

Codificadores de imagem usando planos de bits generalizados

Lara Cristiana R. L. Feio

Eduardo A. B. da Silva

PEE/COPPE/DEL/EE, Universidade Federal do Rio de Janeiro

Cx. P. 68504, Rio de Janeiro, RJ

21945-970, BRASIL

{lclf, eduardo}@lps.ufrj.br

Sumário— Codificadores baseados em transformadas wavelet e usando planos de bits representam o estado da arte em compressão de imagens. Recentemente, este conceito foi generalizado com a introdução de planos de bits vetoriais. Neste artigo, são propostos novos algoritmos de codificação de imagens. Eles são generalizações de codificadores por planos de bits escalares. A generalização é baseada na substituição dos planos de bits escalares por planos de bits vetoriais. Também é feita uma análise comparativa do desempenho dos diversos codificadores usando tanto planos de bits escalares quanto vetoriais. Resultados experimentais mostram que o uso de planos de bits vetoriais em substituição aos escalares pode ser vantajoso sob o ponto de vista taxa \times distorção.

1 INTRODUÇÃO

Entre os métodos de compressão de imagens os que mais se destacam são os que usam transformada wavelet e o conceito de planos de bits. Aqui, os coeficientes da imagem transformada são codificados através de iterações sucessivas, isto é, em cada iteração é codificado um plano de bit dos coeficientes da imagem. Este tipo de codificador tem vindo a apresentar um excelente desempenho na codificação de imagens estáticas, sendo, por isso, parte integrante de alguns padrões, entre eles o JPEG-2000 [1] e o procedimento de codificação de textura do MPEG4 [2]. Os codificadores de planos de bits são geralmente usados associados a uma técnica de quantização escalar, onde cada coeficiente é individualmente decomposto em planos de bits. Contudo, em [3] este conceito foi generalizado para o uso de vetores. Neste caso, cada vetor, composto por vários coeficientes, é decomposto em planos de bits vetoriais.

A quantização de um escalar $-1 \leq c \leq 1$ usando planos de bits pode ser representada por uma

sequência $s\{b_1, b_2 \dots, b_n \dots\}$ tal que:

$$c = s \sum_{i=1}^{\infty} b_i 2^{-i} \quad (1)$$

onde $s \in \{-1, 1\}$ representa o sinal de c e $b_i \in \{0, 1\}$.

Alternativamente, pode-se considerar que s seja sempre igual a 1 e $b_i \in \{-1, 1\}$. Assim sendo vem que:

$$c = \sum_{i=1}^{\infty} b_i 2^{-i} \quad (2)$$

Em aplicações práticas de codificação o somatório não é infinito, mas sim limitado a um valor P , que representa o número de planos de bits, e que se encontra diretamente relacionado com o nível de distorção pretendido para a imagem reconstruída.

Estes codificadores, principalmente os que utilizam transformada wavelet, têm motivado muitos estudos sendo atualmente o estado da arte da codificação em imagens estáticas. Assim, têm vindo a ser desenvolvidos diversos codificadores de alto desempenho com estas características.

O primeiro codificador desse tipo foi o **EZW** (*Embedded Zerotree Wavelet*) [4], e o seu funcionamento baseia-se no uso de árvores de zeros (*zerotrees*) e de um codificador aritmético adaptativo [5].

Seguindo o sucesso do **EZW**, Said e Pearlman [6] propuseram algumas alterações a ele no codificador **SPIHT** (*Set Partitional in Hierarchical Trees*) que apresenta desempenho significativamente superior ao do **EZW**.

Recentemente, foi proposto o codificador **MGE** (*Multigrid Embedding* [7]), que possui um desempenho similar ao do **SPIHT**. Ele também se baseia em planos de bits, mas utiliza uma decomposição em *quadtrees*.

Recente também foi o desenvolvimento de uma representação usando planos de bits vetoriais [3]. Esta representação envolve a decomposição do vetor \mathbf{v} da seguinte forma.

$$\mathbf{v}_{\mathbf{Q}} = \sum_{i=1}^Q \mathbf{u}_{ni} \alpha^{-i} \quad (3)$$

onde \mathbf{u}_{ni} é um vetor unitário que pertence a um dado dicionário C_N .

Pode-se mostrar que se α for escolhido convenientemente, o erro $|\mathbf{v} - \mathbf{v}_{\mathbf{Q}}|$ tende a zero à medida que o número de vetores Q tende a infinito. De [3], os valores de α para os quais isto se verifica são dados por:

$$\frac{1}{2 \cos[\Theta(C_N)]} \leq \alpha < 1, \quad \Theta(C_N) \leq 45^\circ \quad (4)$$

$$\sin[\Theta(C_N)] \leq \alpha < 1, \quad \Theta(C_N) \geq 45^\circ \quad (5)$$

onde $\Theta(C_N)$ é o ângulo máximo de um vector no espaço \mathbb{R}^N e o vector mais próximo representado pelo dicionário C_N .

Uma indagação que surge naturalmente é se a substituição dos planos de bits escalares, nos algoritmos acima descritos, por planos de bits vetoriais pode levar a uma melhoria do seu desempenho.

Um trabalho neste sentido foi apresentado em [8]. Aí foi proposto o algoritmo SAWVQ (*Successive Approximation Wavelet Vector Quantizer*), uma extensão do **EZW** para planos de bits vetoriais. Nele, os coeficientes da transformada wavelet foram divididos em vetores e cada um deles representado usando planos de bits. As árvores de zeros deixaram de ser compostas por coeficientes insignificantes passando a ser árvores de vetores insignificantes. A melhoria conseguida foi de quase 1dB para a codificação da imagem *Lena* 512x512 com 0.5 bits/pixel, o que é um excelente resultado.

Contudo, os resultados apresentados neste trabalho não permitem que se tire uma conclusão acerca da maior ou menor eficiência dos planos de bits vetoriais em relação aos escalares. Isto porque o fato deste usar árvores de zero vetoriais tende a aumentar a eficiência da codificação podendo esta ser a principal responsável pelo aumento do desempenho do SAWVQ.

Neste trabalho serão apresentadas versões usando quantização vetorial dos codificadores **SPIHT** e **MGE**. Espera-se com isto, além da obtenção de codificadores com desempenhos superiores aos já existentes, ganhar um entendimento mais profundo tanto das decomposições usando planos de bit

vetoriais, como do efeito do seu uso em métodos práticos de codificação usando planos de bits.

O restante deste artigo está dividido da seguinte forma: Na seção 2 são descritos os algoritmos escalares bem como a sua adaptação para planos de bits vetoriais. Na seção 3 serão analisadas estatísticas do seu desempenho incluindo o número de bits gasto com a informação de significância e com a transmissão dos coeficientes propriamente ditos. Na seção 4 serão apresentadas conclusões e propostas melhorias futuras.

2 DESCRIÇÃO DOS ALGORITMOS

Em todos os codificadores usando planos de bits, além da decomposição em planos de bits propriamente ditos, é usado o conceito de significância dos coeficientes. Isto é, dado um valor de referência, que corresponde à precisão do plano de bits corrente, um coeficiente é dito significativo se o seu módulo é maior que esse valor de referência. Caso contrário, é dito que o coeficiente é insignificante. Assim, a codificação de cada plano de bits é dividida em duas partes: o mapa de significâncias, que indica quais coeficientes são significantes para um dado valor referência, e os planos de bits propriamente ditos. A principal diferença que existe entre os diversos codificadores por planos de bits diz respeito à forma como este mapa de significância é codificado.

No caso do algoritmo **EZW**, o mapa de significâncias é codificado usando o conceito de árvores de zeros (*zerotrees*). Uma árvore de zeros significa que um dado coeficiente e todos os coeficientes correspondentes nas bandas de mesma orientação e com menor escala são insignificantes. No **EZW** os planos de bits são definidos através de um valor de referência que é inicializado como sendo metade do coeficiente da imagem com maior valor absoluto. No passo dominante, para um dado valor de referência, a transformada wavelet da imagem é varrida e cada coeficiente é classificado como sendo positivo (“+”), negativo (“-”), zero isolado (“Z”) ou árvore de zero (“ZT”) e os respectivos símbolos enviados. Em seguida é aplicado um passo de refinamento onde os coeficientes considerados significantes (positivos ou negativos) no passo anterior são refinados com a adição de mais um plano de bits. No final deste passo o valor de referência é dividido por 2 e o passo dominante é iniciado novamente. A codificação para quando é atingida uma distorção alvo ou uma taxa de bits pretendida. Usualmente os símbolos gerados (“+”, “-”, “Z” ou “ZT”) são codificados usando um codificador aritmético.

Para implementar o SAWVQ tiveram que ser efetuadas algumas modificações no algoritmo **EZW**.

A unidade básica de codificação deixa de ser o coeficiente e passa a ser um vetor de coeficientes. Desta forma, ele será considerado significativo quando o seu módulo for maior que um dado valor de referência, que é inicializado como sendo o maior módulo de todos os vetores de coeficientes da imagem. Neste caso, uma árvore de zeros acontece quando um vetor e todos os vetores correspondentes nas bandas de mesma orientação e com menor escala são insignificantes. No passo dominante, para um dado valor de referência, um vetor é classificado de zero (“Z”), árvore de zero (“ZT”) ou significativo, sendo neste último caso também enviado o índice do vetor no dicionário C_N , correspondente ao plano de bits vetorial, (como pode ser visto na equação 3). No passo de refinamento é adicionado um plano de bits vetorial àqueles vetores que já tinham sido considerados significantes em iterações anteriores. No final deste passo, o valor de referência é multiplicado por α (como se vê na equação 3). É usado um codificador aritmético para fazer a codificação tanto da informação de significância como dos índices dos vetores correspondentes no dicionário C_N .

Em [9] foi proposta uma melhoria do algoritmo **EZW**, dando origem ao aqui denominado **EZW/C**, onde as árvores de zeros foram substituídas por uma codificação dependente de contextos da informação de significância. Nele, o contexto do codificador aritmético a ser usado é chaveado levando em consideração os seguintes parâmetros: passo corrente (dominante ou refinamento), escala de decomposição atual, coeficientes vizinhos e coeficiente correspondente na banda imediatamente superior (coeficiente “pai”). Para o caso de uma wavelet de 6 níveis são usados 102 contextos distintos. Desta forma, de cada vez apenas são transmitidos 2 símbolos.

A extensão deste algoritmo usando quantização vetorial (**EZWVQ/C**) que é apresentada na seção 3 foi feita de forma trivial. A única diferença é que a significância passou a ser de vetores de coeficientes, mas os contextos continuam a ser rigorosamente os mesmos. A informação de sinal e refino consiste dos índices dos vetores no dicionário C_N .

A principal diferença do **SPIHT** para o **EZW** é que ele assume, inicialmente, que todos os coeficientes são insignificantes. Nos passos dominantes são transmitidos bits que indicam de forma eficiente, com a ajuda de uma complexa estrutura de listas, quais os coeficientes que passaram a ser significantes para o plano de bits correspondente. A grosso modo, as listas são uma forma sofisticada de implementar árvores de zeros. Nos passos de refino, os planos de bits propriamente ditos são en-

viados (para maiores detalhes deste algoritmo ver [6]). A eficiência deste algoritmo é tal que possui um desempenho superior ao **EZW** mesmo sem nenhuma codificação aritmética dos símbolos. Com codificação aritmética pode-se obter ganhos de até 0.5dB para a imagem *Lena* 512x512.

Para a extensão vetorial do **SPIHT**, denominada **SPIHTVQ**, tem-se que no passo dominante a significância dos coeficientes é substituída pela significância dos vetores. No passo de refino, são transmitidos os índices dos vetores correspondentes no dicionário C_N para o plano de bits onde se está.

No codificador **MGE** a localização dos vetores significantes é determinada segundo um algoritmo *quadtree*. Para cada valor de referência somente é enviado um bit de cada um dos coeficientes que são significantes. Os demais bits são enviados nas iterações seguintes.

Na extensão vetorial do **MGE**, aqui chamada de **MGEVQ** a localização os vetores significantes é também determinada segundo um algoritmo *quadtree* e os índices correspondentes aos planos de bits vetoriais são transmitidos.

3 ANÁLISE DE RESULTADOS

Nesta seção é feita uma análise de desempenho dos algoritmos apresentados na seção 2. Espera-se, com esta análise, obter um entendimento mais profundo acerca do comportamento de cada um destes quando implementados tanto usando planos de bits escalares quanto vetoriais. Nela, são comparados o número de bits que são gastos no envio da informação de significância com aqueles que são gastos com o envio dos coeficientes propriamente ditos. É igualmente analisada a relação sinal-ruído (PSNR) obtida para os diferentes tipos de algoritmos.

Para comparar o desempenho dos diversos algoritmos foram utilizadas, como referências, as imagens *Lena*, *Barbara* e *Boats* com dimensão 512x512. Elas foram reconstruídas em 3 diferentes taxas: 0,2, 0,5 e 1 bit/pixel. A transformada wavelet foi baseada no banco de filtros biortogonal F definido em [10], com 6 níveis de decomposição.

Para os algoritmos **SPIHT** e **MGE** é feita a análise dos seus desempenhos com e sem o uso do codificador aritmético adaptativo [5], tanto no caso escalar quanto no vetorial.

Na implementação usando planos de bits vetoriais foram empregados dicionários com diferentes dimensões na codificação dos vetores: em dimensão 4, foi usada a 1ª camada do reticulado D4; em dimensão 8, foi usada a 1ª camada do reticulado E8; em dimensão 16, foi usada a 1ª camada do reticulado $\lambda 16$ (para mais informação sobre estes dicionários ver [11]).

onários ver [11]). Para cada um destes dicionários e para cada uma das imagens foram utilizados os fatores de escala apresentados na Tabela 1 (de notar que estes fatores de escala são aqueles que permitem obter uma maior relação sinal-ruído para cada um dos casos):

	Lena 512x512	Barbara 512x512	Boats 512x512
D4	$\alpha = 0,58$	$\alpha = 0,60$	$\alpha = 0,58$
E8	$\alpha = 0,61$	$\alpha = 0,61$	$\alpha = 0,59$
$\lambda 16$	$\alpha = 0,63$	$\alpha = 0,62$	$\alpha = 0,64$

Tabela 1: Fatores de escala por dicionário e por imagem.

Observando o comportamento dos vários algoritmos com planos de bits escalares, apresentados na Figura 1(a), pode-se concluir que, para todos os casos, o que mais pesa neste tipo de quantização é o envio dos bits correspondentes à informação de significância. Na tabela 2 são apresentadas as PSNR para todos os algoritmos e pode-se observar que, de forma geral, o algoritmo **SPIHT** apresenta melhor desempenho.

Na Figura 1(b) são comparados diferentes algoritmos para a imagem *Lena* 512x 512 codificada com 0.5 bits/pixel e usando o reticulado $\lambda 16$ na codificação dos vetores. Nesta figura pode-se observar que todos os algoritmos têm um comportamento muito similar, ou seja, todos gastam poucos bits com a codificação da informação de significância e muitos bits com a informação referente aos planos de bits. Isto pode igualmente ser visto através da tabela 3, que apresenta a PSNR obtida na reconstrução, para algumas imagens e taxas.

Em seguida, são comparados, individualmente, cada um dos codificadores implementados usando planos de bits escalares com a versão que usa planos de bits vetoriais. Esta versão é apresentada usando dicionários de diferentes dimensões na codificação dos vetores.

Nas Figuras 1(c)–(f) pode-se observar que, no caso escalar a maior parte dos bits é gasta com a informação de significância. No caso vetorial isto inverte-se completamente, sendo muito maior o número de bits gastos com a informação referente aos planos de bits de vetores de coeficientes.

Uma outra coisa que pode ser observada é que, à medida que aumenta a dimensão do dicionário usado para a codificação dos vetores, maior é a quantidade de informação referente aos planos de bits e menor aquela que diz respeito à significância. Esta diminuição do número de bits gastos com a informação de significância é justificada pelo fato de apenas se ter uma significância vetorial, ou seja, se

um símbolo indica que um vetor é significativo, este mesmo símbolo vai indicar a significância de um número de coeficientes igual à dimensão do vetor. Assim sendo, no caso vetorial tem-se uma diminuição do número de bits gastos com a informação de significância da mesma ordem da dimensão do vetor utilizado para a codificação. Por outro lado, o aumento do número de bits gastos com a codificação dos planos de bits tem duas causas possíveis: A primeira está relacionada com o fato de um vetor significativo poder haver diversas coordenadas insignificantes, fazendo com que os planos de bits vetoriais também carreguem informação que, no caso escalar é puramente de significância. Uma segunda causa para este comportamento poderia ser a menor eficiência intrínseca da representação usando planos de bits vetoriais em relação àquela que usa planos de bits escalares.

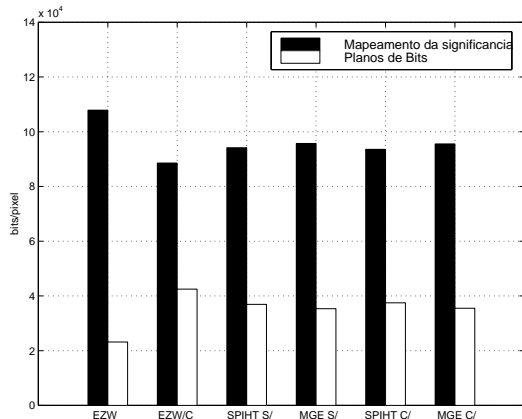
É importante notar que, a principal diferença entre os codificadores usando planos de bits, encontra-se na forma como a informação de significância é transmitida. No caso escalar, isto é confirmado pela Figura 1(a), onde se vê que são gastos muito mais bits com o envio da informação de significância do que com o envio dos planos de bits. Assim, neste tipo de codificador, quanto maior a eficiência na codificação da informação referente à significância, melhor o desempenho obtido. Contudo, a Figura 1(b) mostra que, para o caso do uso de codificadores vetoriais, a quantidade de bits gastos com a significância é muito baixa quando comparada com aquela gasta com os planos de bits. Desta forma, é de se esperar que o desempenho dos diversos codificadores vetoriais seja equivalente, como pode ser confirmado por inspeção da tabela 3.

Finalmente, uma comparação entra as tabelas 2 e 3 mostra que, em geral, usando como dicionário o reticulado $\lambda 16$, os codificadores usando planos de bits vetoriais têm um desempenho superior. Isto é uma indicação de que o uso deste tipo de codificadores pode levar a um aumento do desempenho.

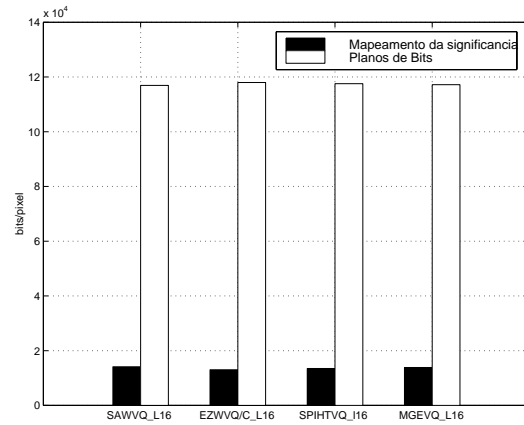
4 CONCLUSÕES

Neste trabalho, foram propostos novos algoritmos de codificação de imagens. Eles são generalizações de algoritmos que utilizam codificação por planos de bits de coeficientes de transformadas wavelet. Esta generalização consiste da substituição dos planos de bits escalares por vetoriais.

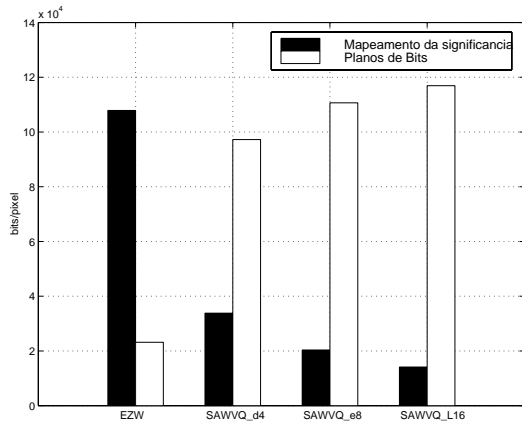
Foi também realizado um estudo comparativo entre os codificadores aqui propostos e as suas versões escalares. Após a análise dos resultados verificou-se que o uso de planos de bits vetoriais leva, em muitos casos, a uma melhoria de desempenho segundo o critério de taxa \times distorção. Além dis-



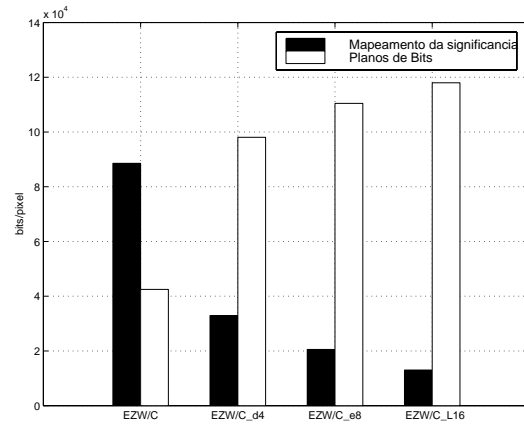
(a)



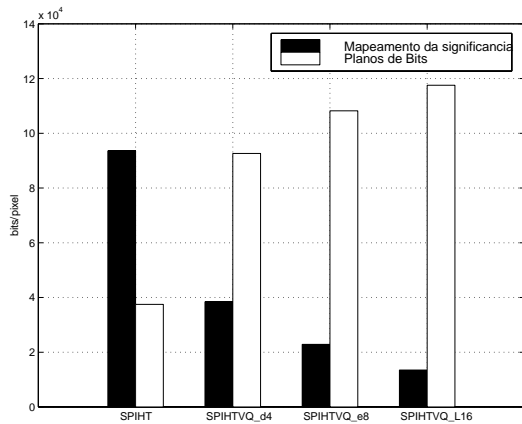
(b)



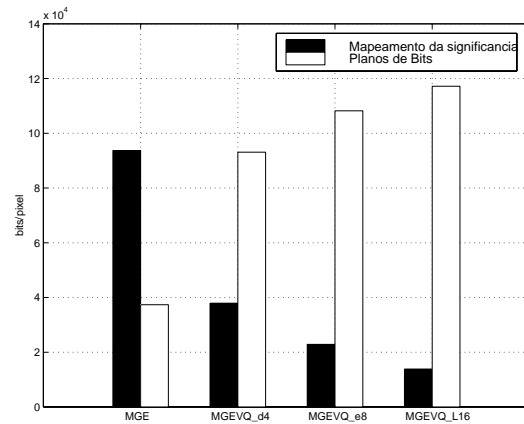
(c)



(d)



(e)



(f)

Figura 1: Comparação dos: (a) codificadores escalares; (b) codificadores vetoriais; (c) codificadores escalares e vetoriais EZW; (d) codificadores escalares e vetoriais EZW/C; (e) codificadores escalares e vetoriais SPIHT; (f) codificadores escalares e vetoriais MGE

	Lena 512x512			Barbara			Boats		
	0,2	0,5	1	0,2	0,5	1	0,2	0,5	1
EZW	32,09	35,91	38,94	25,91	30,49	35,30	29,88	35,49	35,59
EZW/C	32,30	36,85	39,82	25,81	30,45	35,25	30,05	35,47	40,32
SPIHT S/ COD	32,72	36,80	39,94	26,29	31,37	36,62	30,52	35,61	40,18
MGE S/ COD	32,69	36,58	39,60	26,45	31,37	36,68	30,45	35,51	39,79
SPIHT C/ COD	32,93	36,94	40,07	26,37	31,57	36,79	30,71	35,82	40,42
MGE C/ COD	32,81	36,68	39,78	26,48	31,49	36,86	30,63	35,66	40,01

Tabela 2: Comparação entre as PSNR (dB) para os diferentes algoritmos usando planos de bits escalares.

		Lena 512x512			Barbara			Boats		
		0,2	0,5	1	0,2	0,5	1	0,2	0,5	1
SAWVQ	D4	32,52	36,60	39,96	26,56	31,16	35,92	30,81	36,07	39,85
	e8	32,79	36,74	39,92	26,67	31,75	36,16	30,40	35,49	39,89
	λ 16	33,03	37,18	40,25	27,35	32,31	37,11	30,81	35,74	40,42
EZWVQ/C	D4	32,62	36,74	40,08	26,64	31,24	36,14	30,77	35,21	40,13
	e8	32,79	36,84	40,00	26,95	31,90	36,57	30,42	35,56	40,00
	λ 16	33,01	37,19	40,24	27,44	32,38	37,32	30,81	35,79	40,43
SPIHTVQ S/ COD	D4	32,46	36,23	39,36	26,58	30,90	35,93	30,32	34,98	39,27
	e8	32,76	36,63	40,09	27,07	31,88	36,43	30,58	35,58	39,87
	λ 16	32,66	36,76	39,92	26,97	31,89	36,90	30,79	35,51	39,89
SPIHTVQ C/ COD	D4	32,72	36,55	40,02	26,83	31,25	36,48	30,78	35,33	40,15
	e8	32,84	36,77	40,17	27,22	32,05	36,62	30,69	35,70	40,00
	λ 16	33,03	37,18	40,18	27,51	32,42	37,36	30,87	35,82	40,55
MGEVQ S/ COD	D4	32,44	36,42	39,61	26,64	30,95	35,89	30,36	34,96	39,33
	λ 16	32,65	36,79	39,84	27,21	31,92	36,91	30,82	35,54	39,88
MGEVQ C/ COD	D4	32,62	36,56	39,93	26,82	31,61	36,41	30,73	35,25	39,93
	λ 16	32,97	37,11	40,13	27,50	32,43	37,48	30,93	35,84	40,29

Tabela 3: Comparação entre as PSNR (dB) para os diferentes algoritmos usando planos de bits vetoriais.

so, a diferença de desempenho entre os diferentes algoritmos que empregam planos de bits vetoriais é insignificante.

Pode-se então concluir que há ganhos em utilizar implementações usando planos de bits vetoriais e que há indicações de que aumentos significativos de desempenho podem ser obtidos se for aumentada a eficiência de codificação dos planos de bits. Entre as tentativas neste sentido inclui-se, por exemplo, o uso de outros tipos de dicionários.

REFERÊNCIAS

- [1] JPEG2000, "Jpeg2000 verification model 5.3 (technical description)." JPEG2000 Website, JPEG2000 core and VM reflectors. Editor: Charilaos Christopoulos.
- [2] ISO/IEC JTC1/SC29/WG11, "MPEG-4 video verification model version 8.0," July 1997.
- [3] E. A. B. da SILVA and M. Craizer, "Generalized bit-planes for embedded codes," in *1998 IEEE International Conference on Image Processing*, (Chicago, Illinois), October 1998.
- [4] J.M.Shapiro, "Embedded image coding using zerotrees os wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
- [5] I. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, vol. 30, pp. 520–540, June 1987.
- [6] A. Said and W. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," in *1996 IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, (Chicago, Illinois), June 1996.
- [7] T.-H. Lan and A. H. Tewfik, "Multigrid embedding (mge) image coding," in *ICIP*, (Kobe), 1999.
- [8] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, vol. 5, pp. 299–310, February 1996.
- [9] V. R. Algazi and J. Robert R. Estes, "Analysis based coding of image transform and subband coefficients," in *Proceeding SPIE*, pp. 11–21, 1995.
- [10] E. A. B. da Silva and M. Ghanbari, "On the performance of linear phase wavelet transforms in low bit rate image coding," *IEEE Transactions on Image Processing*, vol. 5, pp. 689–704, May 1996.
- [11] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.