FAST INTERFRAME TRANSFORM CODING BASED ON CHARACTERISTICS OF TRANSFORM COEFFICIENTS AND FRAME DIFFERENCE

Yui-Lam CHAN and Wan-Chi SIU

Department of Electronic Engineering,
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Tel : (852)-27666229 Fax : (852)-23628439

ABSTRACT

The interframe transform coding has been seldom used in practice because of the considerable computational complexity. To reduce the computational complexity, a fast algorithm is proposed in this paper. The algorithm reduces the number of operations by limiting the calculation of transform coefficients without significant quality degradation based on the distribution of transform coefficients. Then, different modes of transformation are performed according to frame difference. This proposed fast interframe coding, like the MPEG, has an asymmetric property with decoding being much faster than encoding. Computer simulations show that this fast algorithm can significantly reduce the computational burden in interframe transform coding and is even faster than the MPEG-like coding.

1. INTRODUCTION

An interframe discrete cosine transform(DCT) of a block can reduce both the spatial and temporal correlation of the picture elements[1,2]. Besides, the interframe DCT is more suitable in "quick search mode" than the interframe hybrid transform/DPCM coding, and also it is less sensitive to channel errors [3]. However, the interframe transform coding has been seldom used in practice because of the considerable computational complexity and the requirement of extensive memory. The three dimensional discrete cosine transform(3D-DCT) of an $8 \times 8 \times 8$ pixel block, being a separable transform, can be implemented by a series of 1D-DCT along each dimension. Totally 192 1D-DCT per block are required to perform. To reduce the computational complexity, a fast algorithm is proposed in this paper. The algorithm reduces the number of operations by limiting the calculation of transform coefficients based on the property of the distribution of the transform coefficients. And then, only low frequency coefficients are calculated in temporal DCT(along temporal direction) based on the frame difference. This proposed algorithm, like the MPEG, has an asymmetric property with decoding being much faster than encoding. Computer simulations show that this fast algorithm can significantly reduce the computational burden in interframe transform coding without significant quality degradation and is even faster than the MPEG-like coder.

2. THREE DIMENSIONAL DISCRETE COSINE TRANSFORM(3D-DCT)

For interframe transform coding, each image sequence is divided into $8 \times 8 \times 8$ pixels and a forward 3D-DCT is carried out on each block, defined as follows:

$$X(u,v,w) = \frac{1}{64}\sum_{k=0}^{7}\sum_{j=0}^{7}\sum_{i=0}^{7} C(u)C(v)C(w)x(k,j,i)$$
$$\cos\pi\frac{(2k+1)u}{16}\cos\pi\frac{(2j+1)v}{16}\cos\pi\frac{(2i+1)w}{16}$$

where $x(k,j,i)$ is the pixel value in the image sequence and $C(n) = \frac{1}{\sqrt{2}}$ for $n = 0$, otherwise $C(n) = 1$. In this paper, Lee's algorithm [4] is used to perform the 1D-DCT. The number of real additions and real multiplications for an N-point sequence of the Lee's algorithm is $(\frac{3N}{2})log_2N - N + 1$ and $(\frac{3N}{2})log_2N$ recpectively. The 3D-DCT, being a separable transform, can be implemented by cascading three 1D-DCT along each dimension. Totally 192 1D-DCT are required to perform. So the numbers of operations, for N=8, are 5568(129×29) real additions and 2304(192×12) real multiplications.

3. A FRAMEWORK OF PROPOSED FAST ALGORITHM

3.1. 3D-DCT's coefficients distribution

In image sequences of the real world, block motion is usually modest in speed, smooth in the spatial domain

---

Acknowledgement : This work was supported by The Croucher Foundation PolyU340/055.

0-7803-2570-2/95 $4.00 ©1995 IEEE

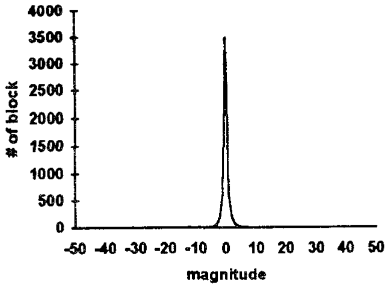


Figure 1: 3D-DCT's coefficient distribution of X(1,2,2).

and varying slowly in the temporal domain. This property has several implications: one of them is related to the magnitude distribution of the coefficients of the 3D-DCT; it says that the distribution of the high frequency coefficients tends to zero after *quantization*. In the block with still motion, the coefficients except the first frame tend to zero. Even, in the block with modest motion, the coefficients in the third, forth, etc., frames also tend to zero, as illustrated in figure 1.

3.2. Low frequency DCT coefficient calculation

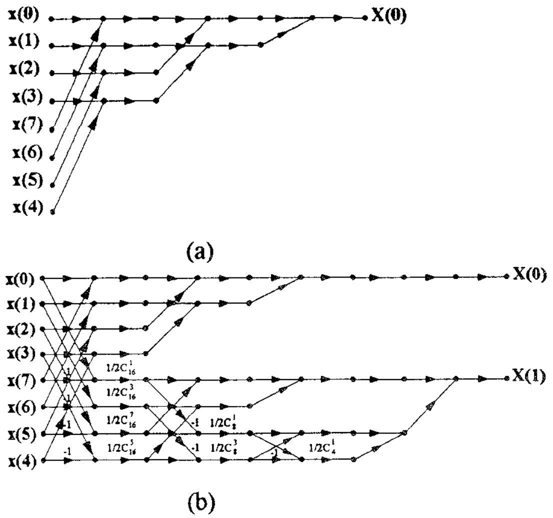
To perform the three dimensional discrete cosine transform coding, the number of operations required is often too high. However, based on the property discussed above, it is unnecessary to calculate all coefficients in each block(in low or modest motion block). In other words, the most useful information about the signal is kept by the low frequency, therefore, only low-frequency DCT coefficients are necessary to be computed in large portion of the block. In this section, we will discuss the computational save for calculating only *low frequency coefficients* based *on* Lee's *algorithm*. Figures 2 and 3 show the flowgraphs of the one dimensional forward and inverse discrete cosine transforms for only calculating low frequency coefficients respectively. The numbers of operations in various lengths for obtaining the required coefficients are shown in table 1. It is seen that the computational complexity will be greatly reduced, especially for the inverse transform.

3.3. Motion Detection

The basic idea of motion detection is to find the degree of motion in the block. In this paper, we propose to use the frame difference to identify the degree of motion. The frame difference(FD) of each block is defined as:

$$FD = \sum_{i=0}^{7} \sum_{j=0}^{7} \sum_{k=0}^{6} |I(k+1,j,i) - I(k,j,i)|$$

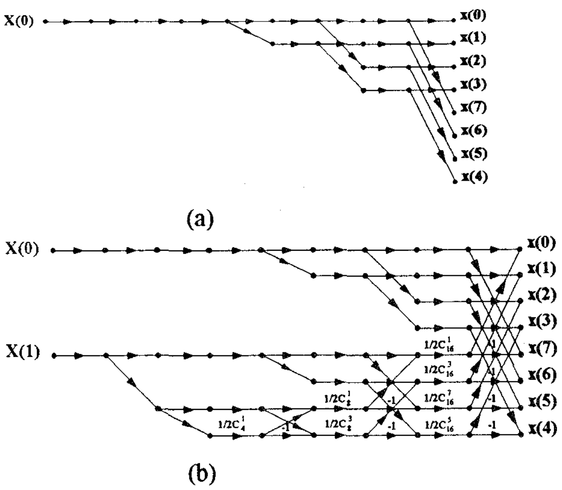
where $I(k,j,i)$ is the pixel value in the image sequence.



(a)

(b)

Figure 2: Signal flowgraphs for computing forward 8-length DCT to produce only (a) one output coefficient and (b) two output coefficients.



(a)

(b)

Figure 3: Signal flowgraphs for computing inverse 8-length DCT with only (a) one input coefficient and (b) two input coefficients available.

Table 1: Number of operations for n coefficients of 8-length DCT

	n	real additions	real multiplications
Forward	1	7	0
	2	20	7
	8	29	12
Inverse	1	0	0
	2	14	7
	8	29	12

4. THE NEW PROPOSED ALGORITHM

Considering all discussions above, we can now propose a fast 3D-DCT coding and decoding scheme.

4.1. Description of encoding

Since, a large number of transform coefficients are equal to zero after quantization, it is not necessary to calculate all transform coefficients in order to reduce the computational burden. Based on the frame difference of each block, all image blocks are classified into one of the following four classes, where T_i is the predefined threshold for i=1,2 and 3.

Class $1(FD < T_1)$:they can be identified as non moving blocks by the motion detector. They usually belong to the unchanged area like the background in images. In other words, the pixel value of each frame is nearly equal, so only the first frame 2D-DCT is needed to perform(16 1D-DCT).

Classes $2(T_1 \leq FD < T_2)$:these blocks are originally identified as low moving blocks. Their transform coefficients except the ones in the first frame are nearly zero after 3D-DCT. For such blocks, all eight frames 2D-DCT are performed, and only the lowest coefficient in temporal DCT is calculated as shown in figure 2(a).

Classes $3(T_2 \leq FD < T_3)$:these blocks are identified as modest moving blocks. But, their transform coefficients except the ones in the first and second frames are nearly zero. For such blocks, all eight frames 2D-DCT are performed, and the first and second low-frequency coefficients in temporal DCT are calculated as shown in figure 2(b).

Class $4(T_3 \leq FD)$:these blocks are considered as high moving block, a large number of transform coefficients have significant value. So all coefficients are required to calculate.

The transformed block is quantized and scanned according to first spatial, second temporal zig-zag scanning. Then, a VLC is applied and finally the encoding signal is transmitted.

4.2. Description of decoding

The bitstream goes through the VLC decoding, and onward to inverse first spatial, second temporal zig-zag scanning and inverse quantization.

Before performing each separate inverse 1D-DCT, coefficients in each row, column and temporal direction are scanned. Different modes of 1D-DCT are performed as summarized below.

1. If all coefficients are equal to zero, no inverse 1D-DCT operation is required. Otherwise, the operation is processed to step(2).

Table 2: Number of operations for n coefficients of 8-length DCT

	Encoding		Decoding	
Class	add.	mult.	max. number of add.	max. number of mult.
1	464	192	464	192
2	4160	1536	464	192
3	4992	1984	1824	832
4	5568	2304	5568	2303

2. If only the lowest frequency coefficient has value, the inverse 1D-DCT is calculated as shown in figure 3(a). Otherwise, the operation is processed to step(3).
3. If only the first and second lowest frequency coefficients are not equal to zero, the inverse 1D-DCT is calculated as shown in figure 3(b). Otherwise, the operation is processed to step(4).
4. A fully inverse 1D-DCT is needed to perform.

5. COMPUTATIONAL COMPLEXITY

Different classes perform different modes of forward 3D-DCT in encoding. The numbers of operations in different classes are shown in table 2. In decoding, the number of operations is based on the number of zero coefficients in the block. The number of additions and multiplications required are $14N_2 + 29N_3$ and $7N_2 + 12N_3$ respectively, where N_2 is the number of inverse 1D-DCT of figure 3(b) required and N_3 is the number of fully inverse 1D-DCT required within the block. The maximum number of operations is also shown in table 2. From table 2, an asymmetric property with decoding much faster than encoding is shown. The number of additions required for calculating FD per block is 847.

6. EXPERIMENTAL RESULTS

A series of computer simulations have been conducted to evaluate the performance of our proposed algorithm. The video sequences "football", "table tennis" and "salesman" have been used. Table 3 shows the number of blocks in different classes. The computational save compared with the conventional 3D-DCT with frame difference classification is shown in table 4. The computational complexity is significantly reduced by, about

Table 3: Number of blocks in different classes

Class	football	table tennis	salesman
1	1360	1226	1340
2	1129	1247	1435
3	659	1097	1830
4	2132	1830	2493

Table 4: Computational save in encoding and decoding

	Encoding		Decoding	
video sequence	add.	mult.	add.	mult.
football	15.09%	32.47%	60.87%	60.25%
table tennis	13.54%	31.33%	68.80%	68.13%
salesman	11.28%	28.94%	78.75%	78.39%

Table 5: Number of operations in our proposed coding algorithm and MPEG like coding[5] with 8 × 8 block size and w=8.

	MPEG like coding			
	Encoding		Decoding	
video sequence	add.	mult.	add.	mult.
all	15472	2880	4160	1536
	Our proposed algorithm			
football	4726	1556	2179	916
table tennis	4814	1582	1737	734
salesman	4940	1637	1183	498

25%-32% and 60%-78%, in encoding and decoding respectively, with only 0.2-0.5 dB degradation.

The computational complexity of our proposed coding algorithm is also compared with the MPEG like coder[5]. The comparisons presented in this paper are based on a MPEG coding with IPPPPPPP group of picture (GOP). In MPEG coding, it requires 8 forward 2D-DCT, 7 inverse 2D-DCT and 128 × 7 pixel additions/subtractions for one GOP. Besides, it also requires $7(N \times N) \times (2w+1)^2$ additions for full search with the maximum displacement w for one GOP. In decoding, it only requires 8 inverse 2D-DCT and 64 × 7 pixel additions for one GOP. Table 5 compares the computational complexity of our proposed algorithm and the MPEG-like coding. It shows that our proposed algorithm is much faster than the MPEG method especially in decoding, while in table 5, the numbers of additions and multiplications of our proposed algorithm in decoding are 1.91-3.52 and 1.67-3.08 times less than that for the MPEG-like coder respectively, and the numbers of additions and multiplications of our proposed algorithm are about 3.2 and 1.8 times less than that for MPEG-like coder in encoding.

7. CONCLUSIONS

A new fast interframe discrete cosine transform coding is proposed to compensate the computational burden of the interframe transform coding. This proposed fast algorithm uses the distribution of the 3D-DCT transform coefficients, then only low frequency coefficients are calculated in temporal DCT based on the frame difference. This proposed algorithm can reduce by, about 25%-32% and 60%-78% computational complexity of traditional 3D-DCT in encoding and decoding respectively without significant quality degradation of the reconstructed image sequences. The computational burden is even much lower than that required for MPEG-like coder, especially in decoding. Like the MPEG, the proposed algorithm has an asymmetric property with decoding being much faster than encoding. Thus, this approach is certainly an efficient technique for asymmetric applications which require frequent use of the decoding process, but for which the encoding process is performed once and for all at the production of the program. This proposed algorithm can still be further improved for combining with some fast DCT algorithms[6,7] which reduce the operations for calculating the low-frequency DCT components, a fruitful direction for further investigation.

8. REFERENCES

[1] J.A.Roese and W.K.Pratt, "Theoretical performance methods for interframe transform and hybrid transform/DPCM coders", Proceedings SPIE, Vol-87, August 1976,pp.172-179.

[2] J.A.Roese and W.K.Pratt, "Interframe cosine Transform image coding", IEEE Transactions on communications, Vol-25, November 1977, pp.1329-1338.

[3] Olivier Chantelou, Christian Remus, "Adaptive transform coding of HDTV picture", International workshop on signal processing of HDTV, L'Aguila, Italy, Elsevier Science Publishers, North Holland, Feb-March,1988, pp.231-238.

[4] B.G.Lee, "A new algorithm for the discrete cosine transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-32, Dec. 1984, pp. 1243-1245.

[5] D.J. LeGall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, Vol-34, April 1991, pp.47-58.

[6] D.W.Meghanad, H. Ganesh, "A new algorithm for the discrete cosine transform of arbitrary number of points", IEEE Transactions on Computers, Vol. C-29, April 1980, pp.269-277.

[7] Z. Wang, "Pruning the fast discrete cosine transform", IEEE Transactions on Communications, Vol-39, May 1991, pp.640-643.