avanço encontra-se na substituição da wavelet de Daubechies de ordem 6 (db6), pela biortogonal 4.4 (bior4.4). Para atestar a qualidade das imagens processadas pela bior4.4, além da PSNR, outras duas métricas foram utilizadas: MSE e $SSIM$.

O restante do texto é organizado da seguinte forma: Na Seção 2 são apresentados os algoritmos de Silva (2008), Silva, Duarte e Villarreal (2011), Travassos, Duarte e Villarreal (2016) e Travassos, Duarte e Villarreal (2018). Na Seção 3 são apresentadas as três contribuições propostas; e os testes encontram-se na Seção 4, onde é feita a comparação dos resultados usando a wavelet bior4.4 e db6; por fim, na Seção 5 são dadas as considerações finais.

2 Armazenamento de imagens comprimidas: uma revisão

Como exemplo, considere uma matriz A, que representa uma imagem de dimensões 4×4, $\hat{A}$, sua versão decomposta pela transformada wavelet discreta (TWD) e A_L a matriz após aplicação do limiar rígido (DONOHO; JOHNSTONE, 1994):

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \overrightarrow{TWD}\ \hat{A} = \begin{bmatrix} b_{11} & c_{12} & b_{13} & c_{14} \\ b_{21} & c_{22} & b_{23} & c_{24} \\ b_{31} & c_{32} & c_{33} & c_{34} \\ b_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \overrightarrow{l\imath m\imath ar}\ A_L = \begin{bmatrix} b_{11} & 0 & b_{13} & 0 \\ b_{21} & 0 & b_{23} & 0 \\ b_{31} & 0 & 0 & 0 \\ b_{41} & 0 & 0 & 0 \end{bmatrix}$$

Em $\hat{A}$, os coeficientes representados por b_{ij} são coeficientes significativos, enquanto que os que estão representados por c_{ij} são redundantes, ou seja, podem ser eliminados, pois estão abaixo de um limiar a ser definido. A matriz A_L é uma matriz esparsa, com poucos coeficientes representativos. Por isso, só há a necessidade de armazenar tais coeficientes.

2.1 Método de Silva (2008)

Silva (2008) propôs a criação de um vetor, denominado V_cod, capaz de representar a matriz comprimida. Logo, para a matriz A_L, V_cod é dado na equação (1):

$$V\_cod = [4 \quad b_{11} \quad 1 \quad 1 \quad b_{13} \quad 1 \quad 3 \quad b_{21} \quad 2 \quad 1 \quad b_{23} \quad 2 \quad 3 \quad b_{31} \quad 3 \quad 1 \quad b_{41} \quad 4 \quad 1], \qquad (1)$$

sendo que o número 4 na primeira posição indica a dimensão da matriz quadrada. Após o valor do primeiro elemento, b_{11}, os números 1 e 1 indicam que este coeficiente se encontra na linha 1 e na coluna 1 e, assim por diante. Assim, para uma matriz transformada e comprimida de ordem m, o vetor V_cod é escrito, de modo geral, como na equação (2):

$$V\_cod = \left[m \quad b_{i_1 j_1} \quad i_1 \quad j_1 \quad b_{i_2 j_2} \quad i_2 \quad j_2 \quad \dots \quad b_{i_R j_R} \quad i_R \quad j_R\right], \qquad (2)$$

onde R representa a quantidade de elementos não-nulos da matriz A_L, i_r e j_r a linha e a coluna, respectivamente, em que se encontra o elemento$b_{i_r j_r}$, com $r = 1, \dots, R$.

2.2 Método de Silva, Duarte e Villarreal (2011)

A fim de reduzir o tamanho do vetor V_cod, Silva, Duarte e Villarreal (2011) propuseram um novo vetor denominado V_cod_mat, que ao invés de armazenar os valores de i_r e j_r, armazenaria um único valor $a^3_{i_r j_r}$. Sendo assim, tal proposta causa uma redução de praticamente um terço do espaço computacional necessário para o armazenamento de uma imagem comprimida quando comparamos os vetores V_cod e V_cod_mat, respectivamente. Em seu artigo, Silva, Duarte e Villarreal (2011) detalham como é possível obter $a^3_{i_r j_r}$ a partir de i_r e j_r depois que uma sequência de operações matemáticas. Em 2016, Travassos, Duarte e Villarreal (2016) mostraram que todas as operações feitas para obter $a^3_{i_r j_r}$ poderiam ser substituídas por uma função invertível. A regra de correspondência dessa função é dada em (3) e sua inversa, usada na decodificação, é a função dada pela equação (4).

$$N: D_m = \{1,2,\dots,m\} \times \{1,2,\dots,m\} \rightarrow \quad I_m = \{3,4,\dots,m^2+2\}$$
$$(i,j) \mapsto \quad N(i,j) = a^3_{i_r j_r} = -(m-2) + mi + j. \tag{3}$$

$$M(a^3_{ij}) = \left(\frac{a^3_{ij} - 3 - mod(a^3_{ij} - 3, m)}{m} + 1, mod\left(a^3_{ij} - 3, m\right) + 1\right), \tag{4}$$

e para $a, b \in \mathbb{R}$, $mod(a, b)$ retorna o resto da divisão de a por b.

Para a matriz A_L do início dessa seção, V_cod_mat é dado na equação (5), e para uma matriz transformada e comprimida de ordem m, o vetor V_cod_mat é escrito, de modo geral, como na equação (6):

$$V\_cod\_mat = [4 \quad b_{11} \quad 3 \quad b_{13} \quad 5 \quad b_{21} \quad 7 \quad b_{23} \quad 9 \quad b_{31} \quad 11 \quad b_{41} \quad 15]. \tag{5}$$

$$V\_cod\_mat = \left[m \quad b_{i_1 j_1} \quad a^3_{i_1 j_1} \quad b_{i_2 j_2} \quad a^3_{i_2 j_2} \quad \dots \quad b_{i_R j_R} \quad a^3_{i_R j_R}\right]. \tag{6}$$

2.3 Método de Travassos, Duarte e Villarreal (2016)

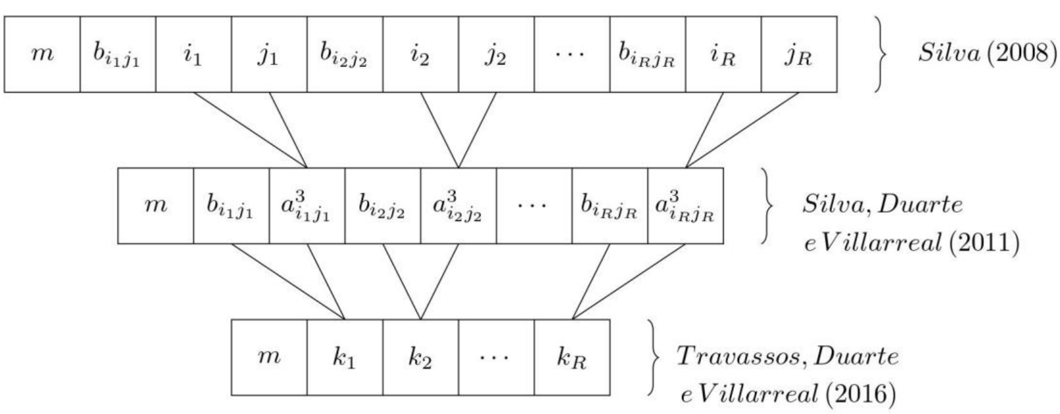
Travassos, Duarte e Villarreal (2016) propuseram que o vetor V_cod_mat fosse reduzido, criando assim o vetor V_cod_prop. Para isso, bastava criar uma função f capaz de relacionar b_{ij}, i e j de tal modo que $f(b_{ij}, i, j) = f(b_{ij}, a^3_{ij})$. Para cada pixel e suas coordenadas, $f(b_{ij}, a^3_{ij}) = k_r$ é armazenado em um vetor que é usado na reconstrução da imagem sem que sua qualidade seja comprometida. Sendo assim, o vetor V_cod_prop é descrito em (7):

$$V\_cod\_prop = [m \quad k_1 \quad k_2 \quad k_3 \quad \dots \quad k_R]. \tag{7}$$

Nesse trabalho os autores usaram $f(b_{ij}, a^3_{ij}) = sign(b_{ij})(\log_{m^5}|b_{ij}| + a^3_{ij}) = k_r$. Para determinar a^3_{ij} e b_{ij} através de k_r basta substituí-lo nas equações $a^3_{ij} = \lfloor |k_r| \rfloor$ e $b_{ij} = sign(k_r)(m^5)^{||k_r| - \lfloor |k_r| \rfloor|}$, respectivamente. $sign$ representa a função sinal, que retorna 1, -1 ou 0, se o argumento for positivo, negativo ou nulo, respectivamente.

De modo geral pode-se resumir a evolução entre os três vetores de armazenamento propostos, de acordo com a Figura 1:

Figura 1 - Redução do vetor de armazenamento de imagens comprimidas



Fonte: Elaboração da autora

Portanto, o método de Travassos, Duarte e Villarreal (2016) é em média 66,66% mais econômico que o de Silva (2008), e é em média 49,99% mais econômico que o de Silva, Duarte e Villarreal (2011).

2.4 Contribuições de Travassos, Duarte e Villarreal (2018)

O trabalho de Travassos, Duarte e Villarreal (2018) teve por objetivo mostrar a evolução obtida com os três métodos citados nas seções 2.1, 2.2 e 2.3, além de propor três importantes contribuições a este último trabalho, que serão detalhadas nas subseções a seguir.

2.4.1 Criando funções invertíveis

O trabalho de Travassos, Duarte e Villarreal (2016) apresentou uma única função f tal que $f\left(b_{ij}, a_{ij}^3\right) = k_r$. Travassos, Duarte e Villarreal (2018) propuseram uma maneira prática de construir funções f que satisfizessem tal igualdade e que fossem invertíveis, o que possibilitou a reconstrução da imagem comprimida através do vetor V_cod_prop.

Para cada $m \in \{2^n : n \in \mathbb{N}\}$ seja $I_m = \{3,4,\dots,m^2+2\}$. Considere o problema de determinar uma função invertível $f : \mathbb{R} \times I_m \to \mathbb{R}$ que a cada par $(b_{ij}, a_{ij}^3) \in \mathbb{R} \times I_m$ faz corresponder um único número $f\left(b_{ij}, a_{ij}^3\right)$. Como a variável a_{ij}^3 é sempre um número natural, se considerarmos uma função $g : \mathbb{R} \to [0,1)$ e definirmos

$$f\left(b_{ij}, a_{ij}^3\right) = g\left(b_{ij}\right) + a_{ij}^3, \tag{8}$$

então pode-se facilmente determinar a_{ij}^3 a partir de $f\left(b_{ij}, a_{ij}^3\right)$ aplicando a função piso ($\lfloor x \rfloor$), que arredonda x na direção de menos infinito. Logo,

$$\left\lfloor f\left(b_{ij}, a_{ij}^3\right) \right\rfloor = a_{ij}^3. \tag{9}$$

Para se determinar a variável b_{ij} a partir de $f\left(b_{ij}, a_{ij}^3\right)$, observe que (8) e (9) implicam em (10).

$$f\left(b_{ij}, a_{ij}^3\right) - \left\lfloor f\left(b_{ij}, a_{ij}^3\right) \right\rfloor = g\left(b_{ij}\right). \tag{10}$$

Supondo que a função g seja invertível, então de (10) é possível determinar a variável b_{ij} a partir de $f\left(b_{ij}, a_{ij}^3\right)$, segundo a equação (11).

$$g^{-1}\left(f\left(b_{ij}, a_{ij}^3\right) - \left\lfloor f\left(b_{ij}, a_{ij}^3\right) \right\rfloor\right) = g^{-1}\left(g\left(b_{ij}\right)\right) = b_{ij}. \tag{11}$$

Logo, se $g : \mathbb{R} \to [0,1)$ é uma função invertível, a função f definida pela equação (8) será invertível, e sua inversa é dada pela equação (12).

$$f^{-1}(k) = \left(g^{-1}(k - \lfloor k \rfloor), \lfloor k \rfloor\right). \tag{12}$$

Portanto, o problema de determinar uma função invertível $f : \mathbb{R} \times I_m \to \mathbb{R}$ pode ser resumido a encontrar uma função $g : \mathbb{R} \to [0,1)$ que seja invertível e definir f pela equação (8).

Como exemplo, Travassos, Duarte e Villarreal (2018) apresentaram a função (13) que satisfaz tais condições. Sua inversa é a função dada na equação (14).

$$f\left(b_{ij}, a_{ij}^3\right) = \frac{\tan^{-1}(b_{ij})}{\pi} + \frac{1}{2} + a_{ij}^3 = k_r. \tag{13}$$

$$f^{-1}(k_r) = \left(\tan\left(\pi\left(k_r - \lfloor k_r \rfloor - \frac{1}{2}\right)\right), \lfloor k_r \rfloor\right). \tag{14}$$

Este método para a criação de funções invertíveis possibilita a reconstrução exata da imagem comprimida, fazendo com que não haja o erro de aproximação obtido em Travassos, Duarte e Villarreal (2016). Ainda vale a pena dizer que o uso de uma função no armazenamento de imagens comprimidas torna o processo mais rápido.

2.4.2 Taxa de compressão

Os trabalhos de Silva (2008) e Silva, Duarte e Villarreal (2011) não permitem que o usuário escolha uma taxa de compressão. A cada imagem fornecida, o próprio código define um limiar para cada linha da matriz e esse limiar determina quantos coeficientes serão eliminados em cada linha; após eliminar os coeficientes da última linha é obtida a taxa de compressão. Com o intuito de aprimorar tal método, Travassos, Duarte e Villarreal (2016) apresentaram uma versão modificada do algoritmo, onde o usuário pode dizer ao algoritmo a taxa de compressão mínima desejada. Apesar de representar uma evolução, pode acontecer de a taxa de compressão obtida ser muito superior à taxa mínima. Para contornar este problema, Travassos, Duarte Villarreal (2018) propuseram outra alteração no código, que permite obter exatamente a mesma taxa de compressão escolhida pelo usuário.

Desse modo, um vetor recebe os coeficientes da matriz transformada, os ordena e, de acordo com a taxa de compressão, determina quantos elementos devem ser eliminados para que a mesma seja atingida. Dentre os menores elementos a serem eliminados, o maior é escolhido para ser o limiar.

Embora pareça simples, a proposta de determinação de uma taxa de compressão a priori proporcionou grande economia no tempo de processamento dos métodos de Silva (2008) e Silva, Duarte e Villarreal (2011). Além disso, foi possível contornar o problema de Travassos, Duarte e Villarreal (2016), onde havia a possibilidade de encontrar uma taxa de compressão superior à desejada.

2.4.3 Processamento de imagens coloridas

Os trabalhos de Silva (2008), Silva, Duarte e Villarreal (2011) e Travassos, Duarte e Villarreal (2016) abordaram apenas a compressão, armazenamento e reconstrução de imagens em escala de cinza, pois todo o processamento é feito em apenas um canal da imagem, o canal R (vermelho).

Usando o método de armazenamento de Travassos, Duarte e Villarreal (2016), Travassos, Duarte e Villarreal (2018) conseguiram processar imagens coloridas, armazená-las e reconstruí-las.

3 Contribuições propostas

Travassos, Duarte e Villarreal (2018) propuseram um método para comprimir, armazenar e reconstruir imagens coloridas cujas dimensões fossem $m \times n$, sendo $m = n = 2^k$, $k \in \mathbb{N}$. Neste estudo, propõe-se estender tal algoritmo para abranger os casos de imagens onde m e n são quaisquer $(m, n > 0)$, e não necessariamente $m = n = 2^k$, $k \in \mathbb{N}$.

O processo de armazenamento da imagem codificada é semelhante ao dos trabalhos estudados, porém com algumas modificações: Dada uma imagem, o algoritmo verifica qual o comprimento da maior dimensão, para definir o nível máximo de decomposição que poderá ser aplicado. Na sequência, é feita a decomposição wavelet em todos os níveis e os coeficientes são armazenados em um vetor.

Seja C o vetor que contém os coeficientes da decomposição wavelet, sendo $C(j)$ cada elemento e m o seu comprimento. Após definirmos uma taxa de compressão é feita a limiarização, e os elementos de C que estão abaixo do limiar tornam-se zeros.

Usando a função apresentada na equação (13), e fazendo as substituições necessárias tem-se a nova função dada na equação (15). Observe que nesse caso temos $i = 1$, pois deixaremos de trabalhar com matrizes, passando a trabalhar apenas com vetores, que no caso têm apenas uma linha.

$$
\begin{aligned}
f\left(b_{ij}, a_{ij}^3\right) &= f\left(C(j), a_{ij}^3\right) = \frac{\tan^{-1}(C(j))}{\pi} + \frac{1}{2} + a_{ij}^3 \\
&= \frac{\tan^{-1}(C(j))}{\pi} + \frac{1}{2} - (m-2) + mi + j \\
&= \frac{\tan^{-1}\left(C(j)\right)}{\pi} + \frac{1}{2} - m + 2 + mi + j \\
&= \frac{\tan^{-1}\left(C(j)\right)}{\pi} + \frac{5}{2} - m + m + j \\
&= \frac{\tan^{-1}(C(j))}{\pi} + \frac{5}{2} + j, \qquad j = 1, \dots, m. \qquad (15)
\end{aligned}
$$

Com a função f dada na equação (15) é possível armazenar apenas os coeficientes transformados e que estão acima do limiar definido, em um vetor de armazenamento (que chamaremos de V). V recebe em sua primeira posição o comprimento do vetor C, e nas demais receberá o número obtido da equação (15). Usando a inversa da função f torna-se possível reconstruir qualquer imagem comprimida de dimensões $m \times n$, $m, n \in \mathbb{N}^*$. Para encontrar a função f^{-1} basta fazer substituições análogas na equação (14).

No trabalho de Travassos, Duarte e Villarreal (2018) a wavelet usada para decomposição e reconstrução da imagem foi a db6. Neste trabalho, propõe-se sua substituição pela biortogonal 4.4 (bior4.4), com todos os níveis de decomposição.

Silva (2008) justificou sua escolha pela wavelet db6 afirmando que a mesma possuía o número ideal de momentos nulos, segundo a literatura. O número de momentos nulos é uma propriedade útil na compressão de imagens, pois de modo genérico, mais momentos nulos implicam em uma melhor compressão. Desse modo, os trabalhos de Silva, Duarte e Villarreal (2011), Travassos, Duarte e Villarreal (2016) e Travassos, Duarte Villarreal (2018) seguiram utilizando a db6.

Existem duas famílias de funções muito usadas em aplicações de processamento de imagens: as wavelets ortogonais de Daubechies e as wavelets biortogonais. Apesar da popularidade das wavelets de Daubechies, por sua ortogonalidade e suporte compacto, a bior4.4 tam-

TRAVASSOS, N. C. L.; DUARTE, M. A. Q.; ALVARADO, F. V. Estudo de revisão e proposta de aprimoramento para o armazenamento de imagens comprimidas via transformada wavelet. **C.Q.D.– Revista Eletrônica Paulista de Matemática**, Bauru, v. 14, p. 170-184, fev. 2019. Edição Ermac.
DOI: 10.21167/cqdvol14ermac20192316966ncltmaqdfva170184 Disponível em: https://www.fc.unesp.br/#!/departamentos/matematica/revista-cqd/

bém é fortemente usada nos trabalhos de compressão de imagens, pois fornece bons resultados, haja vista que os filtros biortogonais proporcionam suporte compacto, simetria e regularidade, sendo estas condições essenciais para conseguir um bom desempenho na compressão de imagens (SADASHIVAPPA; ANANDABABU, 2009). Sendo assim, este trabalho realizou diversos testes com um conjunto de imagens de tamanhos distintos, e usando diversas funções ortogonais e biortogonais. Dos resultados obtidos, a wavelet biortogonal bior4.4 obteve bom desempenho em todos os testes, não necessariamente sendo a melhor, mas estando sempre entre as melhores e com desempenho superior a db6 em cerca de 85% dos testes, além de fornecer menos coeficientes de decomposição e, por isso, foi a escolhida.

3.1 Medidas para avaliação da qualidade das imagens comprimidas

Seja $X(i,j)$ a imagem original e $X_c(i,j)$ a imagem reconstruída de dimensões $m \times n$. As métricas usadas para avaliar a qualidade da imagem reconstruída foram MSE, $PSNR$ e $SSIM$. As fórmulas do MSE, PSNR e SSIM, são apresentadas nas equações (16), (17) e (19), respectivamente.

O MSE (*Mean Square Error*) representa o erro quadrático médio entre a imagem original e uma com compressão. Ele é a soma do quadrado das diferenças de cada ponto da imagem original e da imagem reconstruída, dividido pela multiplicação das dimensões da imagem. Quanto menor o valor do MSE, melhor a imagem reconstruída se aproxima da original.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m=1} \sum_{j=0}^{n=1} |X(i,j) - X_c(i,j)|^2 . \tag{16}$$

A $PSNR$ (*Peak Signal to Noise Ratio*) é medida em decibéis. Quanto maior seu valor, melhor a qualidade da imagem comprimida ou reconstruída. Valores típicos para compressão com perdas de uma imagem estão entre 30 e 50 dB. Quando a $PSNR$ é maior que 40 dB, as duas imagens são indistinguíveis (WALKER; NGUYEN, 2001).

$$PSNR = 10.\log_{10}\left(\frac{{L_{max}}^2}{MSE}\right). \tag{17}$$

sendo L_{max} o valor máximo de intensidade de cinza.

As métricas MSE e $PSNR$ são conhecidas por se correlacionar fracamente com a qualidade visual para a maioria das aplicações, sendo assim, será usada mais uma métrica chamada $SSIM$ (*Structural Similarity*). A $SSIM$ é uma métrica de qualidade de imagem que avalia o impacto visual de três características de uma imagem: luminância, contraste e estrutura (WANG et al, 2004). Assim, o índice $SSIM$ de avaliação de qualidade baseia-se no cálculo de três termos, a saber, o termo de luminância, o termo de contraste e o termo estrutural. O índice geral é uma combinação multiplicativa dos três termos, dada pela equação (18).

$$SSIM(x,y) = l(x,y)^{\alpha}.c(x,y)^{\beta}.e(x,y)^{\gamma}. \tag{18}$$

sendo

$$l(x,y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right),$$

$$c(x,y) = \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right),$$

$$e(x,y) = \left(\frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}\right).$$

onde μ_x, μ_y, σ_x, σ_y e σ_{xy} são as médias locais, os desvios padrão e a covariância cruzada para as imagens x,y. C_1, C_2 e C_3 são as constantes de regularização para a luminância, contraste e termos estruturais, respectivamente. A $SSIM$ usa essas constantes de regularização para evitar instabilidade em regiões de imagem em que a média local ou o desvio padrão é próximo de zero. Portanto, pequenos valores diferentes de zero devem ser usados para essas constantes.

Se $\alpha = \beta = \gamma = 1$ (o padrão para os expoentes) e $C_3 = C_2/2$ (seleção padrão de C_3), pode-se simplificar o índice conforme a equação (19).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{19}$$

4 Resultados

Nesta seção, as melhorias propostas serão aplicadas em algumas imagens. As imagens utilizadas nos testes foram: Lena, Airplane e Monarch, de tamanhos 128×128, 256×256, e 768×512, respectivamente, mostradas na Figura 2. As figuras de 3 a 8 mostram os resultados para diferentes taxas de compressão aplicadas às imagens de teste. As tabelas de 1 a 6 mostram os valores da $PSNR$, MSE e $SSIM$ para cada imagem comprimida. A fim de comparação foram usadas duas wavelets, db6 e bior4.4.

Figura 2 - Imagens originais

(a) Lena original (a) Airplane original (a) Monarch original

Fonte: Public-Domain Test Images for Homeworks and Projects[1]

Figura 3 - Imagem Lena com diferentes taxas de compressão usando a wavelet bior4.4

(a) 50% (b) 60% (c) 70%

[1] Disponível em: <http://homepages.cae.wisc.edu/~ece533/images/index.html>. Acesso em: 29 jan. 2019.

(d) 80% (e) 90%

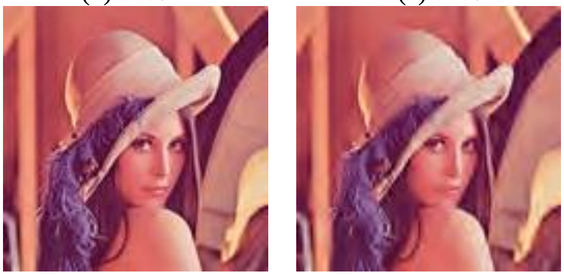


Tabela 1 - Métricas de avaliação da qualidade da imagem Lena com compressão usando a wavelet bior4.4

Imagem	PSNR	MSE	SSIM
(a)	47.7390	1.0944	0.9996
(b)	43.7136	2.7652	0.9990
(c)	39.2572	7.7154	0.9973
(d)	34.3257	24.0167	0.9918
(e)	28.5026	91.7944	0.9711

Figura 4 - Imagem Lena com diferentes taxas de compressão usando a wavelet db6

(a) 50% (b) 60% (c) 70%

(d) 80% (e) 90%

Tabela 2 - Métricas de avaliação da qualidade da imagem Lena com compressão usando a wavelet db6

Imagem	PSNR	MSE	SSIM
(a)	45.9387	1.6566	0.9994
(b)	41.6334	4.4641	0.9984
(c)	37.0161	12.9260	0.9955
(d)	32.1471	39.6615	0.9866
(e)	26.3979	149.0361	0.9538

Figura 5 - Imagem Airplane com diferentes taxas de compressão usando a wavelet bior4.4

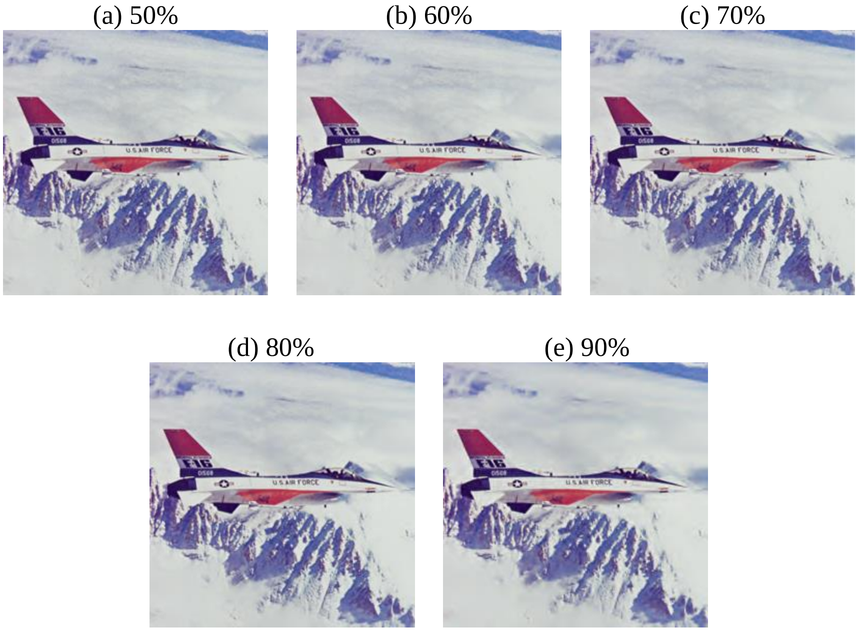


Tabela 3 - Métricas de avaliação da qualidade da imagem Airplane com compressão usando a wavelet bior4.4

Imagem	PSNR	MSE	SSIM
(a)	51.0215	0.5140	0.9957
(b)	47.6886	1.1072	0.9919
(c)	44.0525	2.5576	0.9855
(d)	39.5517	7.2095	0.9736
(e)	33.5634	28.6247	0.9434

Figura 6 - Imagem Airplane com diferentes taxas de compressão usando a wavelet db6

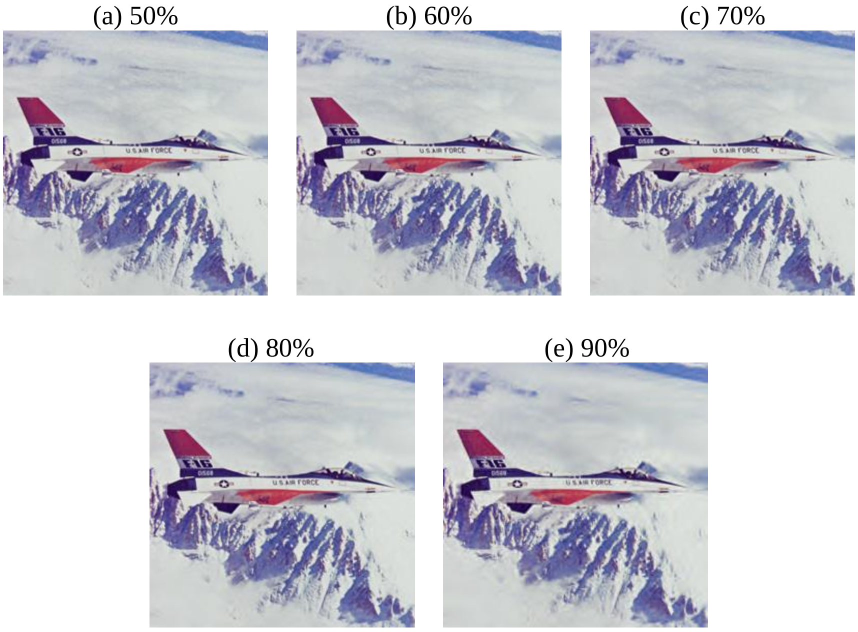


Tabela 4 - Métricas de avaliação da qualidade da imagem Airplane com compressão usando a wavelet db6

Imagem	PSNR	MSE	SSIM
(a)	50.1710	0.6251	0.9948
(b)	46.6927	1.3926	0.9903
(c)	42.8474	3.3755	0.9824
(d)	38.2394	9.7530	0.9680
(e)	32.2721	38.5360	0.9290

Figura 7 - Imagem Monarch com diferentes taxas de compressão usando a wavelet bior4.4

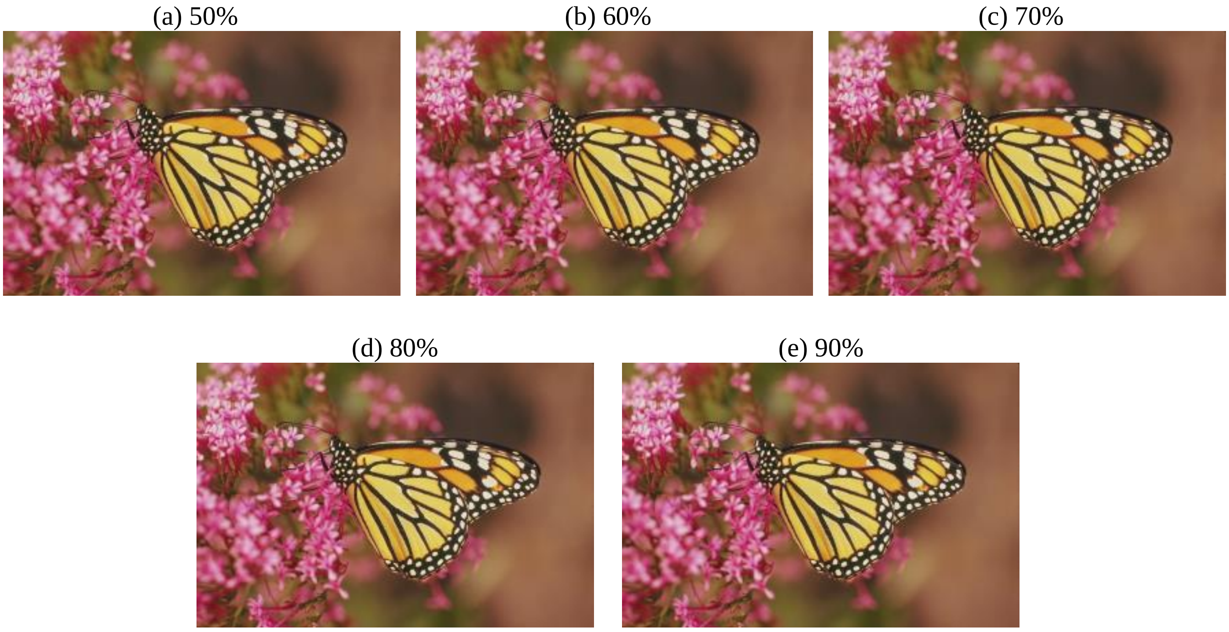
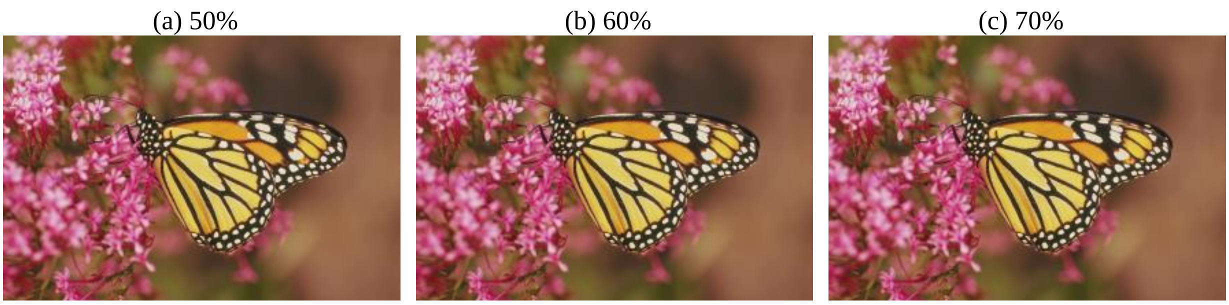


Tabela 5 - Métricas de avaliação da qualidade da imagem Monarch com compressão usando a wavelet bior4.4

Imagem	PSNR	MSE	SSIM
(a)	52.3088	0.3821	0.9997
(b)	49.7853	0.6832	0.9995
(c)	47.3234	1.2043	0.9991
(d)	44.5410	2.2855	0.9984
(e)	39.4042	7.4587	0.9960

Figura 8 - Imagem Monarch com diferentes taxas de compressão usando a wavelet db6



(d) 80%

(e) 90%

Tabela 6 - Métricas de avaliação da qualidade da imagem Monarch com compressão usando a wavelet db6

Imagem	PSNR	MSE	SSIM
(a)	51.9717	0.4130	0.9997
(b)	49.4463	0.7387	0.9994
(c)	46.9165	1.3226	0.9990
(d)	43.9240	2.6344	0.9982
(e)	38.2801	9.6621	0.9950

De acordo com as tabelas apresentadas, a wavelet bior4.4 mostra melhor desempenho quando comparada a db6. Os valores da $PSNR$, MSE e $SSIM$ obtidos ao processar cada imagem atestam o melhor desempenho da bior4.4 provando seu impacto na qualidade da imagem reconstruída.

O tamanho da imagem pode ser entendido como a qualidade da imagem. Portanto, quanto maior a dimensão, mais imperceptível será a degradação visual e melhor será a qualidade da imagem reconstruída, embora a qualidade também dependa da taxa de compressão, uma vez que altas taxas de compressão para imagens de baixa resolução podem resultar em degradações perceptíveis.

Os testes realizados mostraram que a db6 tem baixo desempenho para as imagens de dimensão menor, ao passo que apresenta bons resultados para as imagens de dimensão maior. Porém, os resultados da bior4.4 foram satisfatórios para os três tamanhos de imagens testados. Dependendo da imagem, existem outras wavelets com resultados melhores, porém a bior4.4 foi escolhida por sempre estar entre as funções que forneceram melhor $PSNR$, MSE e $SSIM$ nos testes realizados para a escolha da wavelet a ser usada.

5 Considerações finais

Este trabalho apresentou uma revisão de quatro métodos recentes de compressão e codificação de imagens, propondo a extensão para imagens de qualquer dimensão e com novas métricas de avaliação. Os trabalhos de Silva (2008), Silva, Duarte e Villarreal (2011) e Travassos, Duarte e Villarreal (2016) são voltados ao processamento de imagens quadradas de dimensão 2^k, em escala de cinza. Travassos, Duarte e Villarreal (2018) estenderam o processamento às imagens coloridas, e em complemento a este trabalho, o método proposto processa imagens de tamanho $m \times n$ com m, n quaisquer, abrangendo assim todos os tipos de imagem.

Além de estender a codificação para imagens de qualquer dimensão, foram feitos testes com outros filtros wavelets verificando que a wavelet bior4.4 possibilita melhores resultados que a wavelet que foi usada nos quatro métodos anteriores. A eficiência da nova proposta pode ser verificada ao se aplicar as medidas de qualidade usadas nos testes, pois os valores obtidos mostram que é possível obter altas taxas de compressão, e consequentemente uma codificação que ocupe pouco espaço de memória, sem que haja perda de qualidade na imagem restaurada.

Vale a pena ressaltar que as mudanças alcançadas pelo uso da wavelet bior4.4 vão além dos avanços na qualidade da imagem reconstruída, pois a decomposição com tal wavelet fornece menos coeficientes em relação a decomposição com a db6. Sendo assim, como são necessários menos coeficientes para representar uma mesma imagem, o tamanho do vetor de codificação terá menos elementos, fazendo com que seu tempo de decodificação seja reduzido.

6 Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

7 Referências bibliográficas

BAIRAGI, V. K. Symmetry-based biomedical image compression. **Journal of Digital Imaging**, v. 28, n. 6, p. 718-726, 2015.

COIFMAN, R. Adapted multiresolution analysis, computation, signal processing and operator theory. In: **Proceedings of the International Congress of Mathematicians**. Kyoto: Springer-V., 1990. p. 879-887.

DAUBECHIES, I. **Ten lectures on wavelets**. Philadelphia: Siam, 1992.

DONOHO, D. L.; JOHNSTONE, J. M. Ideal spatial adaptation by wavelet shrinkage. **Biometrika**, v. 81, n. 3, p. 425-455, 1994.

GOMES, J.; VELHO, L.; GOLDENSTEIN, S. **Wavelets:** teoria, software e aplicações. Rio de Janeiro: IMPA, 1997.

MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 11, n. 7, p. 674-693, 1989.

PAN, H.; SIU, W-C; LAW, N-F. A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT. **Signal Processing:** Image Communication, v. 23, n. 3, p. 146-161, 2008.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais:** princípios, algoritmos e aplicações. São Paulo: Thomson Learning, 2008.

SADASHIVAPPA, G.; ANANDABABU, K. V. S. Evaluation of wavelet filters for image compression. **International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering**, v. 3, n. 3, p. 430-436, 2009.

SILVA, J. F. **Sistema de armazenamento de imagens comprimidas através da transformada wavelet**. 2008. 100 f. Dissertação (Mestrado em Engenharia Elétrica)-Universidade Estadual Paulista “Julio de Mesquita Filho”, Ilha Solteira, 2008.

SILVA, J. F.; DUARTE, M. A. Q.; VILLARREAL, F. Um método para codificação de imagens digitais no domínio wavelet. In: CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES, 10., 2011, Águas de Lindóia. **Anais…** São Carlos: SBMAC, 2011. p. 731-734.

STOLLNITZ, E. J.; DEROSE, T. D.; SALESIN, D. H. Wavelets for computer graphics: a primer, part 1. **IEEE Computer Graphics and Applications**, v. 15, n. 3, p. 76-84, 1995.

TRAVASSOS, N. C. L.; DUARTE, M. A. Q.; VILLARREAL, F. Encoding and decoding compressed images in the wavelet domain. In: ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL, 19.; ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS, 7., 2016, João Pessoa. **Anais...** João Pessoa: Universidade Federal da Paraíba, 2016.

TRAVASSOS, N. C. L.; DUARTE, M. A. Q.; VILLARREAL, F. Estudo e aprimoramento de métodos para armazenamento de imagens comprimidas. In: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 5., 2018, Bauru. **Caderno de trabalhos completos e resumos...** Bauru: Unesp, Faculdade de Ciências, 2018. p. 140-146. Disponível em:< https://drive.google.com/file/d/1-8SZZnioKApdmQG6BaOTltJARzfwAB9P/view>. Acesso em: 29 jan. 2019.

WALKER, J. S.; NGUYEN, T. Q. Wavelet-based image compression. In: RAO, R.; YIP; P.C. (Ed.). **The transform and data compression handbook**. Boca Raton: CRC, c2001.

WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600-612, 2004.

WU, K.; OTOO, E. J.; SHOSHANI, A. Optimizing bitmap indices with efficient compression. **ACM Transactions on Database Systems**, v. 31, n. 1, p. 1-38, 2006.