

Codificador de Imagens *Wedgelet-Wavelet* por Blocos

Ricardo Barroso Leite, Ana Lúcia Mendes Cruz, Fernando Silvestre da Silva e Yuzo Iano

Abstract — The great development of mobile devices has an impact on the visual perception of users, which are increasingly demanding. In a noisy channel and with reduced bandwidth, low transmission rates are allowed, leading to visual artifacts, such as ringing. In this context, several studies have been performed, culminating in new basis and geometric dictionaries, such as *wedgelets*. This paper proposes a new method of image compression in a hybrid *wedgelet-wavelet* scheme, according to the *cartoon-texture* methodology, gaining performance by exploiting the geometry of images. We present comparisons with algorithms based on pure *wavelets* and *wedgelets*, and with other hybrid methods and compression standards. Comparisons of PSNR, processing time, and subjective evaluation for various images and bitrates are presented. The primary results show a subjective gain at low data rates, indicating that the methodology could be included in transmission systems with restricted data rate.

Index Terms — image compression, geometric dictionaries, *wedgelets*, image quality.

Resumo — O grande desenvolvimento dos dispositivos móveis tem impactado na percepção visual dos usuários, cada vez mais exigentes. Em um canal ruidoso e com reduzida largura de banda são necessárias baixas taxas de transmissão, levando a artefatos visuais, como *ringing*. Nesse contexto, diversas pesquisas têm sido realizadas, culminando em novas bases e dicionários geométricos, tais como *wedgelets*. Este trabalho propõe um novo método de compressão de imagens híbrido *wedgelet-wavelet* dentro da metodologia *cartoon-textura*, objetivando ganho de desempenho por meio da exploração da geometria das imagens. São apresentadas comparações com algoritmos baseados em *wavelets* e *wedgelet* puros, com outros métodos híbridos e com padrões de compressão completos. Comparações de PSNR, tempo de processamento, e avaliação subjetiva para diversas imagens e bitrates são mostradas. Os ganhos principalmente em qualidade subjetiva para baixas taxas de dados indicam que a metodologia poderia ser incluída em sistemas de transmissão de dados com restrições de taxa.

Palavras chave — compressão de imagens, dicionários geométricos, *wedgelets*, qualidade de imagem.

I. INTRODUÇÃO

Os avanços em comunicações e equipamentos digitais, a disputa de mercados de TV digital e principalmente a padronização do sistema de HDTV (*High Definition Television* – Televisão de Alta Definição), impulsionaram diversas áreas de pesquisa relacionadas à compressão de imagens em comunicações fixas e móveis. No ambiente de comunicações móveis, as restrições são consideravelmente maiores dada a característica extremamente ruidosa do canal e a reduzida largura de banda disponível para manter a operabilidade do sistema. Atingir baixas taxas de transmissão é portanto o foco dos sistemas de compressão envolvidos neste caso. Tais taxas de operação muitas vezes acarretam artefatos

visuais inerentes ao processo de compressão. Com o desenvolvimento de uma gama cada vez maior de dispositivos móveis, é inevitável que surjam problemas na percepção visual dos usuários finais, que estão cada vez mais exigentes em termos de definição e resolução de imagens estáticas e vídeo.

Nos sistemas de compressão, a transformada *wavelet* discreta (TWD) tem obtido grande eficiência em termos de qualidade visual e PSNR (*Peak Signal to Noise Ratio* – Relação Sinal-Ruído de Pico), apresentando algoritmos rápidos, alta concentração de energia, e ausência de blocagem, tendo sido inserida nos padrões JPEG2000 [2], MPEG-4 e MPEG-7 [3].

Entretanto, mesmo imagens codificadas com *wavelets* apresentam artefatos característicos, como *ringing* e embaçamento. Diversas pesquisas têm sido realizadas, utilizando quantização adaptativa e métodos híbridos, para tentar minimizá-los [4]. Propôs-se até mesmo a utilização de novos espaços de funções, as chamadas *directionlets* [5], *curvelets* [6], *bandelets* [7]-[9] e dicionários geométricos como *wedgelets* [10]. Destaque segue para dicionários *wedgelets* de primeira ordem propostos por Donoho [10] e a generalização de Lisowska [11], dentre outros, definindo *arcllets*, *platelets* e *surflets* [12]-[15].

Modelos híbridos combinando *wedgelets* com *wavelets* foram desenvolvidos excelentemente por Wakin et al [16][17], mas ainda há poucos dados disponíveis, tornando este um campo que tem muito a ser explorado.

Nesse contexto, este trabalho propõe um novo método de compressão de imagens híbrido *wedgelet-wavelet* dentro da metodologia *cartoon-textura*, objetivando ganho de desempenho por meio da exploração da geometria das imagens, a ser utilizado em aplicações de comunicações móveis. Do ponto de vista de contribuições inéditas, este trabalho propõe o referido novo método que escolhe o melhor dicionário para cada região da imagem, de acordo com a relação taxa-distorção e a característica *cartoon* ou textura da imagem; a otimização dos parâmetros da *wavelet* para codificação de imagens, buscando minimizar erro objetivo; e uma modificação no algoritmo SPIHT (*Set Partitioning in Hierarchical Trees* – Partição de Conjuntos em Árvores Hierárquicas) de codificação usando *wavelets* para otimizar a relação taxa-distorção ao invés de utilizar taxa de dados pré-fixada.

Os ganhos principalmente em qualidade subjetiva das imagens reconstruídas para baixas taxas de dados indicam que a metodologia apresentada poderia ser incluída em sistemas de transmissão de dados com restrições de taxa.

O presente artigo está organizado como se segue: na seção II são apresentadas as novas formas de representação esparsa e a motivação para seu uso em compressão de imagens; na

seção III é mostrado o dicionário *wedgelet* de 1ª ordem e a seção IV lista modelos híbridos. A seção V apresenta o codificador proposto. Na seção VI residem os resultados experimentais e na seção VII são finalizadas as conclusões. Resultados preliminares e mais detalhes sobre a metodologia proposta podem ser encontrados em [1].

II. O PROBLEMA DAS TRANSFORMADAS ISOTRÓPICAS E O USO DE NOVAS FORMAS DE REPRESENTAÇÃO ESPARSA

Para comprimir imagens digitais, usualmente o sinal é convertido do domínio original para uma nova forma de representação, que destaque suas características e isole componentes desejadas. Representações esparsas são aquelas em que a maior parte das componentes é nula ou insignificante, podendo ser desprezadas sem grandes perdas. Transformadas ortogonais ou biortogonais podem ser usadas para obter a representação esparsa, como no caso da TWD, com a vantagem de isolar completamente as componentes.

Porém, as transformadas ortogonais têm limitações na capacidade de concentrar dados em poucas componentes para sinais com descontinuidades, como imagens tipo *cartoon*. Uma imagem *cartoon* consiste de contornos e regiões com cores homogêneas, como em um desenho [18]. O erro de aproximação depende da regularidade. Para sinais suaves, aproximações lineares e não-lineares têm desempenho similar [19]. Quando a regularidade do sinal não é uniforme, aproximações não-lineares em bases *wavelet* são utilizadas.

Adicionalmente, problemas surgem quando se trata de mais dimensões, devido à importância da geometria na qualidade visual. Essas aproximações são excelentes para imagens de variação limitada, mas não para imagens naturais contendo bordas geometricamente regulares. Além disso, *wavelets d*-dimensionais são usualmente construídas com 2^d-1 produtos tensoriais de *wavelets* 1D (separáveis). Singularidades multidimensionais podem estar organizadas ao longo de subconjuntos de acordo com a geometria, como acontece com contornos de uma imagem no caso 2D, e portanto podem não ser representadas eficientemente por *wavelets* separáveis.

A ineficiência da *wavelet* em representar contornos por não explorar a regularidade deles é mostrada na Fig. 1. O número de coeficientes necessário para representar uma descontinuidade pode ser reduzido conhecendo-se a geometria, evitando artefatos visuais quando em baixa taxa de dados, como o *ringing* (Fig. 2).

Para aproximar funções suaves por partes e com geometria não-separável e para tratar a deficiência da TWD em explorar a geometria, autores têm recorrido a dicionários redundantes, similares a esquemas de quantização vetorial, além das transformadas ortogonais.

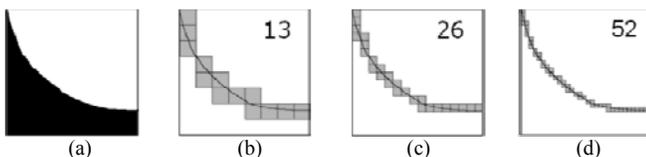


Fig. 1 Ineficiência da *wavelet* isotrópica em representar geometria. Coeficientes: (a) 13 (b) 26 (c) 52. [20]



Fig. 2 – *ringing* gerado pela quantização dos coeficientes *wavelet*

O uso de dicionários redundantes para a codificação é defendido por Mallat [19], pois dicionários pequenos como bases ortogonais têm uma capacidade limitada de representação esparsa, que é necessária quando se trabalha com altas taxas de compressão. Dicionários maiores, que incorporem mais padrões, podem aumentar a esparsidade e assim inovar em aplicações de compressão, remoção de ruído, problemas inversos e reconhecimento de objetos.

Para desenvolver o melhor dicionário devem-se adaptar as bases para características mais comuns nas imagens: bases com funções suaves por partes com descontinuidades bem localizadas representam bem estruturas do tipo *cartoon* e bases com funções oscilatórias, localmente estacionárias, representam estruturas do tipo textura.

III. DICIONÁRIO WEDGELET DE 1ª ORDEM

Neste contexto, surgiu o dicionário de *wedgelets* proposto por Donoho [10], com desempenho superior às TWD para altas taxas de compressão. Um *wedgelet* é definido como uma função constante por partes em um quadrado diádico S contendo uma descontinuidade ao longo de uma linha com orientação θ e deslocamento d , como mostrado na Fig. 3.c.

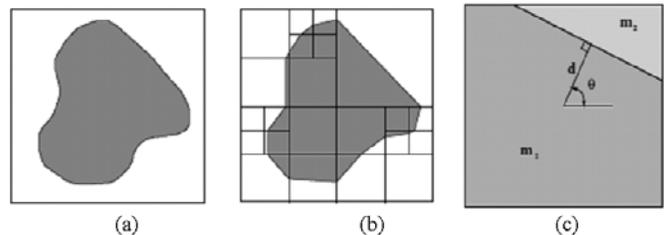


Fig. 3 – Decomposição *wedgelet* (a) Imagem (b) Decomposição multi-escala (c) Parametrização de um *wedgelet* em um quadrado $N \times N$ diádico.

Uma representação *wedgelet* de uma imagem pode ser realizada em blocos de tamanho pré-determinado (Fig. 4.a) ou iterativamente em blocos de diferentes resoluções, organizados em *quadtree* (Figs. 3.b e 4.b). Os nós da *quadtree* definem a partição diádica e nas folhas são armazenados os parâmetros do *wedgelet* que aproximam o segmento de imagem no correspondente quadrado diádico. Da mesma forma que nos algoritmos de aproximações não-lineares em *quadtree wavelet* pode-se cortar a descendência da árvore em escalas mais baixas nas regiões onde a imagem é suave, na *quadtree wedgelet* pode-se cortar a árvore em regiões onde a imagem é bem aproximada por um *wedgelet* em termos de erro quadrático. Assim, imagens suaves por partes (*cartoon*) podem ser bem aproximadas usando poucas componentes.

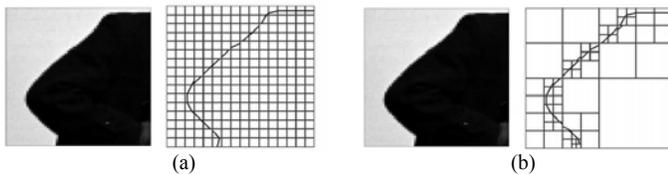


Fig. 4 – Decomposição *wedgelet* de um trecho de imagem *cartoon* (a) escala única (tamanho dos blocos fixo) (b) multi-escala [21]

Em [10] Donoho introduziu o conceito de funções *horizon* para descrever um sinal *cartoon* hipotético que pode ser bem representado usando *wedgelets*. Considerando uma função 1D $H(x)$ chamada de *horizon* (horizonte), definida no intervalo $[0,1]$ e uma imagem f da forma:

$$f(x_1, x_2) = \mathbf{1}_{\{x_2 \geq H(x_1)\}} \quad 0 \leq x_1, x_2 \leq 1 \quad (1)$$

A imagem é definida como “branca” ($f = 1$) para coordenadas acima do *horizon* e “preta” ($f = 0$) para coordenadas abaixo. Deseja-se estudar casos em que $H(x)$ é regular (denotada por $H_{\text{or}}^u([0,1]^2)$). Considerando apenas contornos não degenerados, (que não estejam na fronteira de um quadrado), define-se o *wedgelet* $w(x_1, x_2)$ da partição como:

$$w(x_1, x_2) = m_1 \cdot H_c + m_2 \cdot (1 - H_c), \text{ onde} \quad (2)$$

$$H_c = \mathbf{1}_{\{x_2 \leq b(x_1)\}} \quad (x_1, x_2) \in S \quad (3)$$

O melhor *wedgelet* para um determinado segmento de imagem é escolhido de forma exaustiva. Definido um conjunto discreto de possíveis orientações e distâncias (θ, d) , é calculado o erro quadrático de cada aproximação possível $w_{\theta, d}(x_1, x_2)$ e armazena-se o mínimo. Os valores de m_1 e m_2 para cada *wedgelet* são a média do segmento da imagem na região acima ou abaixo da linha.

$$m_1 = E[f(x_1, x_2) \cdot H_c] \quad m_2 = E[f(x_1, x_2) \cdot (1 - H_c)]$$

$$w(x_1, x_2) = \min_{\theta, d} \|f - f_w\|^2 \quad (4)$$

onde f é a imagem original e f_w é a aproximação *wedgelet*.

Diversos métodos poderiam ser usados para obter a melhor projeção *wedgelet* de um bloco $N \times N$, sendo a busca exaustiva no \mathbb{R}^4 para todos os parâmetros θ , d , m_1 e m_2 o mais pesado computacionalmente. Estimativas do melhor *wedgelet* podem ser obtidas de forma mais rápida pela análise da transformada de Radon ou por transformada *wavelet* complexa [22]. Em [23][24] são apontadas as sugestões mais recentes para otimizar a projeção, usando dicionários discretos e com cálculos parciais específicos para a geometria.

O dicionário *wedgelet* é definido como o conjunto:

$$W = \left\{ \begin{array}{l} w_{i,j,m} : i = 0, \dots, \log_2 N, j = 0, \dots, 4^i - 1, \\ m = 0, \dots, M_W(S_{i,j}) - 1 \end{array} \right\} \quad (5)$$

e a representação *wedgelet* é definida pela seguinte fórmula:

$$f_w(x_1, x_2) = \sum_{i,j,m} w_{i,j,m}(x_1, x_2) \quad (6)$$

Alguns coeficientes $w_{i,j,m}$ da representação podem ser nulos. No primeiro passo é feita uma decomposição completa da imagem, encontrando as melhores representações *wedgelet*

para todos os níveis de resolução em cada quadrado $S_{i,j}$, $0 \leq j < 4^i$, $0 \leq i < \log_2 N$ de acordo com o erro quadrático. Todos os coeficientes *wedgelet* são armazenados como nós da *quadtrees*. No segundo passo, é aplicado o algoritmo de otimização, removendo nós em níveis avançados da árvore, de forma a obter um número mínimo de nós na representação, mas assegurando a qualidade desejada. Busca-se minimizar a seguinte função de custo [45]:

$$R = \min_P \left\{ \|f - f_w\|^2 + \lambda \cdot |P| \cdot K \right\} \quad (7)$$

onde P denota a partição da imagem, K o número de bits necessário para codificar os nós e λ o parâmetro de otimização taxa-distorção. Para reconstrução exata, $\lambda = 0$ e a aproximação *wedgelet* é a imagem original. Usualmente a minimização é feita dos blocos menores para os maiores (*bottom-up*). Além disso, pode-se explorar a regularidade dos contornos, incorporando modelos multiescala ao algoritmo de corte da *quadtrees*, favorecendo *wedgelets* com orientações alinhadas entre os blocos diádicos, ou com pequenos desvios sobre previsões de níveis mais altos [20].

Para a análise do erro das aproximações *wedgelets*, define-se o conjunto \mathcal{W} de todas as possíveis partições com um segmento de contorno (*wedge*) em um quadrado unitário. Para uma dada partição *wedgelet* $\mathcal{W} \in \mathcal{W}$, seja $V(\mathcal{W})$ o espaço linear de funções lineares por partes relativo a \mathcal{W} :

$$V(\mathcal{W}) = \left\{ f|_{\mathcal{W}}(x) = \sum_{W \in \mathcal{W}} P_W(x) \chi_W(x), \text{ para alguma } P_W \in \Pi_1 \right\} \quad (8)$$

sendo Π_1 o conjunto de polinômios de duas variáveis de grau no máximo 1.

Considerando um sinal $f \in L^2([0,1]^2)$ e denotando $f(\mathcal{W})$ como a projeção ortogonal de f no espaço $V(\mathcal{W})$, $f(\mathcal{W})$ é a única função que minimiza a o erro quadrático L^2 :

$$\|f - f(\mathcal{W})\|_{L^2([0,1]^2)} = \inf_{g \in V(\mathcal{W})} \|f - g\|_{L^2([0,1]^2)} \quad (9)$$

pode-se definir uma aproximação não-linear de n termos, como anteriormente, escolhendo a partição \mathcal{W} tal que $|\mathcal{W}| = n$:

$$\|f - f(\mathcal{W}_n)\|_{L^2([0,1]^2)} = \inf_{\mathcal{W} \in \mathcal{W}_n} \|f - f(\mathcal{W})\|_{L^2([0,1]^2)} \quad (10)$$

onde

$$\mathcal{W}_n = \{ \mathcal{W} \in \mathcal{W} \mid |\mathcal{W}| \leq n \} \quad (11)$$

Para determinar a eficiência da aproximação, define-se W_α :

$$W_\alpha = \left\{ f \in L^2[0,1]^2 \mid \|f\|_{W,\alpha} < \infty \right\} \quad (12)$$

$$\|f\|_{W,\alpha} = \sup_{\mathcal{W} \in \mathcal{W}_n} (n^{\frac{\alpha}{2}} \|f - f(\mathcal{W}_n)\|_{L^2([0,1]^2)}) \quad (13)$$

O conjunto W_α é chamado classe de aproximação de ordem α e atua similarmente à norma do espaço de Sobolev. Quando o sinal possui descontinuidades e $1 \leq \alpha \leq 2$, a aproximação *wedgelet* tem taxa de decaimento mais rápida do que a aproximação não-linear pela TWD [10]. Assim:

Seja $\alpha \in [1,2]$, $f \in \text{Hor}^u([1,2]^2)$, e h a função que define o contorno de f , existem constantes $C, C' > 0$ independentes de n e para cada n existe uma partição *wedgelet* \mathcal{W}_n sobre o quadrado unitário $[0,1]^2$, tal que $|\mathcal{W}_n| \leq C'n$ e satisfaz:

$$\|f - f_n(\mathcal{W}_n)\|_{L^2([0,1]^2)} \leq \frac{C'}{n^{\frac{\alpha}{2}}} \quad (14)$$

onde $f_n = f_n(W_n)$ é a projeção L^2 de f no espaço de funções *wedgelets* sobre a partição W_n . Para imagens horizon, a aproximação não-linear por *wavelets* tem uma taxa de decaimento de erro mais lenta:

$$\|f - \hat{f}_n\|_{L^2[0,1]^2} = O\left(\frac{C}{n^2}\right) \quad (15)$$

A Fig. 5 ilustra a comparação da aproximação de uma imagem tipo *horizon* por *quadtrees* compostas por nós quadrados constantes (a) e por nós *wedgelet* (b). Para uma reconstrução com qualidade comparável a aproximação *wedgelet* requer aproximadamente metade do número de componentes em relação à *quadtree* constante.

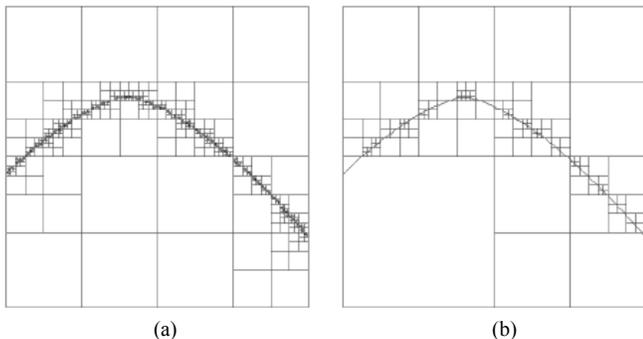


Fig. 5 – Aproximações em *quadtree* (a) 967 nós constantes, PSNR: 33.69 dB. (b) 248 nós *wedgelet* (496 coeficientes *wedgelet*), PSNR: 33.71 dB [99]

IV. MODELOS HÍBRIDOS – DECOMPOSIÇÃO CT

No entanto, modelos *horizon* não representam com perfeição imagens naturais, embora a informação primordial de contornos seja preservada. Para garantir a reconstrução perfeita é preciso completar o modelo com informação não geométrica, de acordo com a metodologia *cartoon*-*textura*.

Alguns modelos híbridos têm sido usados na literatura. A metodologia consiste em determinar um método para codificar a estrutura geométrica dos contornos, como os métodos apresentados nas seções anteriores e idealmente toda a informação não-codificada (resíduo) terá uma estrutura do tipo *textura* e pode ser codificada por meio das aproximações *wavelets* não-lineares com taxa de erro $(N^{-2\alpha})$.

O método proposto por Wakin et al. em [16] utiliza aproximações *wedgelets* preditivas e quantização dos parâmetros dos nós para identificar e o resíduo é codificado usando *wavelets*. Para eliminar efeitos de blocagem indesejados produzidos pela decomposição *wedgelet* com resolução limitada, a componente *cartoon* passa por filtragem passa-baixas. Ao final do processo de decodificação as duas componentes são somadas para compor novamente a imagem. No entanto, esse método não atinge resultados em PSNR superiores a algoritmos mais recentes usando *wavelets*, como SPIHT e EZW, apesar de reduzir o *ringing*.

Em [17] Wakin et al. propõem um método de codificação híbrida *wedgelets-wavelet* que otimiza a relação taxa-distorção para cada subregião da imagem, ao invés de usar o resíduo. Basicamente é uma modificação do algoritmo SFQ (*Space Frequency Quantization* – Quantização Espaço Frequência)

[25], usando *wedgeprints*, projeções dos *wedgelets* no domínio *wavelet* (Fig. 6).

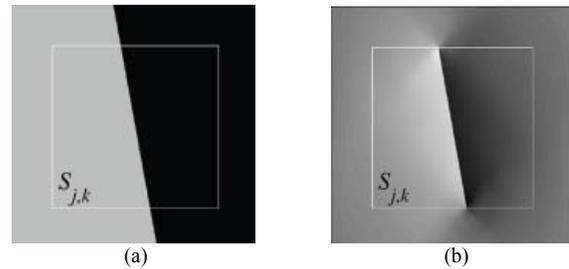


Fig. 6 – *Wedgelets* no domínio *wavelet* (a) porção com um *wedgelet*; (b) *wedgeprint* obtido pela projeção do *wedgelet* na subbanda vertical *wavelet*.

Considerando esses resultados teóricos em termos de aproximação de sinais, as *wedgelets* se apresentam como um paradigma inovador para a representação esparsa da geometria das imagens, ou mais especificamente, para a estrutura *cartoon*. Com o uso híbrido combinado com outros dicionários e uma estrutura em árvore multi-resolução, melhores métodos de codificação ainda podem ser implementados no futuro.

V. MÉTODO PROPOSTO WWB

O método proposto, referenciado no presente artigo por WWB (*Wedgelet-Wavelet por Blocos*), é mostrado na Fig. 7. As etapas tracejadas indicam as modificações inéditas propostas. Primeiramente, a imagem é dividida em blocos de tamanho 64×64 e cada um desses blocos é classificado como *cartoon* (a ser codificado com *wedgelets*) ou *textura* (a ser codificado com *wavelets*).

A decomposição *cartoon*-*textura* usada em outros métodos híbridos (seção IV), é proposta neste trabalho usando uma nova abordagem, fazendo uma classificação em cada bloco. A escolha é feita com base na função de custo taxa-distorção. O bloco é pré-codificado com símbolos *wavelet* e *wedgelet* e o dicionário que determinar a menor função custo é escolhido.

Essa estratégia de procura exaustiva *bottom-up* é o que torna o algoritmo mais complexo, mas também bastante adaptável para as diferentes regiões da imagem. Assim, blocos em que a geometria é mais importante (*cartoon*) tendem a ser escolhidos por *wedgelets* e blocos com mais detalhes são classificados como blocos de *textura*. Após todos os blocos serem lidos, os coeficientes *wavelet* dos blocos *textura* são codificados por uma versão modificada do SPIHT, que otimiza a função taxa-distorção ao invés de manter uma taxa de bits pré-fixada. Os bits gerados pelo SPIHT modificado são então adicionados ao *bitstream* correspondente ao bloco.

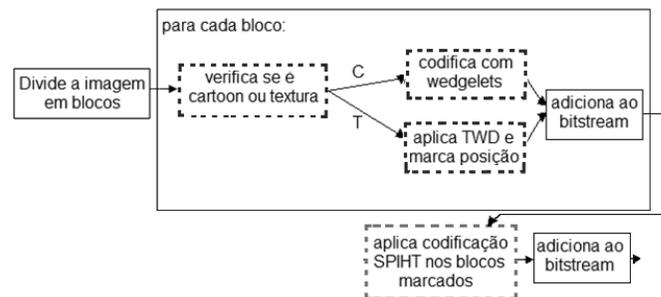


Fig. 7 – Método proposto (WWB).

Para o dicionário *wedgelet*, o processo é dividido em 3 etapas (Fig. 8): geração da árvore *wedgelet* (uma vez para cada imagem e para cada conjunto de parâmetros de dicionário), codificação e decodificação (ambos uma vez para cada taxa escolhida).

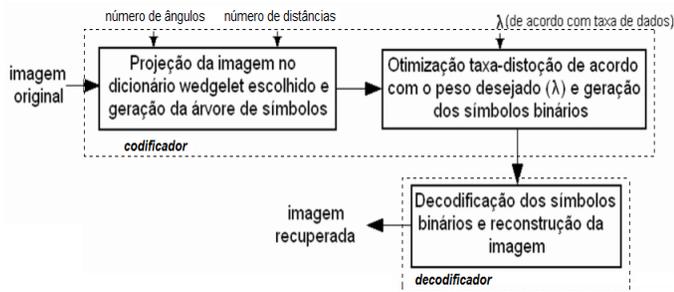


Fig. 8 – Processo de codificação-decodificação *wedgelet*.

Buscou-se inicialmente a configuração de parâmetros (especificamente, número de possíveis *wedges* no dicionário) que otimiza a codificação em termos de maximizar a qualidade em PSNR, trabalhada como uma das modificações propostas no presente método. O número de *wedges* possíveis depende do número de ângulos θ (são consideradas partições uniformes de 0° a 180°) e do número de distâncias d (consideradas partições uniformes de um domínio quadrado). O tamanho do dicionário de *wedges* tem impacto na qualidade da representação da imagem e também no número de bits para codificar cada elemento do dicionário. Para a codificação da imagem, os símbolos do dicionário são transformados em símbolos binários pelo método de Huffman.

Em [1] são apresentadas simulações exaustivas que determinaram os parâmetros ótimos como sendo 8 ângulos e 4 distâncias, considerando qualidades subjetivas, objetivas e tempos de processamento para dois amplos conjuntos de imagens naturais e com característica “desenho”.

Além da otimização do dicionário *wedgelet*, também é necessário determinar os melhores parâmetros para a codificação *wavelet*, como será mostrado na seção de resultados.

VI. RESULTADOS EXPERIMENTAIS

Os algoritmos foram implementados em Matlab 7.5.0 (R2007b) em um computador PC Intel Core 2-Quad 2.50 GHz, 3.24 GB de RAM Windows XP, isolado de outros processos externos, tendo sido utilizada a função *cputime* do Matlab para dar confiabilidade aos valores, em especial com relação aos tempos de processamento. Os resultados obtidos são analisados com base no tempo de processamento, na análise subjetiva, no PSNR, no número de componentes utilizadas para a aproximação e no número de bits por *pixel* (*bitrate*) utilizado na codificação. Foram utilizados 2 conjuntos de 8 imagens 256×256 cada, em escala de cinza: imagens naturais da base de dados Waterloo Bragzone [26] e imagens tipo “desenho” extraídos de artigos de processamento de imagens referenciados no texto. Foram escolhidas imagens com características diversas em cada conjunto.

O método WWB é comparado com a codificação isolada por *wavelets* e por *wedgelets*, destacando a análise subjetiva, dado o objetivo de redução de artefatos visuais. Comparações

adicionais com outros métodos baseados em *wedgelets* foram feitas diretamente do material apresentado nos artigos de referência. Para a comparação com o método constante em [27] foi usado um software desenvolvido pelos autores e disponível em [28]. Para a comparação com o padrão JPEG foi utilizado o codificador presente no Matlab com a função *imwrite*, variando o parâmetro de qualidade de acordo com a taxa de bits desejada. Para a comparação com o padrão JPEG 2000 foi usada a versão aberta OpenJPEG [29].

A. Determinação da Função Wavelet

Faz-se necessário definir a melhor função *wavelet* para a codificação híbrida proposta, tendo sido testadas 2 famílias muito usadas em compressão de imagens: as *wavelets* ortogonais de Daubechies e as *wavelets* biortogonais, também chamadas de CDF (Cohen-Daubechies-Fauveau). Foram utilizadas as funções db1(Haar), db2, db4, CDF 5/3 e CDF 9/7. Os resultados encontram-se na Fig. 9.

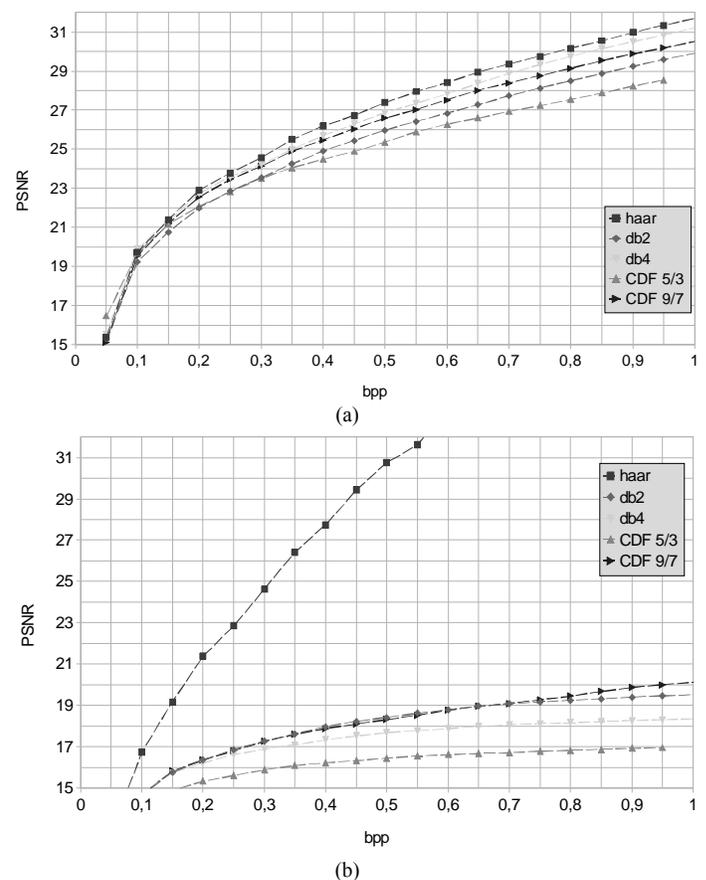


Fig. 9 – Variação da função *wavelet* (a) imagens naturais (b) desenhos.

Observa-se que a função de Haar obteve os melhores resultados para ambos os tipos de imagens. A variação no tempo de processamento entre as funções *wavelet* é desprezível frente aos outros tempos envolvidos no sistema, como é possível ver nas Tabelas 1 e 2.

TABELA 1

TEMPO DE PROCESSAMENTO VARIANDO A WAVELET (IMAGENS NATURAIS)

Função <i>wavelet</i>	Tempo de codificação (s)	Tempo de decodificação (s)
Haar	3,90	3,60
db2	3,90	3,63
db4	3,96	3,67
CDF 5/3	3,96	3,66
CDF 9/7	3,93	3,65

TABELA 2

TEMPO DE PROCESSAMENTO VARIANDO A WAVELET (DESENHOS)

Função <i>wavelet</i>	Tempo de codificação (s)	Tempo de decodificação (s)
Haar	3,77	3,43
db2	3,81	3,50
db4	3,88	3,59
CDF 5/3	3,87	3,52
CDF 9/7	3,84	3,51

B. Codificação Wedgelet-Wavelet

Os resultados do WWB encontram-se na Fig. 10, em comparação com codificações por blocos *wedgelet* e *wavelet* puras utilizando os mesmos parâmetros (8 ângulos, 4 distâncias, *wavelet* Haar). Do ponto de vista de taxa-distorção, para imagens naturais observa-se que o WWB obtém desempenho superior à codificação *wavelet* para baixas taxas (ganhos de até 1,5 dB) e se aproxima da codificação *wedgelet* (menos de 0,5 dB de diferença). Para taxas maiores do que 0,5 bpp o método WWB se aproxima da curva *wavelet* (diferença de 0,3 dB) e supera o método de *wedgelets* sozinho (ganho maior que 0,5 dB). No geral, o WWB tem uma qualidade uniforme em termos de PSNR para todas as taxas. Para as imagens de desenho, o uso da *wavelet* sozinha não possui bom desempenho e o método WWB consegue ser superior aos dois métodos comparados (ganhos em torno de 0,5 dB em relação às *wedgelets* e maiores que 5 dB em relação às *wavelets*). Esse fato, combinado com a análise subjetiva posteriormente apresentada, torna o WWB um bom método de codificação para aplicações que envolvam desenhos, mapas, imagens com textos, dentre outras.

Conforme o esperado, para imagens naturais, a tendência é que o sistema utilize mais componentes *wedgelet* quando são utilizadas baixas taxas de bits e passe a usar *wavelets* nas altas taxas (Fig. 11). Para imagens de desenho, ocorre uma inflexão a partir de 0,5 bpp, isto é, quando em altas taxas o sistema continua usando componentes *wedgelets*, dada a ausência de detalhes na imagem.

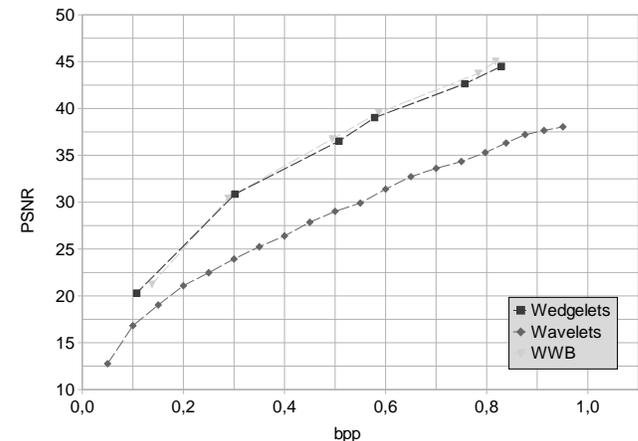
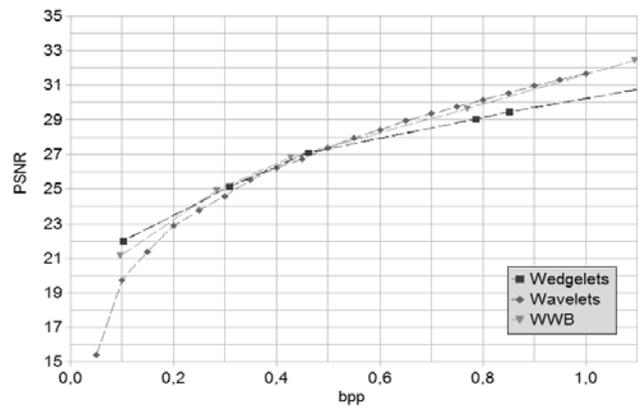


Fig. 10 – Resultados do método WWB (a) imagens naturais (b) desenhos.

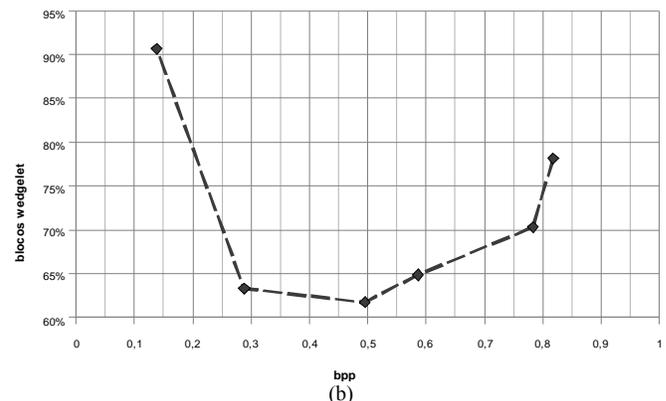
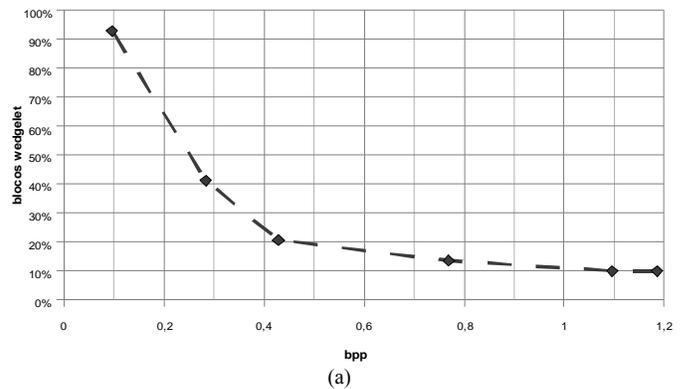


Fig. 11 – Uso de componentes *wedgelet* de acordo com a taxa (a) imagens naturais; (b) desenhos.

As Tabelas 3 e 4 mostram tempos de processamento. Nota-se que a etapa de *wedgelets* aumenta significativamente o tempo do processo de codificação, tendo menor impacto na decodificação. Por ser um processo de busca exaustiva, a complexidade computacional cresce linearmente com o tamanho da imagem com fator de multiplicidade igual ao número de direções consideradas ($O(|\theta| \cdot |d| \cdot n)$). Quando o número de direções consideradas aumenta, esse tempo tende a complexidade exponencial. Como no caso da codificação *wedgelet* isolada, a complexidade do codificador é muito maior do que do decodificador, adequado para aplicações de broadcast. No entanto, mesmo em aplicações móveis, a evolução do poder de processamento dos dispositivos reduz essa desvantagem (última linha de *smartphones* já apresenta capacidade computacional igual ou superior a desktops [30]). A geração da árvore pode ser desenvolvida de maneira mais rápida futuramente, modificando a forma de se realizar a decomposição ou diminuindo o dicionário.

TABELA 3

TEMPO DE PROCESSAMENTO DOS DIFERENTES MÉTODOS (IMAGENS NATURAIS)

Método	Tempo geração da árvore	Tempo de codificação (s)	Tempo de decodificação (s)
<i>Wedgelet</i>	291,19	59,23	4,18
<i>Wavelet</i>	-	3,90	3,60
WWB	306,95	80,88	4,37

TABELA 4

TEMPO DE PROCESSAMENTO DOS DIFERENTES MÉTODOS (DESENHOS)

Método	Tempo geração da árvore	Tempo de codificação (s)	Tempo de decodificação (s)
<i>Wedgelet</i>	181,10	13,02	8,14
<i>Wavelet</i>	-	3,77	3,43
WWB	174,41	24,42	3,62

Nas Figs 12 e 13 são apresentadas comparações visuais das imagens codificadas por *wedgelets*, *wavelets* e o método WWB para a imagem 'cameraman'. Observa-se que o WWB apresenta menos artefatos que o método de *wavelets* e tem melhor qualidade que as *wedgelets* sozinhas para regiões com mais detalhe, como na região da mão.

C. Comparações adicionais com Métodos *Wedgelet*

Métodos *wedgelet* de codificação de imagens são bastante recentes e é difícil encontrar um padrão para comparação. Os artigos apresentam poucos resultados, em geral sem menção a tempos de processamento, e utilizando apenas imagens *cartoons*. Além disso, na maior parte das vezes, os métodos se baseiam apenas em gerar símbolos *wedgelets* não-binários, que invalidam a comparação para compressão de dados.

Em [27] é feito um estudo aprofundado sobre a decomposição *wedgelet*. Em [28] autores disponibilizam um software para decomposição e representação de imagens. Utilizando esse software e convertendo os símbolos *wedgelets* para uma taxa binária fixa, a Fig. 14 mostra uma comparação entre o método WWB e o método de [27]. O WWB consegue valores de PSNR consideravelmente superiores,

principalmente para imagens desenho. A comparação do tempo de processamento é inviável, visto que os algoritmos foram implementados em ambientes completamente distintos.

D. Comparações adicionais com Métodos *wavelet*

Na Fig. 15, é apresentada uma comparação entre o WWB e sistemas de codificação com *wavelet* biortogonal CDF 9/7, considerada estado-da-arte: a codificação SPIHT e o padrão JPEG 2000. Diferente do WWB, esses sistemas aplicam a TWD na imagem toda, não é dividido em blocos, e a taxa de dados é pré-fixada.

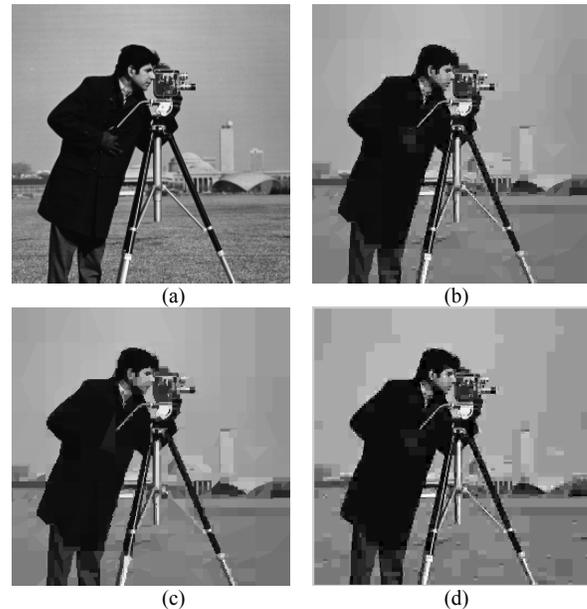


Fig. 12 – Comparação subjetiva da imagem 'cameraman' contra métodos isolados (a) original (b) WWB 0,3251 bpp PSNR = 26,5223 dB (c) *wedgelets* 0,3194 bpp 25,8670 dB (d) *wavelets* 0,3194 bpp PSNR = 25,6961 dB.

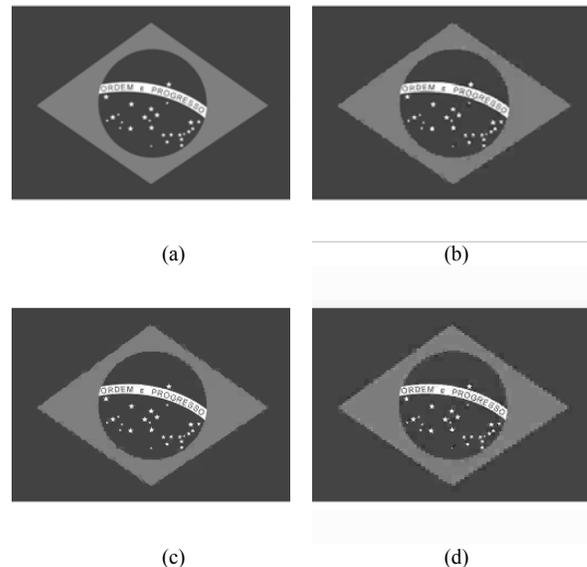


Fig. 13 – Comparação subjetiva da imagem 'Brasil' contra métodos isolados (a) original (b) WWB 0,3431 bpp PSNR = 37,77 dB (c) *wedgelets* 0,3560 bpp 37,6629 dB (d) *wavelets* 0,3500 bpp PSNR = 32,6461 dB.

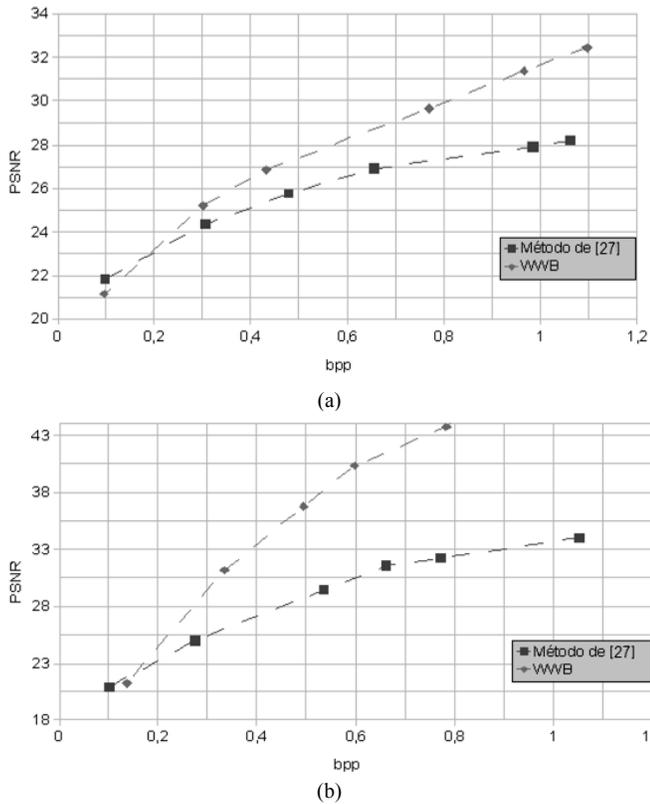


Fig. 14 – Comparação da codificação *wedgelet* de [27] com símbolos binários e o método WWB. (a) imagens naturais (b) desenhos.

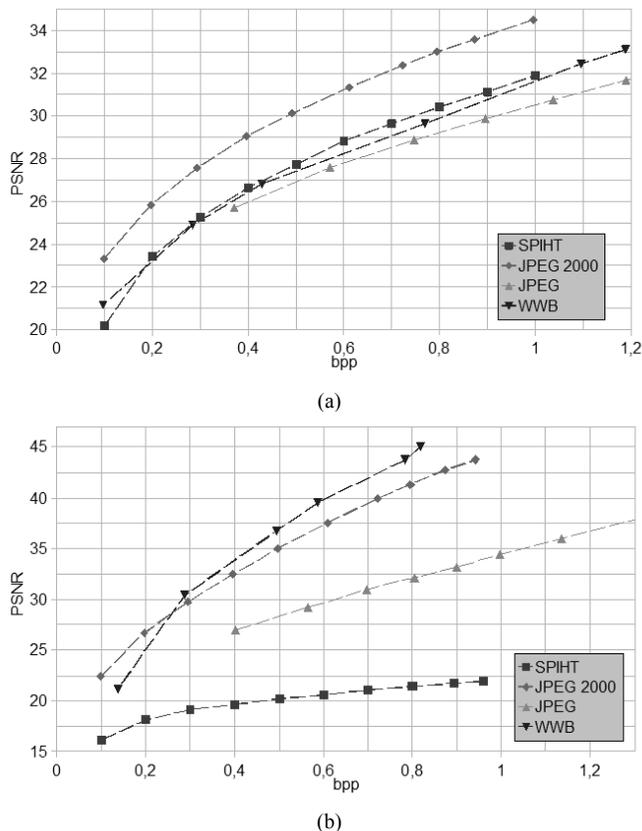


Fig. 15 – Comparação entre WWB e métodos *wavelet*: (a) imagens naturais; (b) desenhos.

Apesar de esses sistemas serem consolidados em anos de otimizações, incluindo codificação entrópica, predição e parametrização de cabeçalho, o método WWB se aproximou bastante, com diferença de 2,5 dB em relação ao JPEG 2000, menos de 0,5 dB em relação ao SPIHT completo e mais de 0,5 dB superior ao JPEG, para as imagens naturais. Para as imagens de desenho, o método WWB é superior aos 3, com ganhos maiores que 1,5 dB. Além disso, embora o WWB tenha PSNR mais baixo a taxas mais altas, otimiza localmente a relação taxa-distorção (comparação subjetiva na Fig. 16). Caso o WWB fosse baseado na codificação preditiva do JPEG 2000 [2], o desempenho poderia ser melhor, mas elevaria a complexidade. A curva para o padrão JPEG (baseado em DCT – *Discrete Cosine Transform* – Transformada Discreta de Cossenos) foi incluída, apenas para fins de comparação. Novamente, a comparação do tempo de processamento é inviável devido aos ambientes distintos em que os algoritmos foram implementados.

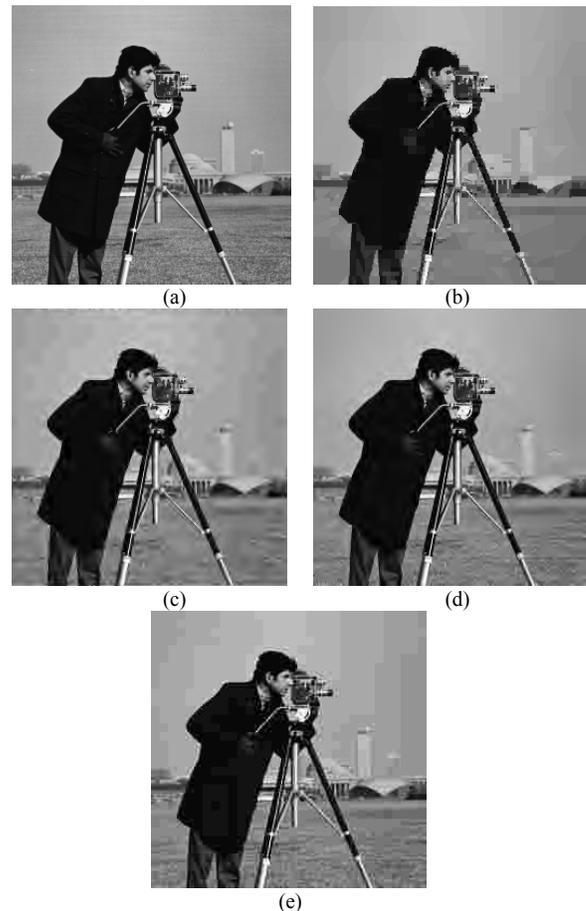


Fig. 16 – Comparação subjetiva da imagem ‘cameraman’ com métodos *wavelet* (a) original (b) WWB 0,3251 bpp PSNR = 26,5223 dB (c) SPIHT 0,3431 bpp PSNR = 27,0453 dB (d) JPEG 2000 0,3479 bpp PSNR = 29,1556 dB (e) JPEG 0,3347 bpp PSNR = 26,4713 dB.

E. Comparações com outros Métodos Wedgelet-wavelet

Wakin et al. propuseram métodos *wedgelet-wavelet* [16][17] (seção IV) com resultados excelentes, mas com poucas curvas para permitir comparação completa. Considerando apenas a imagem 'cameraman', as Figs 17 e 18 mostram comparações entre WWB e dois métodos de Wakin et al. Não é possível obter a taxa exata para comparação devido ao processo de escolha baseado em otimização taxa-distorção. Na Fig. 17, nota-se o desempenho em PSNR muito próximo e a imagem reconstruída pelo WWB apresenta menor *ringing* nos contornos, embora regiões tenham sido distorcidas.

Na Fig. 18.b, o *ringing* é notável embora o desempenho em PSNR seja superior. A imagem reconstruída na Fig. 18.a, embora não tenha *ringing*, tem a aparência muito *cartoonizada*. Como o método de [17] é baseado na SFQ [25], muito superior ao SPIHT, já era esperado que seu desempenho fosse superior ao WWB. Esse ganho tem como contrapartida grande peso computacional para a quantização adaptativa do SFQ e o método de escolha dos *wedgeprints*, embora não se disponha de dados numéricos no artigo de Wakin et al [17].

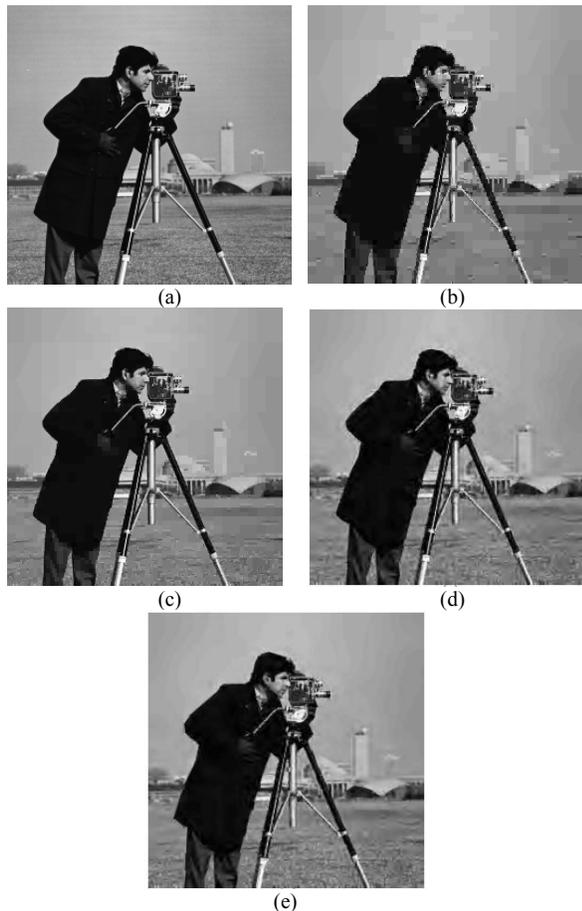


Fig. 17 – Comparação subjetiva com outros métodos *wedgelet-wavelet* (a)original (b) WWB 0,3522 bpp PSNR = 26,6524 dB (c) WWB 0,6278 bpp PSNR = 30,3371dB (d) método de [16] a 0,4 bpp PSNR = 28,48 dB (e) método de [16] com filtro posterior a 0,4 bpp PSNR = 27,35 dB.

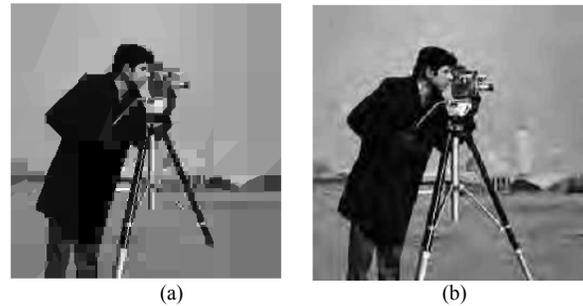


Fig. 18 – Comparação subjetiva com outros métodos *wedgelet-wavelet* (a) WWB 0,1519 bpp PSNR = 23,0640 dB (b) método apresentado em [17] (W-SFQ) 0,146 bpp PSNR = 25,94 dB.

VII. CONCLUSÕES

O presente artigo apresentou resultados experimentais da codificação *wedgelet-wavelet* proposta, denominada WWB, comparativos a outros métodos. Os resultados mostraram que a integração da TWD melhora o desempenho obtido apenas com *wedgelets* na maioria dos casos. Em comparação com métodos baseados somente em TWD, feita sem divisão por blocos, o WWB teve o desempenho esperado (ganho em PSNR nas baixas taxas de dados e uma perda para as altas taxas). A maior vantagem do WWB reside, no entanto, na melhoria local de acordo com a classificação dos blocos em região *cartoon* ou textura, como pode ser constatado nas comparações subjetivas. Para desenhos o ganho de desempenho é ainda maior do que para imagens naturais, um fato que já era esperado considerando que nessas imagens a geometria é mais importante.

Constatou-se que o processo computacionalmente mais custoso é a projeção da árvore *wedgelet* inicial e que esse processo depende diretamente do tamanho do dicionário *wedgelet* considerado. A vantagem é que essa etapa só é realizada pelo codificador, tendo o decodificador um processamento mais simples. Embora não exista uma literatura sobre codificação *wedgelets* tão vasta e principalmente com resultados comparativos detalhados, foram mostradas comparações adicionais com o método *wedgelet* de [27] e com os métodos híbridos de Wakin et al [16][17].

Com um estudo mais aprofundado sobre como esses métodos funcionam e principalmente como codificam os componentes *wedgelet*, pode-se introduzir novas melhorias na codificação *wedgelet-wavelet*.

AGRADECIMENTOS

Agradecimentos à CAPES (Fundação de Aperfeiçoamento de Pessoal de Nível Superior) e à FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo).

REFERÊNCIAS

- [1] Leite, R. B.; Iano, Y.; Silva, A. L. M. C. S.; Silva, F. S.; Aplicação de Dicionários Geométricos à Compressão de Imagens, Revista Ciência e Tecnologia, 2011;
- [2] ISO/IEC 15444 Information technology - JPEG 2000 image coding system, 2004;
- [3] ISO/IEC 14496-2 Information technology -- Coding of audio-visual objects -- Part 2: Visual, Motion Pictures Experts Group, 1999;

- [4] T.F. Chan, H.M. Zhou, Adaptive ENO-Wavelet Transforms for Discontinuous Functions, *12th International Conference on Domain Decomposition Methods*, 2001;
- [5] Velisavljevic, V., Beferull-Lozano, B., Vetterli, M., Space-Frequency Quantization for Image Compression With Directionlets, *IEEE Transactions on Image Processing*, 2007;
- [6] Candès, E. J., Donoho, D. L., *Curvelets: A surprisingly effective nonadaptive representation for objects with edges, Curve and Surface fitting*, A. Cohen, C. Rabut and L. L. Schumaker (Eds.), 1999;
- [7] Le Pennec, E., Mallat, S., *Bandelet Image Approximation and Compression, SIAM Multiscale Model. Simul.*, 4(3):992-1039, 2005;
- [8] Mallat, S., Peyré, G., Orthogonal *bandelet*s for geometric image approximation, *Comm. Pure Appl. Math.*, 61(9):1173-1212, 2008;
- [9] Le Pennec, E., Mallat, S., Sparse geometric image representations with *bandelet*s, *IEEE Trans. Image Process.*, 14(4):423-438, 2005;
- [10] D. L. Donoho, *Wedgelets: nearly minimax estimation of edges*, *Annals of Stat.*, vol. 27, pp. 859-897, 1999;
- [11] Lisowska, A., Extended *Wedgelets* — Geometrical *Wavelets* in Efficient Image Coding, *Machine Graphics & Vision International Journal*, 2004;
- [12] Lisowska, A., Second Order *Wedgelets* in Image Coding, EUROCON - The International Conference on Computer as a Tool, 2007;
- [13] Piyanut Pongpiyaipoon, Development of Efficient Algorithm for Geometrical Representation based on *Arcllet* Decomposition, *Master's Thesis, Technische Universität München, Computational Science and Engineering*, 2005;
- [14] Willett, R. M., Nowak, R. D., *Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging, IEEE Transactions on Medical Imaging*, 2003;
- [15] Chandrasekaran, V., Wakin, M. B., Baron, D., Baraniuk, R. G., Representation and Compression of multidimensional piecewise functions using *surflets*, *IEEE Transactions on Information Theory*, 2009;
- [16] Wakin, M., Romberg, J., Choi, H., Baraniuk, R., Image Compression using an Efficient Edge *Cartoon* + Texture Model, *Proceedings Of The Data Compression Conference (DCC'02)*, IEEE Computer Science, 2002;
- [17] Wakin, M.B., Romberg, J.K., Choi, H., Baraniuk, R.G., Rate-Distortion Optimized Image Compression Using *Wedgelets*, in *IEEE 2002 International Conference on Image Processing - ICIP*, Rochester, New York, September 2002;
- [18] Aujol, J.F., Aubert, G., Feraud, L.B., Chambolle, A., Image Decomposition into a Bounded Variation Component and an Oscillating Component, *Journal of Mathematical Imaging and Vision* 22: 71-88, 2005, Springer Science + Business Media, Inc.;
- [19] Mallat, S., *A Wavelet Tour of Signal Processing*, Academic Press, 1999;
- [20] Wakin, M., Hyeokho Choi, Baraniuk, R., Romberg, J., *Wedgelets: A Multiscale Geometric Representation for Images, UCLA IPAM Tutorials on Multiscale Geometry and Analysis in High Dimensions*, Los Angeles, California, 2004;
- [21] Baraniuk, R., H. Choi, Romberg, J. Wakin, M., Multiscale Geometric Image Compression using *Wavelets* and *Wedgelets*, Rice University, 2010;
- [22] Radon, J.; Parks, P.C., On the determination of functions from their integral values along certain manifolds, *IEEE Transactions on Medical Imaging* 5 (4): 170-176, 1986;
- [23] Lisowska, A., Moments-Based Fast *Wedgelet* Transform, *Journal of Mathematical Imaging and Vision*, pp. 1-13, 2010;
- [24] Friedrich, F., Demaret, L., Führ, H., Wicker, K., Efficient Moment Computation over Polygonal Domains with an Application to Rapid *Wedgelet* Approximation, *SIAM J. Sci. Comput.* Volume 29, Issue 2, pp. 842-863, 2007;
- [25] Xiong, Z., Ramchandran, K., Orchard, M. T., Space-frequency quantization for *wavelet* image coding, *IEEE Trans. Image Proc.*, vol 6, no 5, pp. 677-693, 1997;
- [26] The Waterloo Fractal and Coding and Analysis Group, <http://links.uwaterloo.ca>;
- [27] Wicker, K., Das Potts Modell zur Segmentation verrauschter Daten, *Diploma Thesis*, TU. München, 2004;
- [28] Wicker, K., Friedrich, F., Demaret, L., Führ, H., Poongpiyaipoon, P., *Wedgelet* decomposition Oberon-based software, <http://www.wedgelet.de>;
- [29] OpenJPEG, open-source JPEG 2000 codec, <http://www.openjpeg.org>;
- [30] Lopes, A., Falar é o de menos, *Revista INFO, Editora Abril*, Março 2011.

Ricardo Barroso Leite Nasceu em Valinhos-SP, 1984. Concluiu o curso de Graduação em Engenharia Elétrica em 2008 e obteve o título de Mestrado em 2011, pela UNICAMP. Atualmente realiza seu doutorado no Laboratório de Comunicações Visuais da mesma universidade. Tem experiência em projetos de comunicações via satélite (ITASAT com apoio do Instituto Tecnológico da Aeronáutica) e Televisão Digital (RNP STB-SCAN e CAPES-RHTVD). Seus interesses incluem codificação de áudio e vídeo, vídeo digital, transmissão digital de sinais, televisão digital e comunicações por satélite. (rleite@decom.fee.unicamp.br)

Ana Lúcia Mendes Cruz Nasceu em São Paulo, 1974. Concluiu o curso de Graduação em Engenharia Elétrica em 1998 e obteve o título de Mestrado em 2001 e de Doutorado em 2005 e Pós-Doutorado em 2008, pela UNICAMP. Atualmente está no programa de pesquisadora colaboradora do Departamento de Comunicações da Faculdade de Engenharia Elétrica e Computação da Unicamp. Seus interesses incluem Processamento, compressão e codificação de Imagem e vídeo digitais, Televisão Digital, Codificação de Canal e Comunicações por satélite. (analucia@decom.fee.unicamp.br)

Fernando Silvestre da Silva Nasceu em São Paulo, 1973. Concluiu o curso de Graduação em Engenharia Elétrica em 1998 e obteve o título de Mestrado em 2001 e de Doutorado em 2005, pela UNICAMP. Atualmente está no programa de pesquisador colaborador no Departamento de Comunicações da Faculdade de Engenharia Elétrica e Computação da Unicamp e leciona no Centro Universitário Salesiano de São Paulo (Unisal). Seus interesses incluem processamento, compressão e codificação de imagem e vídeo digitais, televisão digital, codificação de canal e comunicações por satélite. (silva@decom.fee.unicamp.br)

Yuzo Iano Nasceu em Lins, 1950. Concluiu o curso de Graduação em Engenharia Elétrica em 1972 e obteve o título de Mestrado em 1974 e de Doutorado em 1986, pela UNICAMP. Atualmente é Professor Adjunto no Departamento de Comunicações da Faculdade de Engenharia Elétrica e Computação da Unicamp, responsável pelo Laboratório de Comunicações Visuais e coordenador do Projeto LVC-ITASAT. Já desenvolveu projeto de processamento digital de sinais (áudio e imagem) em conjunto com o CPqD. Seus interesses incluem codificação de áudio e vídeo, vídeo digital, transmissão digital de sinais, codificação de canal, televisão digital e comunicações por satélite. (yuzo@decom.fee.unicamp.br)