16×16 Integer Cosine Transform for HD Video Coding

Jie Dong and King N. Ngan

Dept. of Electronic Engineering, The Chinese University of Hong Kong
Hong Kong SAR
{jdong, knngan}@ee.cuhk.edu.hk

Abstract. High-Definition (HD) videos often contain rich details as well as large homogeneous regions. To exploit such a property, Variable Block-size Transforms (VBT) should be in place so that transform block size can adapt to local activities. In this paper, we propose a 16×16 Integer Cosine Transform (ICT) for HD video coding, which is simple and efficient. This 16×16 ICT is integrated into the AVS Zengqiang Profile and used adaptively as an alternative to the 8×8 ICT. Experimental results show that 16×16 transform can be a very efficient coding tool especially for HD video coding.

Keywords: HD video coding, ICT, AVS.

1 Introduction

High-Definition (HD) videos often contain rich details as well as large homogeneous regions. In other words, spatial correlation varies greatly throughout an HD video sequence. To exploit such a property, Variable Block-size Transforms (VBT) should be employed, such that smoother regions can be transformed using larger transforms for better energy compaction and better visual quality, and for more detailed areas, smaller transforms can avoid the ringing artifacts and reduce complexity.

The existing video coding systems with VBT use 8×8 as the largest transform block size. However, our study shows that 16×16 transforms are very efficient in coding large homogeneous regions in HD videos, thus improving the overall performance. In this paper, a 16×16 transform is proposed to the Audio Video Coding Standard (AVS) Zengqiang Profile [1] which is the developing profile of the national standard of China aiming at HD video coding. Since there is an 8×8 ICT already existing in the AVS Zengqiang Profile, the 2 transforms are used adaptively according to the local activities of frames. The proposed 16×16 transform is designed as a type of Integer Cosine Transform (ICT), which was first introduced by W. K. Cham in 1989 [2] and has been further developed in recent years. ICT can be implemented using integer arithmetic without mismatch between encoder and decoder and if well-designed, can provide almost the same compression efficiency as Discrete Cosine Transform (DCT).

The contribution of this paper mainly lies in 2 parts. Firstly, a 16×16 ICT is designed, which avoids mismatch between encoder and decoder and has very close decorrelation capability to that of the DCT. Furthermore, to minimize the increasing computational complexity caused by the additional larger transform, we design the 16×16 ICT suitable for simple 16-bit integer arithmetic implementation and compatible

Y. Zhang et al. (Eds.): PCM 2006, LNCS 4261, pp. 114 – 121, 2006.

with the 8×8 one. Secondly, to integrate the proposed 16×16 ICT into the AVS Zengqiang Profile, the related problems of transform size selection, 16×16 intra prediction, and 16×16 block entropy coding have been solved.

The remainder of the paper is organized as follows. A brief review of ICT is given in Section 2. Section 3 describes the proposed 16×16 ICT. Section 4 introduces in detail how the related problems are solved when the 16×16 ICT is integrated into the AVS Zengqiang Profile. Section 5 reports the experimental results, followed by the conclusion in Section 6.

2 Review of ICT

ICT originates from the DCT in order to simplify the computation of DCT and it enables bit-exact implementations. Its transform matrix is generated from the DCT matrix with the principle of dyadic symmetry [2] and contains only integers. Since the transform matrix is not normalized, a normalization process, known as scaling, is required after transformation. The process of transformation and scaling can be expressed by (1) and (2), respectively,

$$\mathrm{F}_{n\times n} = \mathrm{T}_n \times \mathrm{f}_{n\times n} \times \mathrm{T}_n^{\mathrm{T}} \tag{1}$$

$$\mathrm{S}_{n\times n} = \mathrm{F}_{n\times n} \,//\, \mathrm{R}_{n\times n} \tag{2}$$

where $\mathrm{f}_{n\times n}$ is the input data, T_n is the n×n transform matrix, $\mathrm{S}_{n\times n}$ is the transformed data and symbol // indicates that each element of the left matrix is divided by the element at the same position of the right one. The elements in matrix $\mathrm{R}_{n\times n}$ are all integers and can be derived from the norms of basis vectors of T_n. Interested readers may refer to [2] for details.

The divisions in (2) are usually approximated by integer multiplication and shifting as shown in (3),

$$\mathrm{S}_{n\times n} = \mathrm{F}_{n\times n} \otimes \mathrm{P}_{n\times n} >> N \tag{3}$$

where symbol $\otimes$ indicates that each element of the left matrix is multiplied by the element at the same position of the right one and $>>N$ means right shifting N bits. $\mathrm{R}_{n\times n}$ from (2) and $\mathrm{P}_{n\times n}$ from (3) satisfy (4) and $\mathrm{P}_{n\times n}$ is defined as the scaling matrix.

$$\mathrm{P}_{n\times n}(i,j) \times \mathrm{R}_{n\times n}(i,j) = 2^N \tag{4}$$

Similarly, for the inverse ICT, the whole process including inverse scaling and inverse transformation can be represented in (5) as

$$\mathrm{f}_{n\times n} = \left(\mathrm{T}_n^{\mathrm{T}} \times (\mathrm{S}_{n\times n} \otimes \mathrm{P}_{n\times n}) \times \mathrm{T}_n\right) >> N \tag{5}$$

3 The Proposed 16×16 ICT

3.1 16-Bit Integer Implementation

As shown in (6), the transform matrix T_{16} is very simple. All coefficients have only 7 different magnitudes and can be represented by 5-bit integers. Due to the small

magnitudes of the coefficients, the transform matrix is suitable for 16-bit integer arithmetic implementation.

$$T_{16} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 11 & 11 & 11 & 9 & 8 & 6 & 4 & 1 & -1 & -4 & -6 & -8 & -9 & -11 & -11 & -11 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 & -10 & -9 & -6 & -2 & 2 & 6 & 9 & 10 \\ 8 & 6 & 4 & 1 & -11 & -11 & -11 & -9 & 9 & 11 & 11 & 11 & -1 & -4 & -6 & -8 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 & 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 11 & 9 & -11 & -11 & -4 & -1 & 8 & 6 & -6 & -8 & 1 & 4 & 11 & 11 & -9 & -11 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 & -9 & 2 & 10 & 6 & -6 & -10 & -2 & 9 \\ 1 & 4 & -6 & -8 & 9 & 11 & -11 & -11 & 11 & 11 & -11 & -9 & 8 & 6 & -4 & -1 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 4 & -1 & -8 & 6 & 11 & -9 & -11 & 11 & -11 & 11 & 9 & -11 & -6 & 8 & 1 & -4 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 & -6 & 10 & -2 & -9 & 9 & 2 & -10 & 6 \\ 11 & -11 & -9 & 11 & -6 & 8 & 1 & -4 & 4 & -1 & -8 & 6 & -11 & 9 & 11 & -11 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 & 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 9 & -11 & 11 & -11 & -1 & 4 & -6 & 8 & -8 & 6 & -4 & 1 & 11 & -11 & 11 & -9 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 & -2 & 6 & -9 & 10 & -10 & 9 & -6 & 2 \\ 6 & -8 & 1 & -4 & 11 & -11 & 9 & -11 & 11 & -9 & 11 & -11 & 4 & -1 & 8 & -6 \end{bmatrix} \tag{6}$$

The 8×8 ICT in the AVS Zengqiang Profile is a type of Pre-scaled Integer Transform (PIT) [3], which means that the scaling of the inverse ICT is moved to the encoder side and combined with that of the forward ICT as one single process. With PIT, there is no scaling and thus no scaling matrix stored for decoding, whilst the computational complexity and memory requirement on the encoder side remain unchanged. Due to these advantages of PIT and to be compatible with the 8×8 ICT, the proposed 16×16 ICT is also designed as a PIT.

The whole process of ICT, including transformation and scaling, can be implemented by (7) and (8), respectively.

$$F_{16\times16} = \left(T_{16} \times f_{16\times16} \times T_{16}^{T} + 2^{6}\right) >> 7 \tag{7}$$

$$S_{16\times16} = (F_{16\times16} \otimes P_{16\times16} + 2^{15}) >> 16 \tag{8}$$

Right shifting 7 bits in (7) and 16 bits in (8) guarantee that the intermediate results for multiplication or addition are all smaller than 16 bits. And the factors, 2^6 and 2^{15} in (7) and (8) are for rounding.

For inverse ICT, since the PIT is employed, the inverse scaling step is saved and the implementation of (5) is simplified as below:

$$b_{16\times16} = \left(S_{16\times16} \times T_{16} + 2^{2}\right) >> 3 \tag{9}$$

$$f_{16\times16} = \left(T_{16}^{T} \times b_{16\times16} + 2^{6}\right) >> 7 \tag{10}$$

Right shifting 3 bits for horizontal transform in (9) and 7 bits for vertical transform in (10) ensure that any intermediate result in the inverse ICT process is smaller than 16 bits.

3.2 The Performance of the 16×16 ICT

Besides the simple structure, the proposed ICT also shows good energy compaction capability especially in those homogeneous regions in HD videos. This is because the

first 3 basis vectors of the transform matrix which represent relatively low frequency components resemble those of the DCT, including the same dyadic symmetries. Suppose that the input data are so highly correlated which can be approximated by a 1-D first-order stationary Markov source with correlation coefficient ρtending to 1. Table 1 shows the transform efficiency of Karhunen-Loeve Transform (KLT), DCT and the proposed ICT, respectively, with different ρ, where the transform efficiencyηcan be calculated by (11). As evident in Table 1, the efficiency of the proposed ICT is very close to that of the DCT, especially when the input data is very highly correlated, e.g., ρ= 0.95.

$$\eta = \frac{\sum_{i=1}^{15} \left| S_{16\times16}(i,i) \right|}{\sum_{i=0}^{15}\sum_{j=0}^{15} \left| S_{16\times16}(i,j) \right|} \tag{11}$$

Table 1. The transform efficiency of different transforms

ρ	KLT	DCT	Proposed ICT
0.95	1.00	0.88	0.86
0.90	1.00	0.83	0.79
0.85	1.00	0.80	0.75

Another merit of the transform matrix is that the norms of basis vectors are very close to each other. This property is very important when PIT is employed, because with PIT, the inverse scaling is moved to the encoder side, which can be viewed as applying a frequency weighting matrix to the transformed signals. In order not to alter the energy distribution of the signals in the transform domain significantly, the coefficients in the scaling matrix should be closed to each other, which can be guaranteed by making the norms of basis vectors in the transform matrix very close to each other.

3.3 The Compatibility with the 8×8 ICT in the AVS Zengqiang Profile

It is nature that the 16×16 DCT is compatible with the 8×8 one. However, it is not always true for ICT. To ensure that the proposed 16×16 ICT can be compatible with the 8×8 one used in the AVS Zengqiang Profile, we extended the 8 basis vectors in the 8×8 transform matrix by the fifteenth even dyadic symmetry to form the even rows in (6). For the odd rows in (6), we slightly modified the dyadic symmetries in the 16×16 DCT matrix in order to ensure orthogonality whilst keeping the magnitudes of the coefficients small enough. It has been proved in 3.2 that the degradation of decorrelation capability caused by alteration of dyadic symmetry is negligible.

3.3.1 Compatible Transformation

To show how the 8×8 transformation in the AVS Zengqiang Profile is involved in the proposed 16×16 one, two modules, defined as T_{8u} and T_{8d} in (12), can be extracted from (6) and T_{8u} is exactly the same as the 8×8 transform matrix. Figure 1 shows how these two modules work for the 16×16 forward transformation when implemented by

butterfly structure. Instead of using butterfly structure, these two modules in Figure 1 can also be implemented directly using matrix multiplication.

$$T_{8u} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix} \quad T_{8d} = \begin{bmatrix} 11 & 11 & 11 & 9 & 8 & 6 & 4 & 1 \\ 8 & 6 & 4 & 1 & -11 & -11 & -11 & -9 \\ 11 & 9 & -11 & -11 & -4 & -1 & 8 & 6 \\ 1 & 4 & -6 & -8 & 9 & 11 & -11 & -11 \\ 4 & -1 & -8 & 6 & 11 & -9 & -11 & 11 \\ 11 & -11 & -9 & 11 & -6 & 8 & 1 & -4 \\ 9 & -11 & 11 & -11 & -1 & 4 & -6 & 8 \\ 6 & -8 & 1 & -4 & 11 & -11 & 9 & -11 \end{bmatrix} \tag{12}$$

Since the 8×8 transform matrix, T_{8u}, is involved in the 16×16 one, not only can an independent module set up specially for the 8×8 transformation be saved, but also the output date from the 8×8 transformation can be reused by the 16×16 one.

The compatibility of the forward transformations has been described in detail, and it is very similar for the case of inverse ones.

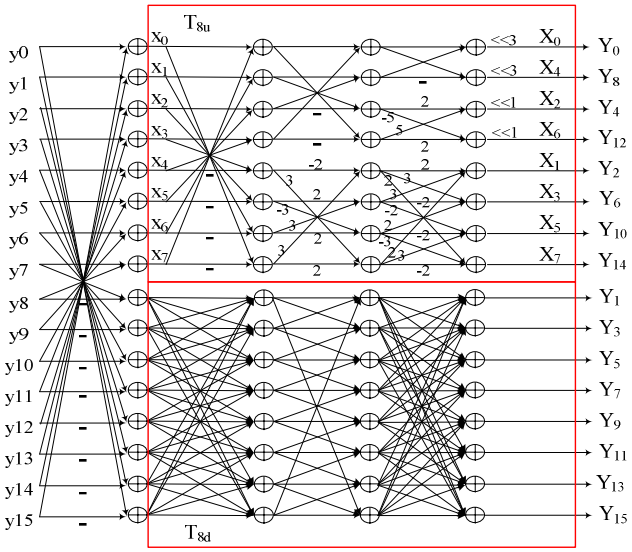

Fig. 1. The 16×16 forward transform butterfly structure

3.3.2 Compatible Scaling Matrix

As mentioned in Section 2, the process of scaling is for normalization and thus the scaling matrix can be derived by the norms of basis vectors in the corresponding transform matrix [2]. Because the even rows in (6) are obtained by extending the 8 basis vectors in the 8×8 transform matrix using the fifteenth even dyadic symmetry, the relationship between their scaling matrices, $P_{8\times8}$ and $P_{16\times16}$ can be shown in (13). Interested readers may refer to [4] for how the relationship is obtained.

$$P_{8\times8}(i, j) = P_{16\times16}(2i, 2j)\times 4 \tag{13}$$

(13) shows that $P_{16\times16}(2i, 2j)$ is one quarter of $P_{8\times8}(i, j)$ and this difference can be easily ironed out by shifting, which means that $P_{16\times16}(2i, 2j)$ can be used as $P_{8\times8}(i, j)$, if N, the bits of right shifting in (3), for the 8×8 scaling is 2 bits less than that for the 16×16 scaling. Therefore, the compatibility of scaling matrix is achieved and the memory specially for storing an 8×8 scaling matrix can be saved.

4 Integration of the 16×16 ICT into the AVS Zengqiang Profile

4.1 Transform Size Selection

Luminance components in a Macroblock (MB) can be transformed by one 16×16 transform, or alternatively by four 8×8 transforms. The selection of the optimum transform size is performed using a criterion of rate-distortion (R-D) cost that can be calculated by (14) with a proper Lagrange multiplierλ[5]. In (14), the SSD(MB) means the summation of square difference between the original and reconstructed MB and bits(MB) is the total bits used to code the MB. The two block-size transforms are tried one by one and their R-D costs are calculated respectively. The one with the lower cost is selected.

$$\text{cost} = \text{SSD(MB)} + \lambda\text{bits(MB)} \tag{14}$$

With this selection criterion, the best R-D performance can be achieved, but a 1-bit binary signal should be transmitted in every MB header to indicate which transform is used.

4.2 16×16 Intra Prediction

The largest block size for intra prediction in the current AVS Zengqiang Profile is 8×8, and to apply the 16×16 ICT to intra-coded MB, the 16×16 intra prediction must be in place. Five prediction directions are used, including DC, horizontal, vertical, down-right and down-left, all of which are very similar to those for 8×8. However, intra prediction with large block size is more likely to introduce visible artifacts, since more pixels are predicted from the same source [6]. So, instead of using the reference pixels directly, a 3-tap low-pass filter [1 2 1]/4 is applied to the reference pixels before they are used, as shown in Figure 2. As for the DC mode, the predicted value is the average of all the available top and left neighboring pixels.

4.3 16×16 Block Entropy Coding

In the AVS Zengqiang Profile, the 8×8 residual blocks are coded by Context-based Adaptive Binary Arithmetic Coding (CABAC) [1]. Its arithmetic coding engine can code sources with various statistics by using different probability models. Therefore, instead of designing new codebooks, an additional set of probability models specially for the 16×16 residual blocks is designed that can be updated adaptively according to the changes of the source statistics. The arithmetic coding engine remains unchanged.

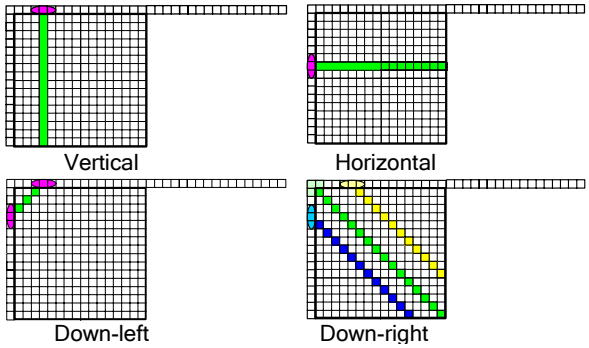

Fig. 2. 16×16 intra prediction modes

5 Experimental Results

The proposed 16×16 ICT is integrated into the RM62b platform which is the latest version of the reference software of the AVS Zengqiang Profile. For every MB, it is used adaptively as an alternative to the 8×8 ICT. Extensive experiments have been done and the test conditions are listed in Table 2. Both the forward and inverse ICT are implemented with 16-bit simple integer arithmetic as described in Section 3.

The performance improvement of coding different HD sequences is shown in Table 3, which can be represented by PSNR gain for equal bit rate in the second column or by saved bit rate in terms of the same PSNR in the third column, using the proposed method in [7]. It is obvious that the coding efficiency is significantly improved

Table 2. Test conditions

Sequence Structure	IBBPBBP….
Intra Period	0.5 second
FME	ON
Deblocking Filter	ON
R-D Optimization	ON
QP	Fixed(22, 28, 34, 40)
Rate Control	OFF
Interlace Handling	PAFF
Reference Frame	2
Search Range	±32
Frame Rate	60 FPS (Progressive)
	30 FPS (Interlace)
Resolution	1280x720 (Progressive),
	1920x1088 (Interlace)

since none of them is smaller than 0.1 dB. Whilst the average gain is almost 0.2 dB, for the best case of Kayak, the gain is up to 0.39 dB. From the fourth column of Table 3 which gives the percentage of MBs coded by the 16×16 ICT, we can safely conclude that the 16×16 ICT is very useful in HD video coding because more than half of the MBs are coded by it. The only case where the 16×16 ICT usage is less than 50% is the Fireworks which is full of low-correlated details.

Table 3. Experimental results

Test sequence	PSNR gain (dB)	Bit rate saved (%)	MB using 16×16 (%)
City	0.123	-4.39	72.02
Crew	0.219	-9.81	73.07
Fireworks	0.162	-2.19	45.69
Flamingo	0.120	-2.26	50.05
Kayak	0.389	-5.75	66.41
Riverbed	0.235	-4.77	80.88
ShuttleStart	0.163	-6.87	75.34
Optis	0.136	-5.21	70.58
Average	**0.193**	**-5.16**	**66.76**

6 Conclusion

In this paper, we propose a 16×16 ICT, specially designed for HD video coding, which is efficient and simple to implement. This 16×16 ICT is integrated into the AVS Zengqiang Profile and used adaptively as an alternative to the 8×8 ICT. The experimental results show that with the 16×16 ICT, the coding efficiency is greatly improved with the average gain around 0.2 dB. Hence, it can be concluded that the 16×16 ICT is a useful coding tool for HD video coding.

References

1. AVS Video Group, “Information technology – Advanced coding of audio and video – Part 2: Video (AVS-X WD 3.0)”, AVS Doc. AVS-N1242, Dec. 2005.
2. Cham, W.K., “Development of integer cosine transforms by the principle of dyadic symmetry”, Proc. IEE, I, 136,(4), pp. 276-282, 1989.
3. Ci-Xun Zhang, Jian Lou, Lu Yu, Jie Dong, W. K. Cham “The Technique of Pre-Scaled Integer Transform in Hybrid Video Coding”, IEEE ISCAS, 2005. Vol. 1, pp. 316-319, May 2005.
4. Jie Dong, Jian Lou, Ci-Xun Zhang and Lu Yu, “A new approach to compatible adaptive block-size transforms”, Visual Communication and Image Processing, 2005, Vol. 5960, pp. 38-47, July 2005.
5. T. Wiegand and B. Girod, “Lagrangian multiplier selection in hybrid video coder control”, in Proc. ICIP 2001, Thessaloniki, Greece, Oct. 2001.
6. Mathias Wien, “Variable Block-Size Transforms for H.264/AVC”, IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 560–576, July 2003.
7. G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, ITU-T SG16 Doc. VCEG-M33, 2001.