Analysis of Integer Transform In MPEG-4 Video Standard

Dwivedi Abhishek N.[1] Darshna D. Jagiwala[2] Supriya S. Surwade[3] Chintan H Desai[4]
[1]M.Tech (Purs.) CGPIT, Maliba Campus, UTU
[2]M.Tech (Purs.),SVNIT,Surat
[3]M.E. (Purs.), JSPM's BSIOTR (W), Wagholi, Pune
[4]M.Tech, Manipal University, Karnataka, India

Abstract— In this paper, Integer cosine transform (ICT) is introduced in MPEG-4 instead of Discrete cosine transform (DCT. Integer cosine transform (ICT) is adopted by H.264/AVC for its bit-exact implementation and significant complexity reduction. As adoption of the newly standardized H.264 becomes increasingly more widespread, efforts must be made to transcode video from earlier standards, such as MPEG-4, to the H.264 format. The block sizes of most appropriate are 8 or 16 for the transform coding of the image data. Therefore, implementation of the order 8 and 16 DCTs has fast computing time and cost-effectiveness for realization of a transform coding. However, the components of the basis vectors of the DCT exist irrational numbers then cannot be reduced to integers by simple scaling. Therefore, it is hard to implement and using floating point arithmetic is complex and expensive, so integer cosine transforms (ICTs) are proposed to implement the DCT by using simple integer arithmetic. By investigating the structure of the transform kernels, two efficient schemes are introduced to convert an 8 × 8 DCT block into its corresponding four 4 × 4 integer cosine transform blocks. This technique can be used to improve the result of MPEG-4 video compression standard.

Keywords: Complexity, Discrete Cosine Transform (DCT), Integer Discrete Cosine Transform (IDCT), MPEG-4, Video Processing.

I. INTRODUCTION

The ISO SC29 WG11 "Moving Picture Experts Group" (MPEG), within ISO SG 29 responsible for "coding of moving pictures and audio," was established in 1988 [1]. the early 1990s, when the technology was in its infancy, international video coding standards chronologically, H.261, MPEG-1, MPEG-2 / H.262, H.263, and MPEG-4 (Part 2) [2-6] have been the engines behind the commercial success of digital video compression. In August 1993, the MPEG group released the so-called MPEG-1 standard for "coding of moving pictures and associated audio at up to about 1.5 Mb/s" [7], [8]. In 1990, MPEG started the so-called MPEG-2 standardization phase [8]. While the MPEG-1standard was mainly targeted for CD-ROM applications, the MPEG-2 standard addresses substantially higher quality for audio and video with video bit rates between 2 Mb/s and 30 Mb/s, primarily focusing on requirements for digital TV and HDTV applications.

The MPEG group officially initiated a new MPEG-4 standardization phase in 1994 with the mandate to standardize algorithms for audio-visual coding in multimedia applications, allowing for inter-activity, high compression, and/or universal accessibility and portability of audio and video content. Bit rates targeted for the video standard are between 5-64 kb/s for mobile applications and up to 2 Mb/s for TV/film applications.

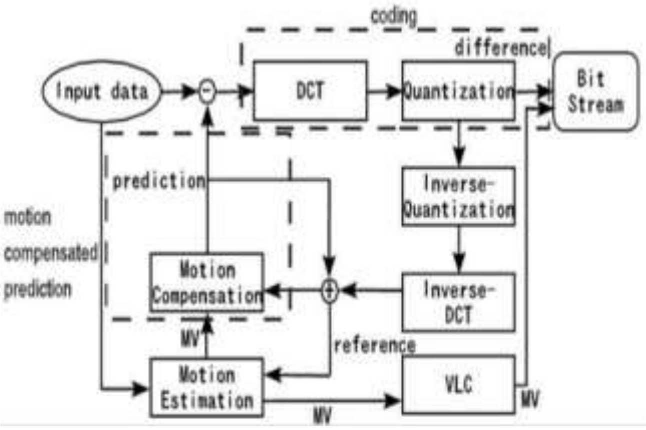

Fig. 1: Block Diagram of MPEG-4

The block diagram of MPEG-4 is shown in figure-1.It is Visual and offers the potential for better compression efficiency (i.e. better-quality compressed video) and greater flexibility in compressing, transmitting and storing video.

It consist of DCT, IDCT, Motion compensation, Motion estimation, quantization, Inverse quantization, VLC, Bit stream. Motion compensation used to remove the temporal redundancy in the video signal. Motion estimation at encoder involves estimating the motion in efficient way so that motion compensation becomes efficient in removing the temporal redundancy. DCT is used in transform to compress the spatial data in information. IDCT is inverse DCT transform to the decoder [1]. The Discrete Cosine Transform, commonly known as DCT , represents a set of data point a as a sum of sinusoidal waves with varying frequencies and magnitudes.

The integer discrete cosine transform (Int DCT) is an integer approximation of the discrete cosine transform. It can be implemented exclusively with integer arithmetic. It proves to be highly advantageous in cost and speed for hardware implementations. In particular, transforms of sizes larger than 4x4 or 8x8, especially 16x16 and 32x32 are proposed because of their increased applicability to the de-correlation of high resolution video signals. For example, order-16 integer transform is simple, low computational complexity transform but has high coding efficiency.

II. BASIC IDEA OF DCT AND INTEGER DCT

The discrete cosine transform (DCT) has long been used in video compression standards such as MPEG-4 for its superior energy compaction abilities. However, recently integer versions of the DCT, as adopted in H.264/AVC, have gained popularity due to their mitigation of the propagation error in video codecs caused by the truncation errors in the original DCT. In particular, we propose two methods for efficient conversion from an 8×8 DCT block to the corresponding four 4×4 integer transform blocks. The second method is further extended to eliminate all multiplications, and two additional conversion schemes are proposed. These four conversion methods then can be utilized in system for rapid transcoding of intra-coded blocks in the transform domain. A direct way to approach the DCT to integer cosine transform conversion challenge is to perform an inverse 8×8 DCT to reconstruct the original signal block, and then perform the 4×4 integer transform on each sub block. Even with the development of a fast integer discrete cosine transform algorithm [9], [10], the computational complexity of direct conversion is significant, and a more efficient method is desired for real-time transcoding. Recently in [11]-[12], the structural properties of the DCT and integer cosine transform kernels have been exploited to produce fast algorithms, resulting in elegant and efficient conversion solutions., it is shown in this work that further examination of the DCT structure results in an even larger reduction in the number of computations at the expense of little or no reduction in PSNR of the transformed and quantized data..

A. *Review Stage*

In this stage of we can show forward transform in fig.2.

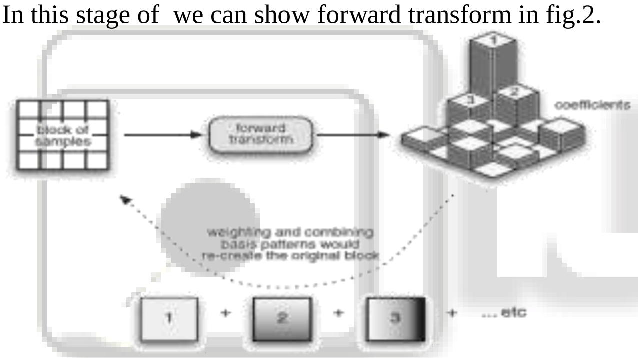

Fig. 2: Forward Transform

B. *Development from the 4x4 DCT*

The 4x4 DCT of an input array X is given by:

$$\mathbf{Y} = \mathbf{AXA}^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \mathbf{X} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

Equation 1

Where, $a = \frac{1}{2}$, $b=\sqrt{\frac{1}{2}}\cos(\pi/8)$, $c=\sqrt{\frac{1}{2}}\cos(3\pi/8)$

This matrix multiplication can be factorized to the following equivalent form (Equation 2)

$$\mathbf{Y} = \left(\mathbf{CXC}^T\right) \otimes \mathbf{E} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \mathbf{X} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix}\right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

Equation 2

CXC^T is a "core" 2-D transform. E is a matrix of scaling factors and the symbol $\otimes$ indicates that each element of (CXC^T) is multiplied by the scaling factor in the same position in matrix E (scalar multiplication rather than matrix multiplication). The constants aandb are as before; d is c/b (approximately 0.414).

To simplify the implementation of the transform, d is approximated by 0.5. To ensure that the transform remains orthogonal, balso needs to be modified so that:

$a = \frac{1}{2}$, $b=\sqrt{\frac{2}{5}}$, $d=\frac{1}{2}$

The 2nd and 4th rows of matrix C and the 2nd and 4th columns of matrix$C^T$are scaled by a factor of 2and the post-scaling matrix E is scaled down to compensate. (This avoids multiplications by ½ in the "core" transform CXC^Twhich would result in loss of accuracy using integer arithmetic). The final forward transform becomes:

$$\mathbf{Y} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T \otimes \mathbf{E}_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \mathbf{X} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}\right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

Equation 3

This transform is an approximation to the 4x4 DCT. Because of the change to factors d and b, the output of the new transform will not be identical to the 4x4 DCT.

C The inverse transform is given by

$$\mathbf{X}' = \mathbf{C}_i^T (\mathbf{Y} \otimes \mathbf{E}_i) \mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left(\left[\mathbf{Y}\right] \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

Equation 4

This time, Y is pre-scaled by multiplying each coefficient by the appropriate weighting factor from matrix Ei. Note the factors +/-½ in the matrices C and C^T; these can be implemented by a right-shift without a significant loss of accuracy because the coefficients Y are pre-scaled .The forward and inverse transforms are orthogonal, i.e. $T^{-1}(T(X)) = X$.DCT usually results in a matrix in which the lower frequencies appear at the top left corner of the matrix known as dc coefficient. The output of a 2-dimensional FDCT is a set of N × N coefficients representing the image block data in the DCT domain which can be considered as 'weights' of a set of standard basis patterns. The basis patterns are composed of combinations of horizontal and vertical cosine functions. Any image block may be reconstructed by combining all N × N basis patterns; with each basis multiplied by the appropriate weighting factor (coefficient).DCT transforms the data from the spatial domain to the frequency domain. The spatial domain shows the amplitude of the color as you move through space as shown in Figure2.The frequency domain shows how quickly the amplitude of the color is changing from one pixel to the next in an image file.

III. COMPRESSIONBETWEEN DCT AND INTEGER DCT

The transform is based on the DCT but with some fundamental differences:

1. It is an integer transform (all operations can be carried out with integer arithmetic, without loss of accuracy).
2. The inverse transform is fully specified in the H.264 standard and if this specification is followed correctly, mismatch between encoders and decoders should not occur.
3. The core part of the transform is multiply-free, i.e. it only requires additions and shifts.
4. A scaling multiplication (part of the complete transform) is integrated into the quantizer (reducing the total number of multiplications).

The entire process of transform and quantization can be carried out using 16-bit integer arithmetic and only a single multiply per coefficient, without any loss of accuracy. IDCT transform can be skipped to reduce the computation and it should be applied is calculated and corresponding reduced complexity. This results in considerable decrease without any quality loss. If the better performance is demanded, the large magnitudes can be used for the integer transforms. If the fast computation speed and the low implementation cost are desired, the integer transforms of small transform component magnitudes are chosen. An engineer can freely choose to tradeoff the performance and speed for the ICTs in designing the transform coding.

IV. CONCLUSION

In this paper, the motto and techniques of the MPEG-4 video standardization has been highlighted and also we discuss about how the higher coding gain is obtained with the use of larger transforms especially in high resolution videos. However, the components of the basis vectors of the DCT exist irrational numbers then cannot be reduced to integers by simple scaling. Therefore, it is hard to implement and using floating point arithmetic is complex and expensive. The DCT-based systems have huge advantage to image applications because they provide a high compression ratio. However, their coding systems are limited to operating in only lossy coding because distortion of decoded image is unavoidable with these lossy algorithms. On the other hand, the integer transform, is becoming popular as a key technique to lossless and lossy unified waveform coding. Especially the integer DCT is attractive as the unified coding comparable to the conventional DCT-based algorithms. IDCT has its bit-exact implementation and significant complexity reduction. Hence we have introduced the method how to use the ICT to achieve the better improved result in MPEG-4.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their useful comments to improve this paper.

REFERENCES

[1] L. Chiariglione, “The development of an integrated audiovisual coding standard: MPEG,” Proc. IEEE, vol. 83, pp. 151-157, Feb. 1995.
[2] Video Codec for Audiovisual Services at p X64 kbit/s, ITU-T Rec. H.261, ITU-T, Version 1: Nov. 1990, Version 2: Mar. 1993.
[3] Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video, ISO/IEC 11172-2 (MPEG-1 Video), ISO/IEC JTC 1, Mar. 1993.
[4] Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.
[5] Video Coding for Low Bit Rate communication, ITU-T Rec. H.263,ITU-T, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov.2000.
[6] Coding of audio-visual objects—Part 2: Visual, ISO/IEC 14492-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3: May 2004.
[7] D. J. Le Gall, “The MPEG video compression algorithm,”Signal Processing: Image Commun., 1992, vol. 4, no. 4, pp.129-140
[8] R. Schäfer and T. Sikora, “Digital video coding standards and their role in video communications,” Proc. IEEE, vol. 83, pp. 907-924, June 1995.
[9] C.-P. Fan, “Fast algorithm designs for low-complexity 4 4 discrete cosine transform,” IEICE Trans. Fundamentals, vol. E88-A, no. 11, pp. 3225-3229, Nov. 2005.
[10] V. Britanak, P. Yip, and K. R. Rao, Discrete Cosine and Sine Trans-forms: General Properties, Fast Algorithms and Integer Approximations. New York: Academic, 2007.
[11] J. Xin, A. Vetro, and H. Sun, “Converting DCT coefficients to H.264/AVC transform coefficients,” in IEEE Pacific-Rim Conf. Multimedia (PCM), Nov. 2004, vol. 3332, pp. 939-946.
[12] Y.-J. Chuang, Y.-C.Huang, and J.-L. Wu, “An efficient algorithm for splitting and 8 8 DCT into four 4 4 modified DCTs used in AVC/H.264,” in Proc. 5th EURASIP Conf. Focused Speech Image Process., MultimediaCommun. Services (EC-SIP-M’2005), Jun. 2005, pp. 311-316.

