1000-9825/2001/12(09)1287-08 ©2001 Journal of Software 软 件 学 报 Vol. 12, No. 9

Embedded Image Coding Based on Hierarchical Discrete Cosine Transform*

ZHAO De-bin[1], ZHANG Da-peng[2], GAO Wen[1,3]

[1](*Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin* 150001, *China*);

[2](*Department of Computing, Hong Kong Polytechnic University, Hong Kong, China*);

[3](*Institute of Computing Technology, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: dbzhao@vilab.hit.edu.cn

http://www.hit.edu.cn

Received July 5, 2000; accepted February 27, 2001

Abstract: In this paper, we illustrate that the zerotree quantizer developed originally for wavelet compression can be effectively applied to Discrete Cosine Transform (DCT) in a hierarchical way. In this Hierarchical DCT (HDCT), the input image is partitioned into a number of 8×8 blocks and a first level DCT is used to each of these blocks individually. As DC coefficients of DCT neighboring blocks are highly correlated and particularly pronounced to obtain the compression results at low bit rates, another level DCT is applied to only DC coefficients re-organized as 8×8 blocks. This procedure is repeated until the last step is reached. All the HDCT coefficients within a DCT block are then rearranged into a sub-band structure in which the zerotree quantizer can be employed. The proposed algorithm yields a fully embedded, low-complexity coder with competitive PSNR performance. For example, when compared with the baseline JPEG on the 512×512 standard image "Lena", it gains 0.8 dB~1.7 dB. In order to remove the blocking effects in the reconstructed images, especially at low bit rates, a simple and efficient method based on Sobel operators is also developed. Experimental results show that the proposed deblocking method works well and enhances decoding for decompressed images both objectively and subjectively.

Key words: discrete cosine transform (DCT); image compression; embedded zerotree coding; data compression; data organization

Transform coding has been widely used in many practical image/video compression systems because its quantized coefficients after linear transform can yield better compression results than direct coding of image intensity in the spatial domain. KL transform can be found to be optimal under certain conditions, however, signal-independent sub-optimal approximations like DCT are often used for computational efficiency.

---

* Supported by the National Natural Science Foundation of China under Grant No. 69789301 (国家自然科学基金)

ZHAO De-bin was born 1963. He received his Ph.D. degree in computer science from Harbin Institute of Technology in 1998. Now He is a professor at Department of Computer Science, Harbin Institute of Technology. His current research interests include data compression, image processing and human-machine interface. **ZHANG Da-peng** was born in 1950. He is a professor at Department of Computing, Hong Kong Polytechnic University. His current research interests are automated biometric-based identification, neural systems and applications, pattern recognition & image processing, VLSI system design methodology and parallel computer architecture. **GAO Wen** was born in 1956. He is the professor and doctoral supervisor of Institute of Computing Technology, CAS and Department of Computer Science, Harbin Institute of Technology. His current research areas include multimedia data compression, image processing, computer vision, multimodal interface, artificial intelligence, virtual reality.

In recent years, most of the research activities in image coding have shifted from DCT to wavelet transforms, especially after Shapiro introduced his famous embedded zerotree wavelet coder (EZW)[1] and Said and Pearlman developed an algorithm by set partition in hierarchical trees (SPIHT)[2]. Their methods provide very high compression efficiency in terms of Peak Signal-to-Noise Ratio (PSNR) versus required bits-per-pixel (bpp). But, it should be noted that the high compression efficiency in the rate/distortion sense quantitatively does not correspond to the same degree of compression efficiency in terms of visual quality versus bit rate. This has been pointed out in Ref. [3] by comparing visual results based on wavelet and DCT.

Although wavelets are capable of providing more flexible space-frequency resolution tradeoffs than DCT, DCT still can produce very high compression efficiency when coupled with a zerotree quantizer instead of the traditional methods used in JPEG. Some better results using such techniques have been reported in Refs. [4,5], where EZDCT (Embedded Zerotree DCT coding) with arithmetic coding outperforms any other DCT-based coder published in the literature[4] and STQ (Significance Tree Quantization) without using arithmetic coding is superior to a version of EZDCT which also doesn't use arithmetic coder[5]. We observe that DC coefficients in EZDCT are still highly correlated and particularly pronounced to obtain the compression results at low bit rates. We also notice that a maximum absolute coefficient is outputted first for all DCT coefficients in EZDCT and STQ, actually the maximum absolute coefficients in each block maybe greatly differ from each other, so it is not reasonable to output only one n for all blocks. We present here an embedded zerotree image coder based on HDCT (EZHDCT) with improvements on using DCT in a hierarchical way and an additional step of outputting the magnitude of each block. EZHDCT achieves competitive PSNR performance when comparing to other DCT-based coders, such as the baseline JPEG[6], EZDCT and STQ. Another component of the work presented here relates to deblocking of DCT artifacts. As block-based image coding schemes always cause annoying blocking effects in the decoded images especially at low bit rates, a simple and efficient deblocking method is proposed to deal with this problem. The parameter of proposed deblocking method can be adaptively adjusted according to the complexity of the input image and the specified bit rate.

1 Embedded Image Coding Method

1.1 HDCT structure

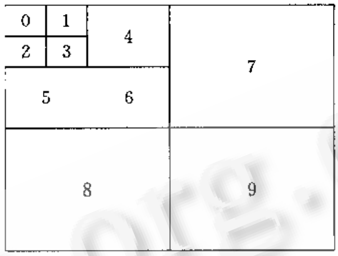
An input image ($M\times N$) is first divided into $n\times n$ blocks, where $n=2^L$, $L>0$. Each block is then transformed by a first level DCT. Each $n\times n$ DCT block can be treated as a L-scale tree of coefficients labeled from 0 to $n\times n-1$. Figure 1(a) is an example of 8×8 DCT ($n=8, L=3$) coefficients labeled from 0 to 63. The tree structure for 8×8 DCT coefficients in Fig. 1(a) can be viewed as a 64-subband decomposition. Because DC coefficients of neighboring blocks are highly correlated and particularly pronounced at low bit rates, a second level DCT transform is applied. This procedure is repeated on all the DC coefficients $((M/n)\times(N/n))$ with DCT block size of $m\times m$ until the last step is reached. This scheme is called HDCT (Hierarchical DCT).

All DCT coefficients with each block can be reorganized when an image is taken as a single entity. In order to obtain the dyadic decomposition such as in EZW and SPIHT. For example, we can take each 8×8 DCT block as a 10-subband decomposition as was shown in Fig. 1(b). In Fig. 1(b), the ten subbands are composed of {0},{1},{2},{3},{4-7},{8-11},{12-15},{16-31},{32-47} and {48-63} coefficients respectively. One important step of HDCT involves the grouping of the same subbands for all DCT blocks. We call this grouping of DCT coefficients into a single DCT clustering entity. This important step is illustrated as in Fig. 2. In Fig. 2, Go0 means Group of subband 0,..., Go9 means Group of subband 9. Figure 3 is the diagrammatic illustration of the proposed organization strategy. This procedure is repeated on all the DC coefficients $((M/n)\times(N/n))$ with DCT block size of

$m \times m$ until the last step is reached. Figure 4 is an illustration of two-level reorganized DCT coefficients on the 512×512 Lena image, where $m = n = 8$. The gray values in Fig. 4 are obtained by $255 - 4 * ABS\ (Coefficient)$ with the exception of DC's for better visual presentation.

0	1	4	5	16	17	20	21
2	3	6	7	18	19	22	23
8	9	12	13	24	25	28	29
10	11	14	15	26	27	30	31
32	33	36	37	48	49	52	53
34	35	38	38	50	51	54	55
40	41	44	45	56	57	60	61
42	43	46	47	58	59	62	63

(a) 8×8 DCT coefficients



(b) 8×8 DCT taken as 10 subbands

Fig. 1 8×8 DCT block treated as three-scale tree

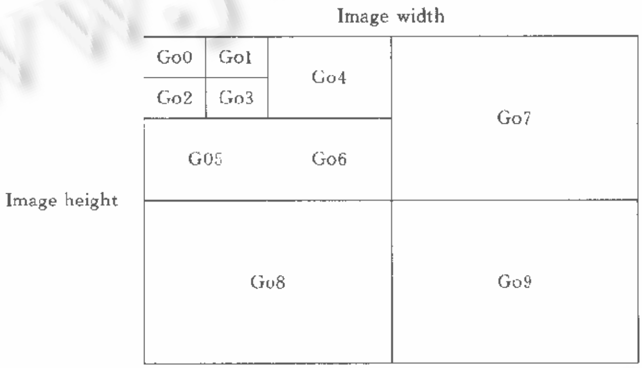


Fig. 2 Reorganization of 8×8 DCT blocks into a single DCT clustering entity

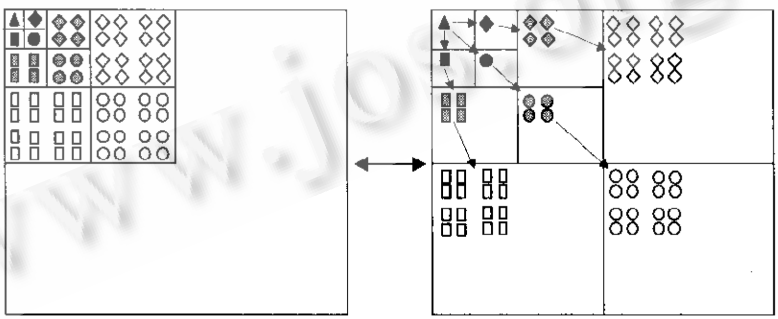


Fig. 3 The illustration of DCT coefficient reorganization

It can be seen that (1) signal energy is compacted mostly into DC coefficients and small number of AC coefficients related to the edges in spatial domain; (2) cross-subband similarity and decay of magnitude across subband; and (3) within subband clustering of significant coefficients. These DCT characteristics can be further utilized to DCT-based coders in order to get better compression performance and will widen DCT applications, such as applications on image compression, image retrieval, image recognition and so on.

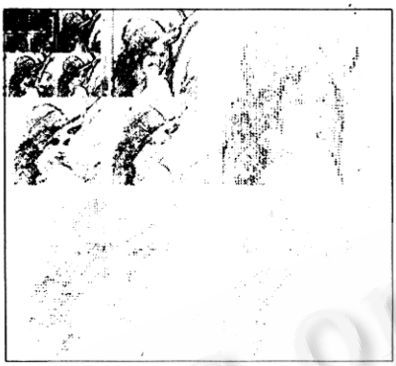


Fig. 4 Two-Level reorganized DCT coefficients on Lena image

1.2 Magnitude output definition

For all coefficients, $c_{i,j}$, in EZDCT, STQ, EZW and SPIHT, a value, n, can be obtained and outputted in their initialization step,

$$n=\lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\})\rfloor. \tag{1}$$

We observed that the maximum absolute coefficients in each block maybe greatly differ from each other, so it is not reasonable to only use one n for all blocks. In EZHDCT, we define n_k for each Block k,

$$n_k=\lfloor \log_2(\max_{(m,n)}\{|b_{m,n}|\})\rfloor, \tag{2}$$

where $b_{m,n}$ is the coefficients in Block k.

As there is much redundancy existed between n_k and n_{k-1}, a simple DPCM ($d_k=n_k-n_{k-1}$) is used to decorrelate them. In order to obtain d_k directly, such a mapping function is described as follows:

$$\delta_i=\begin{cases}2|d_i|-1, & \text{if } d_i<0\\ 2d_i, & \text{otherwise}\end{cases}. \tag{3}$$

The binary output of δ_i is $\overbrace{0\ldots0}^{\delta_i}1$. All its outputs become the header information of EZHDCT. At the current quantization step n_c, if $n_k<n_c$ then no output; else output its zerotree information as was done in Ref. [2].

1.3 Embedded zerotree quantization

We identify the parent-children relationships between DCT coefficients as in Ref. [4]. An embedded zerotree quantizer is then applied to the tree-structure DCT coefficients. During zerotree quantization, DCT coefficients in the higher level are quantized and outputted denoting its significance/insignificance information. The coefficients with the same subband from all DCT blocks are grouped together and scanned, starting from the DC coefficients. Zerotree quantization works by efficiently predicting the children nodes based on significance or insignificance of their parents. An embedded zerotree quantizer refines each input coefficient sequentially using a bitmap type of coding scheme, and stops when the size of the encoded bitstream reaches the exact target bit rate.

1.4 Arithmetic coding

As with any other coding method, the efficiency of our coder can be improved by entropy-coding its output at the expense of coding/decoding time. Practical experiments have shown that normally there is little to be gained by entropy-coding the coefficient signs or the bits put out during the refinement pass. On the other hand, the significant values are not equally probable, and there is a statistical dependence between a node and its descendents and also between the significance of adjacent blocks. In order to improve the compression performance and maintain the low computational complexity of our image coder, a binary adaptive arithmetic coder is designed as it is well known that good implementations of arithmetic coding produced bitstreams of length almost equal to the entropy bound[7]. The probability models of our arithmetic coder are related to the positions of the encoded coefficients within the block. Usually 0.3 to 0.7 dB PSNR improvement can be obtained using our arithmetic coder.

2 Postprocessing on Block Boundary

One of the major disadvantages of block-based image coding techniques is "blocking effect", especially at lower bit rates. There are two general approaches to reduce blocking effects. One is to use overlapping schemes in the encoding stage[8], and the other uses some postprocessing techniques in the decoding stage[9~12]. The above two approaches can effectively reduce the JPEG image compression blocking effects. We want to point out that blocking effects in the reconstructed images of JPEG and the embedded DCT image coders are different because of their quantization strategies. In this paper, a postprocessing technique based on Sobel operators and smoothing constraints is proposed. Sobel operators are used to detect the edges that should be well preserved across block boundaries. Smoothing is only carried out on the background along block boundaries. The threshold of Sobel operators is related to a given quantization matrix (Q factor) in the context of JPEG, but in our case, the threshold of the Sobel operators is related to the last step of the zerotree quantization and therefore is dependent on the input image's complexity and the specified bit rate.

2.1 Edge detection

Sobel operators can be defined by two kernels: $H_h(i,j)$ for the horizontal direction and $H_v(i,j)$ for the vertical direction. The horizontal and vertical gradient images, $G_h(i,j)$ and $G_v(i,j)$, are obtained by linear convolution with Sobel kernels:

$$G_h(i,j)=S(i,j)*H_h(i,j), \tag{4}$$

$$G_v(i,j)=S(i,j)*H_v(i,j), \tag{5}$$

where $S(i,j)$ is the decompressed image of our embedded DCT coder and $*$ is the linear convolution operator. The absolute, $M(i,j)$, is defined by

$$M(i,j)=|G_h(i,j)|+|G_v(i,j)|. \tag{6}$$

2.2 Smoothing constraint

Our smoothing is only carried out on the background along block boundaries. This is shown by the conditions in the smoothing operation:

$$S(i,j)=\begin{cases} aS(i,j)+bS(i-1,j), & \text{if } E(i,j)=0 \text{ and } i=0 \pmod 8 \\ aS(i,j)+bS(i+1,j), & \text{if } E(i,j)=0 \text{ and } i+1=0 \pmod 8 \\ aS(i,j)+bS(i,j-1), & \text{if } E(i,j)=0 \text{ and } j=0 \pmod 8 \\ aS(i,j)+bS(i,j+1), & \text{if } E(i,j)=0 \text{ and } j+1=0 \pmod 8 \\ S(i,j), & \text{otherwise} \end{cases}, \tag{7}$$

where $a+b=1$, and

$$E(i,j)=\begin{cases} 1, & \text{if } M(i,j)>T \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

In Eq. (8), $E(i,j)=1$ means $S(i,j)$ is an edge pixel, while $E(i,j)=0$ means $S(i,j)$ belongs to a background region.

The performance of our deblocking algorithm depends on the threshold T in Eq. (7), which is experimentally determined in consideration of the last step of zerotree quantization n_L of the input image, i.e.,

$$T=40(n_L+1)-150. \tag{9}$$

It should be pointed out that even at the same bpp requirement, various images may have different threshold, T, since their n_L.

3 Experimental Results and Conclusions

Our EZHDCT image coder has been tested on many standard images. As an example, "Lena" image (512×512) is illustrated in this paper where the level of DCT is set to 2. The coding results for "Lena" at different rates compared with the baseline JPEG[6] and the improved JPEG[13] are listed in Table 1. From Table 1, we can see that after arithmetic coding, EZHDCT consistently outperforms the baseline JPEG by a large margin (0.8 dB～1.7 dB), 0.3 dB －0.8 dB better than the improved JPEG in which JPEG's quantization table is optimized. Comparisons with the other two embedded DCT coders (EZDCT[4] and STQ[5]) are tabulated in Table 2 (all without arithmetic coding). Table 2 shows that when arithmetic coding is not used, EZHDCT can perform best compared with EZDCT and STQ (at an average 0.7 dB better than EZDCT, 0.1 dB better than STQ). The results of our deblocking algorithm are shown in Table 3 and Fig. 5. From these results, we see that our deblocking algorithm improves the objective quality of the decoded image in term of PSNR from 0.4 dB to 0.8 dB. In the mean time, most of the blocking effect is removed, and visually improves the decompressed images. From Fig. 5, we can see that the improvement after deblocking is obvious although the deblocking scheme is simple.

Table 1 Performance comparisons of EZHDCT with JPEG

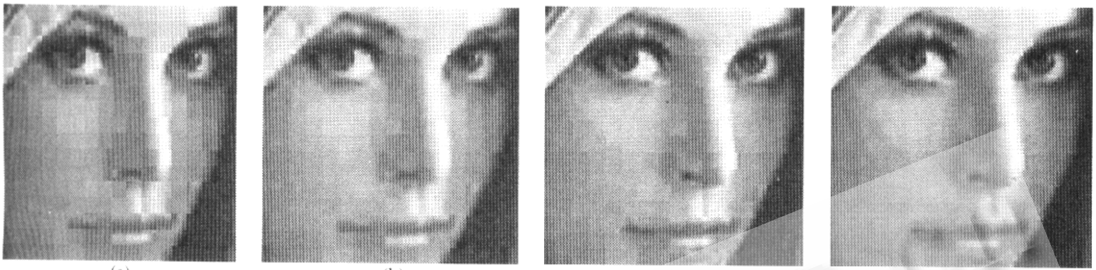
bpp	PSNR (dB)			
	JPEG	Improved JPEG	EZHDCT	
			No arithmetic coding	Arithmetic coding
0.25	31.6	31.9	31.7	32.4
0.50	34.9	35.5	35.5	36.0
0.75	36.6	37.5	37.5	37.8
1.00	37.9	38.8	39.2	39.6

Table 2 Performance comparisons of EZHDCT with embedded DCT coders: EZDCT and STQ

bpp	PSNR (dB)		
	No arithmetic coding		
	EZDCT	STQ	EZHDCT
0.25	30.7	31.2	31.7
0.50	34.8	35.4	35.5
0.75	37.1	37.0	37.5
1.00	38.7	38.8	39.2

Table 3 Deblocking results of the 512×512 "Lena" image

	PSNR (dB)	
bpp	Decoded image	Postprocessed image
0.15	29.4	30.2
0.20	30.9	31.6
0.25	32.4	32.9
0.30	33.2	33.6



(a) and (b) both decoded at 0.15 bpp, where (a) the decompressed image, PSNR=29.4 dB, and (b) the deblocking image from (a), PSNR=30.2 dB; (c) and (d) both decoded at 0.25 bpp, where (c) the decompressed image, PSNR=32.4 dB, and (d) the deblocking image from (c), PSNR=32.9 dB

Fig. 5 Deblocking results for center part of "Lena" image

References:

[1] Shapiro, J. Embedded image coding using zerotree of wavelet coefficients. IEEE Transactions on Signal Processing, 1993, 41(12):3445~3463.

[2] Said, A., Pearlman, W. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on Circuits System Video Technology, 1996,6(3):243~250.

[3] Xiong, Z., Ramchandran, K., Orchard, M. Space-Frequency quantization for wavelet image coding. IEEE Transactions on Image Processing, 1997,6(5):677~693.

[4] Xiong, Z., Guleryuz, O., Orchard, M. A DCT-based embedded image coder. IEEE Signal Processing Letters, 1996,3(11):289~290.

[5] Davis, G., Chawla, S. Significance tree quantization of the discrete cosine transform. In: Proceedings of the ICIP. Santa Barbara, California, 1997,1:600~603.

[6] Pennebaker, W., Mitchell, J. JPEG Still Image Data Compression Standard. New York: Van Nostrand Teinhold, 1993.

[7] Bell, T., Cleary, J.G., Written, I. Text Compression. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[8] Reeve, H., Lim, J. Reduction of blocking artifacts in image coding. Optical Engieering, 1984,23(1):34~37.

[9] Zakhor. Iterative procedures for reduction of blocking effects in transform image coding. IEEE Transactions on Circuits Systen Vedio Technology, 1992,2:91~95.

[10] Yang, Y., Galatsanos, N., Katsaggelos, A. Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images. IEEE Transactions on Circuits System Vedio Technology, 1993,3:421~432.

[11] Xiong, Z., Orchard, M., Zhang, Y. A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations. IEEE Transactions on Circuits System Vedio Technology, 1997,7(2):433~437.

[12] Wang, Z., Zhang, D. A Novel approach for reduction of blocking effects in low-bit-rate image compression. IEEE Transactions on Communication, 1998,46(6):732~734.

[13] Wu, S., Gersho, A. Rate-constrained picture adaptive quantization for JPEG baseline coders. In: Proceedings of the ICASSP, 1993,5:389~392.

基于层次 DCT 变换的嵌入式图像编码方法

赵德斌[1], 张大鹏[2], 高　文[1,3]

[1](哈尔滨工业大学 计算机科学与工程系,黑龙江 哈尔滨　150001);

[2](香港理工大学 计算机系,香港);

[3](中国科学院 计算技术研究所,北京　100080)

摘要: 提出了基于层次 DCT 变换的嵌入式零树编码方法. 尽管嵌入式零树编码方法首先使用于小波变换,但结合层次 DCT 结构,基于 DCT 的嵌入式零树编码方法依然可以取得很好的压缩效果. 层次 DCT 首先将输入图像划分

为8×8的图像块,然后对这些图像块进行第1层的DCT变换.因为相邻的DC系数高度相关,并且特别影响低比特率时的压缩效果,因此第2层的DCT变换被应用于这些DC系数上.这一过程重复进行,直到最后一步为止.因为DCT图像块通过重新组织,可以看做是类似于小波变换的子带结构,因此可以使用嵌入式零树编码方法.实验结果表明,所提出的基于层次DCT变换的嵌入式零树编码方法具有较低的计算复杂性和很好的压缩效果.例如,与JPEG相比较,对标准的512×512 Lena图像而言,此方法可获得0.8dB~1.7dB的提高.为了去除低比特率压缩所引起的图像块效应,提出了基于Sobel算子的后处理算法.实验结果显示,所提出的后处理算法对解码后的图像不论是主观效果还是客观效果均有提高.

关键词: 离散余玄变换;图像压缩;嵌入式零树编码方法;数据压缩;块效应;图像后处理

中图法分类号: TP391　　**文献标识码**: A