1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

AN ADAPTIVE 3-D DCT CODER FOR IMAGE COMPRESSION

Shen-Chuan Tai, Yung-Gi Wu and Chang-Wei Lin
Institute of Electrical Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Email: lcw@rose.ee.ncku.edu.tw

ABSTRACT

Discrete Cosine Transform (*DCT*) coding is a well-established technique for image data compression. As we know that 2-D compression techniques achieve better energy compactness than 1-D case in terms of transform coding. Our method is not only based on that assume but also extending the 2-D to 3-D to get more compression efficiency. A segmentation technique based on the local energy magnitude is developed to classify the non-overlapping sub-image into different energy levels. Then the 3-D block collector based on the segmentation results will gather the blocks with the same energy level together to form a 3-D block. Finally, 3-D DCT coder is employed to compress the 3-D block. Our 3-D technique is an adaptive strategy which collecting the sub-blocks with the same energy level together. Because each 3D block for DCT has same signal characteristics, more compression efficiency can be achieved. From the simulation results, we prove the performance in terms of compression ratio and decoded distortion.

Keywords: DCT, 3-D DCT, 3-D block.

1. Introduction

Due to the limit of network bandwidth and the storage capacity, the images must be compressed before they are transmitted and stored. Therefore, image compression techniques are needed to reduce the space and time for storage and transmission. As we know, 3-D video sequence could get better compression ratio than 2-D still image in general speaking. From the survey of literature, we know there are many papers discussed the 3-D compression techniques. But, the signal to be processed is video signals or consecutive CT images. This paper is the first papers that employing the 3-D compression concept to 2-D still image. We simulate the 2-D sub-blocks as 3-D sequences. Dividing the image into 8×8 non-overlapping sub-blocks. Regard the sub-blocks as the frames in video sequence. It could also raise the compression ratio in still image the same as in video sequence. The 3-D DCT coding is very efficient when the difference between sub-blocks is small. This is a typical case that the amount of energy in higher frequency component is low; hence, the energy compaction could be good. But, the difference of consecutive blocks and the complex of blocks will affect the performance of the 3-D DCT coding. In this paper we gather blocks with the same characteristics together to form a 3-D cube. Then 3-D DCT is used to compress the 3-D cube. Now, the sub-blocks in the 3-D block are similar, the 3-D DCT coding efficiency will be high. The price to be paid is a longer encoding delay and the requirement for a large memory size. Experimental results show that this technique could achieve better quality especially at very low bit rate.

2. Adaptive 3-D DCT Coder

2.1 Classification

The steps of classification are as follows:

Input: a still image

Output: classmap

Step 1. Sub-divide the image into 8×8 *non-overlapping* *sub-blocks.*

Step 2. Perform the discrete cosine transform of these sub-blocks. The location of six coefficients in a 8×8 *matrix is given in* ***Fig.1****.*

Step 3. Classify the class by CV as follows.

IF CV $\leq$ *T1, then use classmap I.*

ELSE IF CV > T1 & CV $\leq$ *T2, then use classmap II.*

ELSE IF CV > T2 & CV $\leq$ *T3, then use classmap III.*

ELSE use classmap IV.

For example: after classification we get a 8×8 classmap from a 64×64 image.

The 8×8 classmap of a 64×64 image is given in **Fig.2**.

2.2. Segmentation

After classification, we collect those subblocks to form 3-D blocks according to the classmap that we already got in the previous classification stage. The complete algorithm of segmentation is given as follows. Here, we define relative positions of blocks as in **Fig.3**.

Input: Classmap of the image. Each entry denotes the class of each 8×8 sub-block.

Output: Segmentation result.

Step 1: Scanning the classmap from top to down and left to right, starting position is (0,0)

Step 2: For each class in position (i,j) (i!=0 & j!=0)

If (classmap [i-1][j] ==classmap [i][j]) then

Segmenting sub-block [i][j] and sub-block [i-1][j] to the same segment.

Assign sub-block [i][j] being segmented

If (classmap [i][j-1]==classmap[I][j]) then

Segmenting sub-block [i][j] and sub-block[i][j-1] to the same segment.

Assign sub-block [i][j] being segmented

Step 3: If all the sub-blocks have being segmented to a segment, then fished else go to Step 2

A simple example of the segmentation is given in **Fig.4**. In this example, there are two segments of class 1 in the image. After segmentation, we form the 3-D blocks. Fig.6 illustrates the circumstance after segmenting the 3-D blocks. Fig.6 (a) shows the 3-D block after classification and segmentation. Four sub-3-D blocks with varied length are shown in Fig.6 (b). These four 3-D blocks will remain to have high inter-block pixel correlation. A possible pattern of the motion activity among the sub-blocks is given in Fig.6 (c). In this figure, there are blocks of segment change at position p1, p2 and p3. Out proposed segmentation technique not only split the 3-D block up into some sub 3-D blocks but also adjusts to the pixel intensity, which varies with position. Because each block has similar signal characteristics, therefore, the compression can achieve more efficiency then 2-D case.

2.3 3-D Compression

After segmentation there are many 3-D blocks, 3-D DCT is employed to these 3-D blocks now. For this image coder the 3-D DCT of an 3-D block (n × n × *length of 3-D block*) is defined as:

$$F(u,v,w)=\frac{8C(u)C(v)C(w)}{n\times n\times length}$$

$$\sum_{i=0}^{length-1}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} f(i,j,k)\cos[\frac{(2i+1)u\pi}{2\times length}]$$

$$\cos[\frac{(2j+1)v\pi}{2\times n}]\cos[\frac{(2k+1)w\pi}{2\times n}]$$

and the 3-D IDCT(inverse DCT) is defined as:

$$f(i,j,k)=$$

$$\sum_{i=0}^{length-1}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} C(u)C(v)C(w)F(u,v,w)\cos[\frac{(2i+1)u\pi}{2\times length}]$$

$$\cos[\frac{(2j+1)v\pi}{2\times n}]\cos[\frac{(2k+1)w\pi}{2\times n}]$$

where $C(w)=1/\sqrt{2}$ for $w = 0$.

1 for w = otherwise.

If the larger 3-D block we have, the higher compression ratio we achieve. For example, in the smooth region we could get huge 3-D blocks, because the subblocks in the smooth region are many and similar and they will be segmented into the same 3-D block. Not only raise compression ratio, but also alleviate blocky effects by making process on total subblocks of the 3-D block together by 3-D DCT, not in a block by block manner. Then, each of the 3-D DCT coefficients is uniformly quantized to produce the quantized coefficient. It is well known that the dc coefficient should be quantized accurately, or the blocking may be noticeable. Thus the dc coefficient is not be quantized. Thus, quantizer quantities the ac coefficients. Most of the energy is contained in few low-frequency coefficients, while the majority of the high-frequency coefficients have zero or near-zero values. The result of the quantization operation is a collection of smaller-valued coefficients, a large number of which are 0. These coefficients are then converted into a compact binary sequence using an entropy coder (here is a Huffman coder). The entropy coding operation starts with reordering the coefficients in descending order of expected value. This sequence has the benefit of collecting sequentially the largest number of zero-valued coefficient. The run-lengths of zero coefficients are computed, and the alphabet of symbols to be encoded becomes the run-length of zeros appended to the length of the non-zero coefficient. This binary sequence represents the compressed 3-D block. The block diagram of the 3-D DCT still image encoder is given in Figure 6(a). In 3-D DCT decoding, the steps from the encoding process are inverted and implemented in reversed order, as shown in Figure 6(b).

3. Simulation Results

In order to prove the effectiveness of our proposed technique, we employ our technique to natural image (Lena A 512×512 monochrome image) shown in Fig.7 (a) to evaluate the performance. Decoded images without blocking effects due to the employment of segmentation and 3-D technique. The decoded image of Lena is shown in Fig.7 (b). Our results show (in table 1) that the proposed strategy is more efficient than the conventional 2-D strategies.

4.Conclusion

In this paper, introduce an adaptive 3-D DCT image coder in order to achieve good subjective performance at low bit rate. *($\leq$ 0.3 bits/pixel).* 3-D DCT image compression compares favorably with other compression algorithms. This strategy not only raises the compression ratio but also alleviates blocky effects. The advantage of 3-D DCT coding is that we can utilize more zero coefficients in transform domain after quantization, then higher compression ratio we achieve. The 3-D DCT coding is employed to the 3-D block not in a block by block manner, therefore decrement of blocking effects can be found. In 3-D DCT coding, a large demand of memory is needed, and the heavy computational effort is unavoidable. Nevertheless, the benefit of 3-D frequency coding is deserved us to study on the techniques for image coding.

Reference

[1] Y. L. Chan and W. C. S, "Variable Temporal-Length 3-D Discrete Cosine Transform Coding," IEEE Trans. Image Processing, vol. 6, NO. 5, pp.758 –763, May, 1997

[2] A. Ramaswamy and W. B. Mikhael," A Mixed Transform Approach for Efficient Compression of Medical Images," IEEE Trans. Medical Imaging, vol. 12, NO. 4, pp.803-811, 1996

[3] W. H. Chen and C. H. Smith, "Adaptive Coding

of Monochrome and Color Images," IEEE Trans. Commun. COM-25, pp.1285-1292, 1977

[4] B. Chitprasert and K. R. Rao, "Discrete cosine transform filtering, Signal Process., vol 19, pp. 233-245, 1990.

[5] K. R. Rao, "Theory and the applications of the discrete cosine transform," Jordan Intl. Electrical and Electronic Engrg. Conf., pp. 259-264, Amman, Jordan, Apr.-May 1985.

[6] K. R. Rao and J. J. Hwang, Techniques and Standards for Image, Video, and Audio Coding, Prentice Hall PTR, New Jersey, USA, 1996

[7] Y. G. Wu and S. C. Tai, "Medical Image Compression Using 2×2 Discrete Cosine Transform," *Optical Engineering*, Vol. 37, NO. 5, pp. 1539-1546, May 1998

DC	V_1	V_2					
H_1	S_1						
H_2		S_2					

Fig. 1 The six coefficients are selected to classify.

1	1	1	2	1	2	3	4
2	1	2	2	1	2	3	3
2	1	1	1	1	3	2	3
2	2	1	1	1	3	2	2
2	3	1	4	4	4	4	4
3	3	3	4	4	1	1	1
2	3	2	4	4	1	1	1
2	2	2	4	4	1	1	2

Fig. 2 The 8×8 classmap of a 64×64 image. Here 1,2,3,4 denote the class each block belongs to.

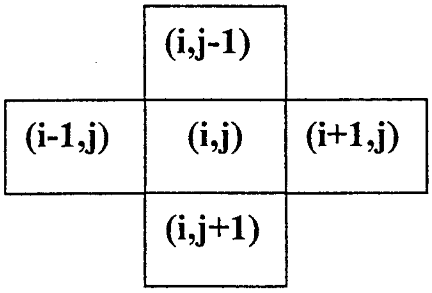


Fig. 3 Relative positions of block

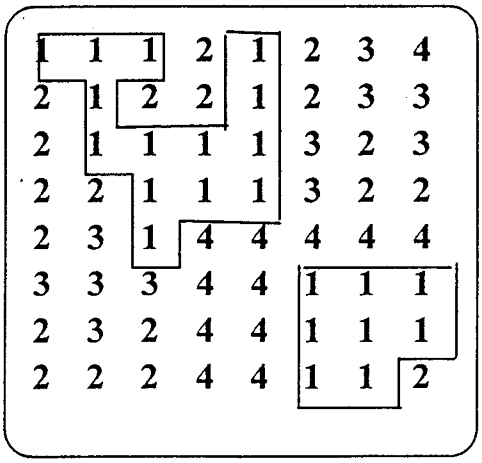


Fig. 4 Collecting and segmentation results of class map

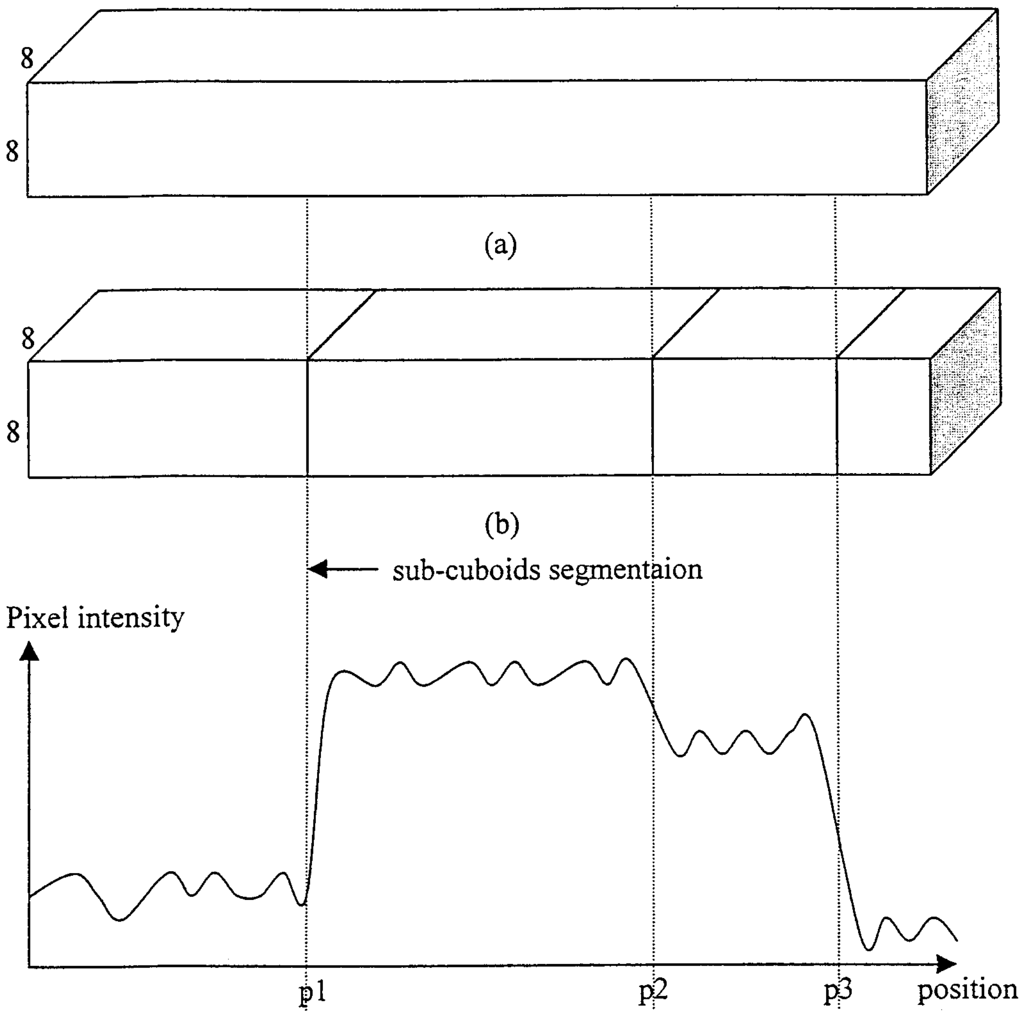
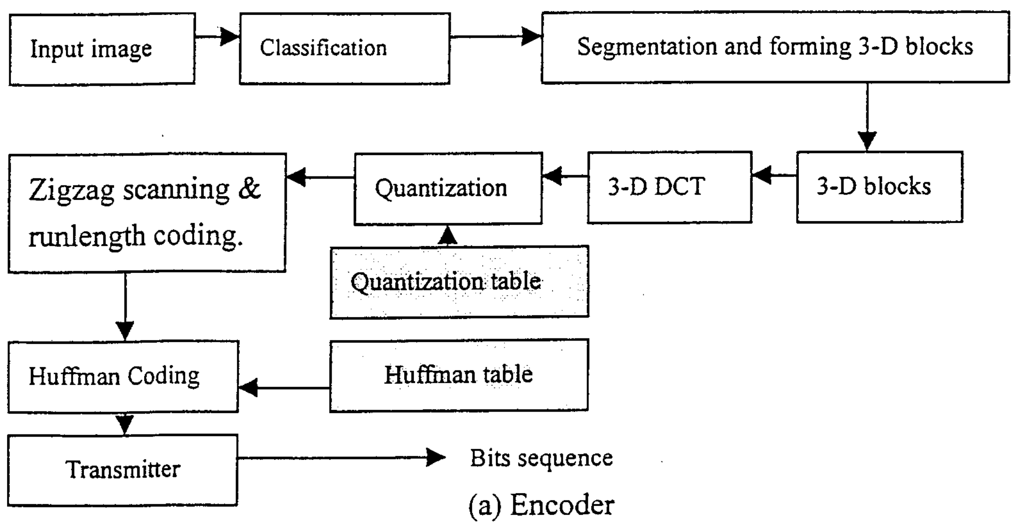
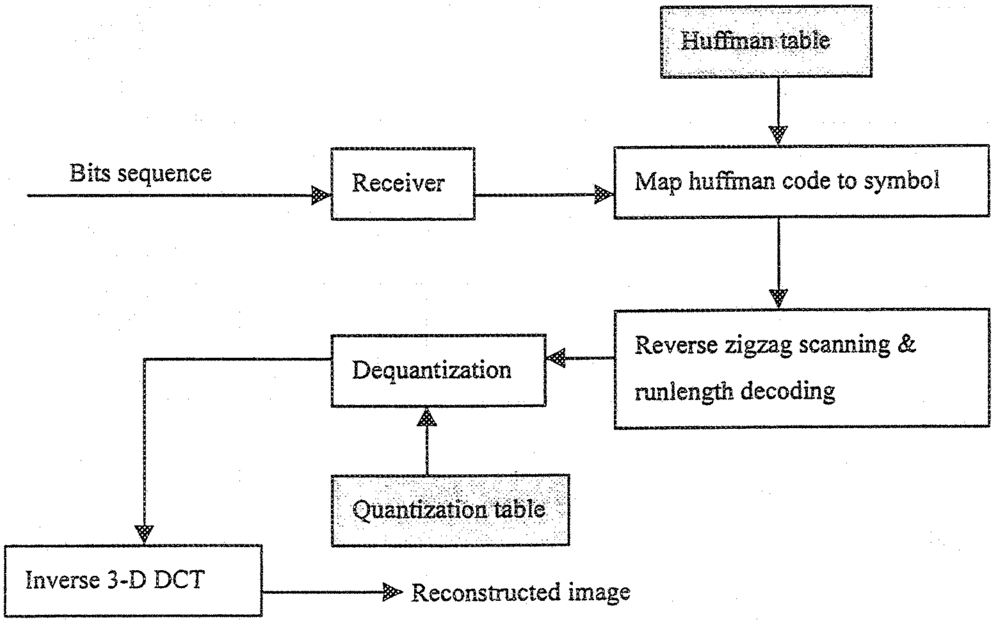


Fig. 5 Splitting cube to sub-cubes

(a) 3-D block of each class. (b) 3-D block after segmentation. (c) Pixel intensity, which varies with position.



(a) Encoder



(b) Decoder

Fig. 6 Block diagrams of proposed system (a) encoder and (b) decoder.

(a) (b) (c)

Fig. 7(a) The original image of Lena. (b) The decoded image of Lena. (c) The difference image

Table 1 Simulation Results (CR denotes compression ratio)

	Proposed Technique		
Test image	Bit Rate	CR	PSNR
Lena	0.293	27.3	30.56 dB