

CODIFICAÇÃO DE IMAGENS POR TRANSFORMADAS USANDO UMA TABELA DE ALOCAÇÃO DE BITS ESTENDIDA

Olavo Gomes Pereira¹ e Evaldo Gonçalves Pelaes²

Departamento de Engenharia Elétrica da Universidade Federal do Pará^{1,2}
Caixa Postal 8619, CEP 66.075-900, Belém PA - Brasil

¹email: olavogp@cesupa.br

²email: pelaes@ufpa.br

Sumário - Este artigo apresenta um método de codificação de imagens, que utiliza uma tabela de alocação de bits estendida. Normalmente as tabelas de alocação de bits tem as mesmas dimensões do bloco usado para calcular a transformada. Neste método a tabela de alocação de bits tem dimensões maiores, isto é, para blocos de tamanho 8x8, essa tem tamanho 16x16, pois ela é determinada levando-se em conta um bloco de 8x8 e seus vizinhos. Este procedimento não introduz um grande overhead, uma vez que, devido a semelhança entre as 4 tabelas de 8x8, apenas uma tabela de tamanho 8x8 e as diferenças entre esta tabela e as demais é armazenada. Este método foi aplicado na imagem Lena(512x512), usando as transformadas DCT e LOT, com quantização escalar de Lloid-Max, com taxas que variam de 1 a 8 bpp, sem codificação de entropia. Os resultados foram comparados com os obtidos para imagens codificadas usando tabelas de alocação fixa (8x8), e apresentaram melhorias de até 3,61 dB para a LOT.

Summary - This paper presents a method of images coding, that uses an extended bit allocation table. Usually, the bit allocation tables have the same dimensions of the block used to calculate her transformed. In this method, the table of bit allocation has larger dimensions, that is, for blocks of size 8x8, this table has size 16x16, because it's certain being taken into account a block of 8x8 and your neighbors. This procedure doesn't introduce a great overhead, once, due to likeness among the 4 tables of 8x8, just a table of size 8x8 and the differences between this table and the others are stored. This method was applied in the image Lena(512x512), using them transformed DCT and LOT, with scalar quantization of Lloid-Max, with rates that vary from 1 to 8 bpp, without entropy code. The results were compared with obtained them for codified images using allocation tables it fastens (8x8), and they presented improvements of until 3,61 dB for LOT.

I. INTRODUÇÃO

A codificação digital de imagens é um procedimento de representação de uma imagem como uma seqüência de bits para armazená-la ou transmiti-la através de um canal, com o objetivo principal de minimizar o número de bits necessários para representar a imagem com um certo critério de fidelidade, preservando os níveis de qualidade e inteligibilidade necessária para uma dada aplicação [1, 8]. Isso se deve principalmente ao custo de armazenamento ou transmissão, que crescerá inevitavelmente com o aumento do número de bits. Na transmissão, a principal meta é reduzir a largura da banda

necessária para transmitir imagens digitalizadas, cujas aplicações incluem, televisão digital, televisão de alta definição (HDTV - High Definition Television), teleconferência, fac-símile, redes de computadores, etc. Em áreas que envolvem armazenamento de imagens, o objetivo da aplicação de codificação está relacionado principalmente com a redução do espaço de sinais ocupado por bancos de dados que guardam imagens digitalizadas, tais como imagens médicas, acervos históricos de bibliotecas, imagens de programas espaciais, imagens obtidas por satélite, etc. Os níveis de qualidade e inteligibilidade dependem da aplicação. Portanto na codificação de imagens pode-se reduzir o número de bits para representá-las sem comprometer a qualidade da imagem reconstruída. A figura 1 mostra o diagrama de blocos de um sistema típico de codificação de imagens por transformadas. Como é mostrada na figura 1, a imagem original é primeiramente transformada para um domínio mais apropriado, a fim de se explorar suas características, tal como a correlação entre os pixels. Esse procedimento é realizado através de transformadas conhecidas, tais como transformada co-seno discreta (DCT), transformada seno discreta com rotação de eixos (DSTr), lapped orthogonal transform (LOT), etc [1, 3, 6, 8]. Em geral, a transformada é calculada em blocos, concentrando grande parte de energia em alguns coeficientes transformados, o que significa que se pode recuperar uma imagem com uma boa qualidade, a partir de um número reduzido de coeficientes.



Figura 1. Diagrama de blocos de um codificador de imagens por transformadas.

O segundo bloco do diagrama da figura 1 realiza a quantização, que é um das operações básicas da codificação. A transformada extrai as redundâncias existentes entre os pixels vizinhos dentro de um bloco, gerando coeficientes transformados com diferentes variâncias. De modo que se possa quantizar os coeficientes com menor variância com um número menor de bits, que é o ponto chave da codificação por transformadas. Com base nessas informações e com uma taxa de codificação previamente estabelecida, uma tabela de alocação de bits é gerada e os coeficientes transformados são quantizados de acordo com essa tabela. A quantização dos coeficientes pode ser feita usando-se quantização escalar, onde cada coeficiente transformado é independentemente representado pelo número de bits de acordo com a tabela de alocação.

O terceiro bloco do codificador de imagens da figura 1 faz o mapeamento do conjunto de coeficientes transformados e quantizados, a um conjunto de palavras-código de tamanho fixo ou variável. Quando é utilizada a codificação de entropia, o mapeamento dos coeficientes quantizados é feito utilizando-se, por exemplo, códigos de Huffman, Lempel-Ziv, Aritméticos, etc. Nesse caso, o número médio de bits por pixel, necessários para representar os coeficientes da imagem transformada, é reduzido e se aproxima da entropia. Para reconstituir a imagem, é realizado o processo inverso, isto é, os coeficientes quantizados são reconstituídos a partir da seqüência de bits e a transformada inversa é aplicada sobre esses coeficientes, obtendo-se a imagem reconstruída, que é uma réplica da original e apresentando portanto algum erro devido a quantização e a codificação.

Na codificação de imagens por transformadas, as imagens são processadas em blocos. Devido a esse processamento independente, alguns erros de codificação produzem descontinuidades entre os blocos da imagem reconstruída, uma vez que as últimas amostras de um bloco não casam perfeitamente com as primeiras amostras dos blocos adjacentes. Esses erros aparecerão como efeitos de bloqueamento (ou efeito de blocos) visível na imagem reconstruída. Com a finalidade de reduzir o efeito de bloqueamento, vários trabalhos foram desenvolvidos, sendo que o mais importante foi desenvolvido por Malvar, a LOT ("Lapped Orthogonal Transform") [1, 2,]. A LOT utiliza o procedimento de sobreposição de blocos para minimizar o efeito de bloqueamento.

II. ALOCAÇÃO DE BITS E QUANTIZAÇÃO DOS COEFICIENTES TRANSFORMADOS

A quantização é uma das operações básicas da codificação de imagens. A imagem é dividida em blocos e a aplicação da transformada extrai as redundâncias existentes entre os pixels vizinhos dentro de cada bloco, de tamanho previamente escolhido, gerando coeficientes transformados com diferentes variâncias. Esses coeficientes são quantizados, de modo que, aqueles que tem menor variância serão quantizados com menor número bits e os coeficientes com maior variância serão quantizados com maior número de bits. Esse é um dos pontos básicos para se obter um bom desempenho na codificação de imagens. Os procedimentos para se alocar a taxa de bits, para cada coeficiente de um bloco transformado será descrito a seguir.

De posse dos coeficientes transformados de uma imagem de tamanho $N \times N$, dividida em blocos de tamanho $M \times M$, agrupa-se os coeficientes na mesma posição, obtendo-se um total de M^2 vetores, $c_v(i,j)$, $i, j = 0, 1, 2, \dots, M-1$, de comprimento $n = (N/M) \times (N/M) = N^2/M^2$. Em seguida determina-se as variâncias $\sigma_{i,j}^2$, $i, j = 0, 1, 2, \dots, M-1$, para cada um dos vetores, que corresponde à variância para cada coeficiente transformado $c(i,j)$ do bloco. A figura 2 ilustra o procedimento de agrupamento desses vetores.

Devido à propriedade de compactação de energia, alguns coeficientes transformados retêm a maior parte da energia; certamente que esses coeficientes devem ser quantizados com uma taxa em bit por pixel (bpp), maior que os outros coeficientes que carregam uma menor parcela de energia. A variância de cada coeficiente serve, então como parâmetro para se alocar as taxas para cada coeficiente. O problema agora é o seguinte: dada uma taxa fixa de R bpp como alocar o número médio de bits para cada coeficiente?

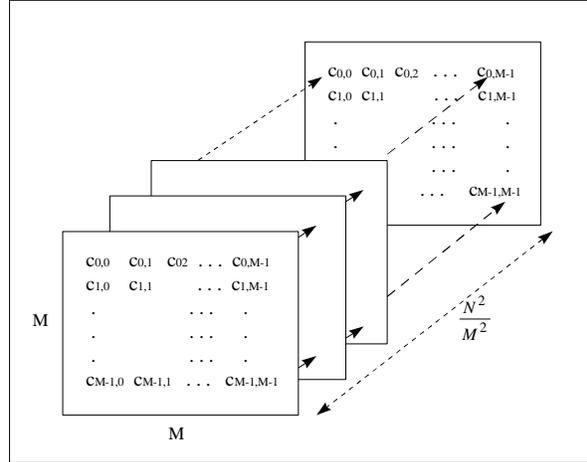


Figura 2. Procedimento de agrupamento dos coeficientes transformados.

Dado um bloco com $M^2 = M \times M$ coeficientes transformados e uma taxa média de R bpp, tem-se um total de $M^2 R$ bits por bloco. Deve-se então determinar o número de bits $B(i,j)$, $i, j = 0, 1, 2, \dots, M-1$ para cada coeficiente transformado de modo que satisfaça:

$$\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} B(i, j) = M^2 R \quad (1)$$

A estratégia tipicamente ótima atribui uma maior taxa aos coeficientes transformados com maior variância de modo que o erro médio quadrático seja minimizado. Isso é um problema de otimização cuja solução é conhecida como regra da log-variância [1, 9]. Assim o número de bits $B(i,j)$, $i, j = 0, 1, 2, \dots, M-1$, para os coeficientes transformados é dado por:

$$B(i, j) = \alpha + \frac{1}{2} \log_2 \frac{\sigma_{i,j}^2}{\sigma_g^2}, \quad i, j = 0, 1, 2, \dots, M-1 \quad (2)$$

onde $\sigma_{i,j}^2$ é a variância do coeficiente transformado $c(i,j)$, $i, j = 0, 1, 2, \dots, M-1$, e σ_g^2 é a média geométrica das variâncias dos coeficientes transformados, isto é:

$$\sigma_g^2 = \left(\prod_{i=0}^{M-1} \prod_{j=0}^{M-1} B(i, j) \right)^{1/M^2} \quad (3)$$

e α é um multiplicador de Lagrange, tal que satisfaça a equação (2). Para taxas relativamente altas, α pode ser aproximado por R , a taxa média em bit/amostra. Se $B(i,j)$ for negativo para algum coeficiente transformado $c(i,j)$, então $B(i,j) = 0$, isto é, nenhum bit será alocado a esse coeficiente.

Com base na tabela de alocação de bits, é então feita a quantização dos coeficientes. Se a quantização usada é escalar (Lloyd-Max), então cada coeficiente é individualmente quantizado

de acordo com o número de bits alocados para aquela posição, obtendo-se a imagem transformada e quantizada, para uma taxa de codificação média de R bit/amostra. Esses coeficientes transformados e quantizados, são mapeados em um conjunto de palavras-código de tamanho fixo ou variável, para que possam ser transmitidos ou armazenados. Quando é utilizada a codificação de entropia, o conjunto de palavras-código é determinado por um código que extrai as redundâncias, tais como os códigos de Huffman, Lempel-Ziv, Aritméticos, etc, cujo objetivo é reduzir, tanto quanto possível, o número médio de bits por pixel, necessários para se representar a imagem transformada e quantizada.

Para reconstituir a imagem, o processo inverso é realizado, isto é, os coeficientes quantizados são reconstituídos pelos procedimentos inversos de quantização e a transformada inversa é aplicada sobre os coeficientes reconstituídos, obtendo-se uma réplica da imagem original, que certamente apresentará alguma distorção devido aos erros de codificação. Essa distorção é medida, em geral, pela relação sinal-ruído (SNR) e pela relação sinal-ruído de pico (PSNR), entre a imagem original e a imagem reconstituída, que para uma imagem $x(i, j)$ de tamanho $N_1 \times N_2$, são calculadas pela expressões:

$$SNR_{dB} = 10 \log_{10} \left(\frac{\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} x[i, j]^2}{\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (x[i, j] - x_r[i, j])^2} \right) \quad (4)$$

$$PSNR_{dB} = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N_1 N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (x[i, j] - x_r[i, j])^2} \right) \quad (5)$$

onde $x_r[i, j]$ representa a imagem reconstituída.

III. O MÉTODO DE ALOCAÇÃO DE BITS PROPOSTO

Semelhante ao método de alocação apresentado na seção anterior, divide-se inicialmente a imagem de tamanho $N \times N$ em blocos de tamanho $M \times M$ e em seguida aplica-se a transformada sobre os blocos. A diferença principal deste método se dá no processo de quantização, onde normalmente a tabela de alocação de bits é do mesmo tamanho do bloco transformado, aqui essa tabela é expandida para a ordem $K \times M$. Onde K é um inteiro positivo e assume valores no intervalo $[1 \dots N/M]$. Essa modificação trará como resultado uma melhoria no SNR e PSNR, entretanto o custo deste benefício acarretou um aumento de overhead. No método convencional guardava-se uma tabela de alocação de tamanho $M \times M$, aqui será necessário o armazenamento de K^2 tabelas de tamanho $M \times M$. O que corresponde a um overhead $K^2 - 1$ tabelas de tamanho $M \times M$ em relação ao método convencional. A figura 3 mostra uma tabela de alocação de bits, obtida para a imagem Lena512 para a taxa de 0.7188 bits/pixel para $M=8$ e $K=2$. Neste caso a imagem foi transformada por DCT, com blocos de tamanho 8×8 e tabela de alocação de bits foi calculada como se estivesse

trabalhando com blocos de tamanho 16×16 , usando-se a regra da log-variância. Observa-se, no entanto que se tem na realidade 4 tabelas de tamanho 8×8 e as 4 tabelas de tamanho 8×8 tem valores bem próximos para uma dada posição (i, j) . A partir disto foi possível buscar uma solução para minimizar os custos gerados pelo novo método, aplicando-se o seguinte procedimento: primeiro fixa-se uma das K^2 tabelas de tamanho $M \times M$ como padrão, em seguida calcula-se as diferenças entre a tabela padrão e cada uma das tabelas restantes, obtendo-se assim $K^2 - 1$ tabelas residuais.

Aplicando-se este procedimento aos dados da figura 3, obtém-se como resultado a tabela padrão e as residuais conforme podem ser visualizadas na figura 4. Essas tabelas residuais por sua vez possuem poucos valores não nulos, permitindo com isto uma redução no armazenamento do overhead. O overhead, agora consiste da tabela padrão de tamanho $M \times M$ e os valores não nulos com suas respectivas posições i e j .

O overhead total é dado pela soma dos overhead individuais das tabelas residuais, $R(k)$, onde k assume valores no intervalo $[1 \dots K^2 - 1]$. Cada tabela residual $R(k)$ tem $m(k)$ valores não nulos e estes valores podem ser codificados por um código de tamanho $l(k)$. Deste modo tem-se a seguinte expressão:

$$l(k) = \max(\max(R(k))) \cdot \min(\min(R(k))) \cdot 1 \quad (6)$$

O overhead da k -ésima tabela residual $R(k)$, é dado

por:

$$ovh(k) = \lceil \log_2 l(k) + 2 * \log_2 bl \rceil * m(k) \quad (7)$$

Numa forma mais compacta, tem-se:

$$ovh(k) = \lceil \log_2 [l(k) * M^2] \rceil * m(k) \quad (8)$$

O overhead total em relação ao método convencional é expresso por:

$$OVH = \sum_{k=1}^{K^2-1} ovh(k) \quad (9)$$

7	4	3	2	2	1	0	0	6	4	3	2	2	1	0	0
4	3	2	2	1	1	0	0	3	3	2	2	1	1	0	0
2	2	2	1	1	0	0	0	2	2	2	1	1	0	0	0
1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	4	3	2	2	1	0	0	6	4	3	2	2	1	1	0
3	3	2	2	1	1	0	0	3	3	2	2	1	1	0	0
2	2	2	2	1	1	0	0	2	2	2	1	1	1	0	0
1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 3. Tabela de alocação de bits de 16×16 , obtida para a imagem Lena512, com taxa de 0,7188 bpp.

7	4	3	2	2	1	0	0	1	0	0	0	0	0	0	0	0	0
4	3	2	2	1	1	0	0	1	0	0	0	0	0	0	0	0	0
2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	-1	0	0	0	0	0	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4. Tabela padrão(8x8) e 3 as tabelas residuais.

Para o exemplo mostrado na figura 4, tem-se:

k	$m(k)$	Faixa (k)	$\log_2 l(k)$ (bits)	$2*\log_2 M$ (bits)	$ovh(k)$ (bits)
1	4	[-1 0 1]	2	6	32
2	8	[-1 0 1]	2	6	64
3	4	[-1 0 1]	2	6	32
Total					128

Sem a minimização do overhead seriam necessárias 3 tabelas com 64 elementos cada, onde cada elemento ocuparia 3 bits o que perfaz um total de 576 bits, ou 72 bytes contra os 16 bytes obtidos.

IV. RESULTADOS OBTIDOS

Os resultados do método proposto foram alcançados adotando-se a seguinte metodologia: Obteve-se os valores SNR, PSNR e OVH para imagem *Lena512*, a partir das transformadas DCT e LOT utilizando-se quantização escalar de Lloid -Max com os quantizadores uniforme, laplaciano e gaussiano para taxas no intervalo [0...8] bits por pixel. Isto foi aplicado tanto para o método proposto como para o método convencional, identificados respectivamente pelos índice II e I. Em seguida calculou-se as diferenças de SNR e PSNR para uma mesma transformada e mesmo quantizador. Com isto pode-se verificar o efeito do método sobre uma dada transformada e um dado quantizador. A partir daí pode-se eleger o quantizador que melhor responde ao método II. Em uma segunda fase se adotou-se o referido quantizador para verificar diferenças de taxas para uma mesma SNR em uma dada transformada, obtendo-se assim quanto se poderia comprimir a imagem utilizando o método II em relação a método I. A Tabela I mostra os resultados obtidos para a DCT. Note que está assinalado em negrito onde os resultados são mais destacados para o método II

Tabela I. Resultados obtidos para a DCT, para a imagem *Lena512*, quando foram usados os métodos I e II de alocação de bits.

Q	Tx	SnrII	PSnrII	SnrI	PSnrI	Δ Snr	Δ Psnr	Ovh
U	1	26.65	32.31	25.85	31.52	0.80	0.79	160.
N	2	29.87	35.53	28.99	34.65	0.88	0.88	120.
I	3	33.22	38.87	32.26	37.92	0.96	0.95	192.
F	4	37.57	43.23	36.27	41.93	1.30	1.30	160.
O	5	43.02	48.68	41.47	47.13	1.55	1.55	176.
R	6	48.16	53.81	46.79	52.44	1.37	1.37	176.
M	7	51.02	56.67	49.95	55.61	1.07	1.06	28.
E	8	51.63	57.29	50.69	56.35	0.94	0.94	0
t		SnrII	PSnrII	SnrI	PSnrI	Δ Snr	Δ Psnr	Ovh
L	1	27.07	32.72	26.68	32.54	0.39	0.18	160.
A	2	30.94	36.59	30.75	36.41	0.19	0.18	120.
P	3	34.53	40.18	34.37	40.02	0.16	0.16	192.
L	4	37.31	42.97	37.19	42.85	0.12	0.12	160.
A	5	38.90	44.56	38.83	44.48	0.07	0.08	176.
C	6	39.49	45.14	39.44	45.10	0.05	0.04	176.
E	7	39.61	45.27	39.59	45.25	0.02	0.02	28.
E	8	39.61	45.27	39.59	45.25	0.02	0.02	0
t		SnrII	PSnrII	SnrI	PSnrI	Δ Snr	Δ Psnr	Ovh
G	1	25.68	31.33	25.55	31.20	0.13	0.13	160.
A	2	28.45	34.10	28.27	33.63	0.18	0.47	120.
U	3	31.07	36.72	30.97	36.63	0.10	0.09	192.
S	4	33.78	39.44	33.64	39.30	0.14	0.14	160.
S	5	36.80	42.46	36.51	42.16	0.29	0.30	176.
I	6	39.87	45.53	39.66	45.31	0.21	0.22	176.
A	7	42.14	47.80	42.08	47.73	0.06	0.07	28.
N	8	42.67	48.33	42.65	48.31	0.02	0.02	0

A figura 5 mostra o gráfico da taxa x SNR, referentes aos resultados obtidos para a DCT, para a imagem *Lena512*, codificada com taxa de 1 a 8 bpp, usando-se os dois métodos de alocação de bits.

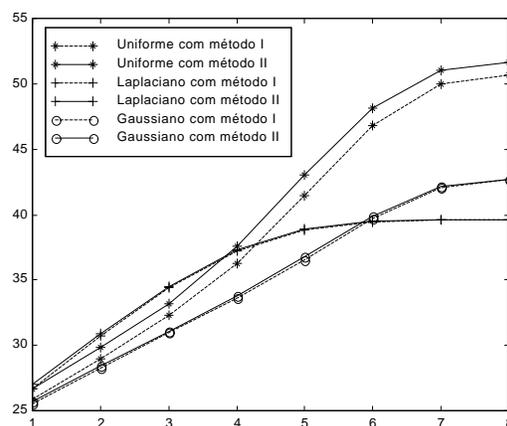


Figura 5. SNR para DCT com os métodos I e II.

A Tabela II mostra os resultados para a Lena512, quando é usada a transformada LOT. Pode-se observar que as vantagens quando se usa o método de alocação proposta é bem mais expressivo para a LOT, chegando-se a uma diferença de até 3,61 dB para uma taxa de 5 bpp. Lembrando que aqui não se

Tabela II. Resultados obtidos para a LOT, para a imagem Lena, quando se usa os métodos I II de alocação de bits.

Q	Tx	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
U	1	27.27	32.93	26.20	31.85	1.07	1.08	128.
N	2	30.25	35.91	29.05	34.71	1.20	1.20	112.
I	3	33.71	39.36	32.08	37.74	1.63	1.62	128.
F	4	38.29	43.95	35.76	41.41	2.53	2.54	96.
O	5	43.88	49.54	40.27	45.93	3.61	3.61	112.
R	6	48.73	54.38	45.42	51.08	3.31	3.30	128.
M	7	51.22	56.88	49.31	54.97	1.91	1.91	24.
E	8	51.73	57.39	50.38	56.04	1.35	1.35	0
Q	Tx	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
L	1	27.80	33.45	27.27	32.93	0.53	0.52	128.
A	2	31.44	37.10	30.81	36.46	0.63	0.64	112.
P	3	34.96	40.62	34.03	39.69	0.93	0.93	128.
L	4	37.58	43.24	36.45	42.10	1.13	1.14	96.
A	5	38.99	44.64	37.81	43.47	1.18	1.17	112.
C	6	39.44	45.09	38.72	44.38	0.72	0.71	128.
E	7	39.52	45.18	39.31	44.97	0.21	0.21	24.
	8	39.52	45.18	39.52	45.18	0	0	0
Q	t	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
G	1	26.38	32.03	25.97	31.62	0.41	0.41	128.
A	2	29.02	34.67	28.54	34.19	0.48	0.48	112.
U	3	31.63	37.28	30.96	36.62	0.67	0.66	128.
S	4	34.27	39.93	33.63	39.30	0.64	0.63	96.
S	5	37.34	43.36	36.66	41.66	1.34	1.34	112.
I	6	40.41	46.06	38.47	44.13	1.94	1.93	128.
A	7	42.29	47.95	40.44	46.10	1.85	1.85	24.
N	8	42.68	48.33	41.23	46.88	1.45	1.45	0
O								

A Tabela II pode ser ilustra através dos gráficos mostrados na figura 6 .

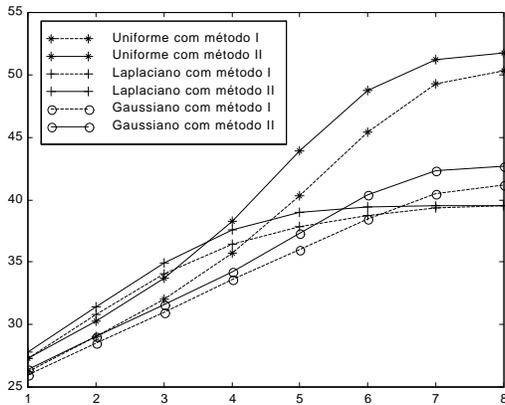


Figura 6. SNR para LOT da imagem Lena512, quando se usa com os métodos I e II de alocação de bits.

Tabela III. Resultados da DSTr, para a imagem Lena512 quando se usou o método I e método II de alocação de bits.

	t	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
D	1	25.58	31.23	25.53	31.17	0.05	0.06	106.
S	2	28.67	34.33	28.21	33.87	0.46	0.46	160.
T	3	31.65	37.31	30.82	36.47	0.83	0.84	208.
R-	4	36.33	41.99	34.99	40.65	1.34	1.34	208.
U	5	42.08	47.74	40.55	46.21	1.53	1.53	208.
N	6	47.60	53.26	46.43	52.08	1.17	1.18	208.
I	7	51.09	56.74	50.67	55.72	0.42	1.02	44.
F	8	51.90	57.56	51.01	56.66	0.89	0.90	0
	t	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
D	1	26.47	32.13	26.58	32.23	-0.11	-0.10	106.
S	2	30.58	36.23	30.63	36.28	-0.05	-0.05	160.
T	3	34.58	40.23	34.49	40.15	0.09	0.08	208.
R-	4	38.32	43.98	38.29	43.94	0.03	0.04	208.
L	5	40.67	46.33	40.64	46.29	0.03	0.04	208.
A	6	41.55	47.21	41.63	47.27	-0.08	-0.06	208.
P	7	41.76	47.41	41.80	47.46	-0.04	-0.05	44.
L	8	41.76	47.42	41.81	47.47	-0.05	-0.05	0
	t	SnrII	PSnrII	SnrI	PSnrI	ΔSnr	ΔPsnr	Ovh
D	1	25.37	31.02	25.43	31.09	-0.06	-0.07	106.
S	2	28.64	34.30	28.63	34.28	0.01	0.02	160.
T	3	30.85	36.50	30.81	36.47	0.04	0.03	208.
R-	4	34.09	39.75	34.09	39.66	0.09	0.09	208.
G	5	37.30	42.95	37.22	42.87	0.08	0.08	208.
A	6	41.17	46.83	41.14	46.80	0.03	0.03	208.
U	7	44.10	49.76	43.96	49.62	0.14	0.14	44.
SS	8	44.82	50.48	44.73	50.39	0.09	0.09	0

A Tabela III pode ser ilustrada através dos gráficos mostrados na figura 7 . Como pode ser observado nos gráficos apresentados, o método de alocação proposto se destaca em relação ao método convencional quando se utilizou o quantizador uniforme independentemente da transformada associada. Também o método proposto teve o seu melhor desempenho em relação ao convencional quando ele foi aplicado sobre os coeficientes transformados da LOT. Nesta foi possível atingir uma diferença de até 3.61 dB em relação ao método convencional., como também observou-se que os maiores valores de SNR e PSNR se concentram na LOT uniforme e Laplaciana, para uma dada taxa. Já no método convencional eles são encontrados dispersos entre as transformadas.

A partir dos resultados anteriores analisou-se para as 3 transformadas com quantizador uniforme, o quanto seria necessário em termos de taxa para o método convencional atingir a mesma SNR para uma dada transformada. O resultados estão ilustrados na Tabela IV.

Na Tabela IV acrescentou-se o tamanho da imagem, em bytes, para a mesma SNR com suas respectivas taxas. O valor ImII corresponde ao tamanho da imagem usando o método II, podendo ser expressa como:

$$ImII = N * N * TaxaII + OVH \quad (10)$$

Onde TaxaII é a taxa no método II para um dado SNR. Já o valor ImI é dado por:

$$ImI = N * N * TaxaI \quad (11)$$

Onde $Taxa_I$ é a taxa no método I para o mesmo SNR dado anteriormente. Com isto pode-se observar que o método II pode proporcionar uma redução de até 21.9% com relação a imagem obtida pelo métodos I. para a mesma SNR.

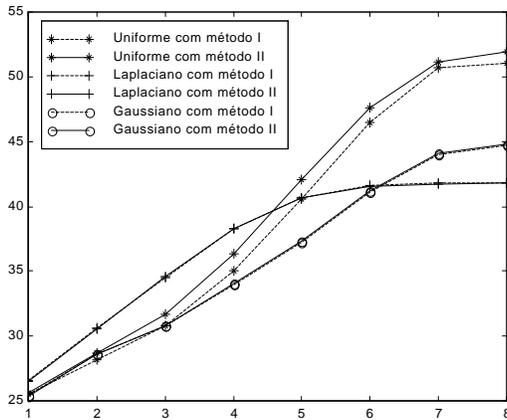


Figura 7. SNR para DSTr da imagem Lena512 com os métodos I e II de alocação de bits.

Tabela IV. Taxas no método convencional (I) necessárias para obter um dado valor de SNR no método proposto (II) para o quantizador uniforme utilizando as transformadas DCT, LOT e DSTr

DCT UNIFORME							
SNR	TaxaII	Taxal	ImI	ImII	ΔImg	$\Delta Im/ImI$	
26.65	1.00	1.13	36.00	32.02	3.98	11.06	
29.87	2.00	2.23	71.50	64.01	7.49	10.47	
33.22	3.00	3.27	104.50	96.02	8.48	8.11	
37.57	4.00	4.27	136.50	128.02	8.48	6.21	
43.02	5.00	5.44	174.00	160.02	13.98	8.03	
48.16	6.00	6.31	202.00	192.02	9.98	4.94	
51.02	7.00	8.00	256.00	224.00	32.00	12.50	
51.63	8.00	8.00	256.00	256.00	0	0	
LOT UNIFORME							
SNR	TaxaII	Taxal	ImI	ImII	ΔImg	$\Delta Im/ImI$	
27.27	1.00	1.28	41.00	32.02	8.98	21.91	
30.25	2.00	2.33	74.50	64.01	10.49	14.07	
33.71	3.00	3.44	110.00	96.02	13.98	12.71	
38.29	4.00	4.53	145.00	128.01	16.99	11.72	
43.88	5.00	5.73	183.50	160.01	23.49	12.80	
48.73	6.00	6.86	219.50	192.02	27.49	12.52	
51.22	7.00	8.00	256.00	224.00	32.00	12.50	
51.73	8.00	8.00	256.00	256.00	0	0	
DSTr UNIFORME							
SNR	TaxaII	Taxal	ImI	ImII	ΔImg	$\Delta Im/ImI$	
25.58	1.00	1.14	36.50	32.01	4.49	12.29	
28.67	2.00	2.34	75.00	64.02	10.98	14.64	
31.65	3.00	3.30	105.50	96.03	9.48	8.98	
36.33	4.00	4.28	137.00	128.03	8.98	6.55	
42.08	5.00	5.26	168.40	160.03	8.38	4.97	
47.60	6.00	6.28	201.00	192.03	8.98	4.47	
51.09	7.00	8.00	256.00	224.01	31.99	12.50	
51.90	8.00	8.00	256.00	256.00	0	0	

V. CONCLUSÃO

O presente trabalho apresentou um método de alocação de bits usado no processo de codificação por transformadas, que consiste em uma pequena variação no processo de alocação de bits que usa a regra da log-variância. Este procedimento foi aplicado aos coeficientes transformados da LOT, DCT e DSTr, oriundos da imagem Lena512 e resultou em melhorias consideráveis com relação ao critério objetivo da SNR. Utilizando-se de um processo comparativo verificou-se que a quantização uniforme apresenta o melhor desempenho para todas as transformadas. Realizando a análise para as transformadas, a LOT é a que absorveu melhor as características do método de alocação, independentemente do tipo de quantização (uniforme, laplaciana ou gaussiana). Isso torna-se evidente quando se verifica que para cada taxa, que varia de 1 a 8 bits, é na LOT que se encontram os maiores valores de SNR. Sendo que no método convencional os maiores valores ficavam distribuídos entre as três transformadas.

Na segunda fase de experimentos se reduziu a investigação à quantização uniforme, e comparou-se para um dado valor de SNR, em uma dada taxa no método proposto, quanto se economizaria em termos de bytes para se obter a mesma qualidade quando se utiliza o método convencional. E se chegou a um obter uma expressiva redução de até 21%.

Os resultados obtidos apontam a perspectiva de investigar os possíveis relacionamentos entre processo de expansão da tabela de alocação de bits e a transformada LOT.

REFERÊNCIAS

1. Malvar, H. S., "Signal Processing with Lapped Transform", Artech House, 1992.
2. Pelaes, E. G. e Yuzo Iano, "Image coding with low Blocking Effects Using Discrete Sine with Axis Rotation", Proc. of XV International Telecommunication Symposium - ITS98, São Paulo - SP, Brazil, August 1998.
3. Pelaes, E. G., and Yuzo Iano, "Image Coding Using Discrete Sine Transform with Axis Rotation", IEEE Transactions on Consumer Electronics, vol. 44, n. 4., nov. 1998.
4. Pelaes, E. G., Yuzo Iano e A. M. L. Araújo, "Interpolação de Imagens por Transformadas com Eliminação de Extrapolações e Reduzido Efeito de Bloqueamento", Anais do XV Simpósio Brasileiro de Telecomunicações, Recife-PE set. 1997.
5. Pelaes, E. G., "Transformada Seno Discreta com Rotação de Eixos Bidimensional (DSTr-2D): Aplicações na Codificação e Interpolação de Imagens para Redução de Efeito de Blocos" Tese de Doutorado, UNICAMP, 1988.
6. Rao, K. R., and P. Yip, "Discrete Cosine Transform", Academic Press, inc, 1990.
7. Max, Joel, "Quantizing for Minimum Distortion", IRE Trans. Inform. Theory, vol. IT-6, pp. Mar. 1960.
8. Lim, Jae S., "Two-Dimensional Signal Processing and Image Processing", Prentice-Hall, 1990

AGRADECIMENTOS

O presente trabalho obteve a colaboração das seguintes entidades: Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Universidade Federal do Pará (UFPA) e Centro de Ensino Superior do Pará (CESUPA).