A Fast Coefficients Convertion Method for the Transform Domain MPEG-2 to H.264 Transcoding

Gao Chen
Institute of Computing Technology, Chinese Academy of Sciences
chengao@ict.ac.cn

Shouxun Lin
Institute of Computing Technology, Chinese Academy of Sciences
sxlin@ict.ac.cn

Yongdong Zhang
Institute of Computing Technology, Chinese Academy of Sciences
zhyd@ict.ac.cn

Abstract

Converting MPEG-2 8-tap discrete cosine transform (DCT) coefficients to H.264/AVC 4-tap integer transform (IT) coefficients is one of the indispensable operations for transcoding MPEG-2 to H.264 in the transform domain. Motion information in the input MPEG-2 sequence should be reused as much as possible to speed up the transcoding process. Motivated by this, we first exploits the distribution information of non-zero DCT coefficients to classify the input 8×8 DCT block into three types, then proposes a fast method to perform coefficients conversion based on the type of 8×8 DCT block. The simulation results show that the proposed method significantly reduces the computational complexity, while maintains almost the same video quality compared with the existing method. Hence, it can be efficiently used in the real time transform domain MPEG-2 to H.264 transcoding.

1. Introduction

The newest video-coding standard, known as H.264/AVC [1], jointly developed by the Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, is highly efficient offering perceptually equivalent quality video at about 1/3 to 1/2 of the bit-rates offered by the MPEG-2 format [2]. Due to its superior compression efficiency, it is expected to replace MPEG-2 over the next several years, but the complete migration to H.264 will take several years given the fact that MPEG-2 has been widely used in many applications nowadays. This creates an important need for transcoding technologies that transcode the wildly available MPEG-2 compressed videos to H.264 compressed format and vice versa [3][4].

The transform domain transcoding is simpler than the one in the conventional pixel domain, since the former avoids the complete decoding and re-encoding which are computationally expensive. For this reason, there has been a great effort in recent time to develop fast algorithms that conduct MPEG-2 to H.264 transcoding in transform domain [5][6]. Unlike the other previous transform domain transcoding, such as H.263 to MPEG-2 transcoding, the decoded DCT coefficients in the MPEG-2 to H.264 transcoder can not be reused directly and have to be converted to IT coefficients. The reason is that MPEG-2 uses 8-tap DCT to produce the transform coefficients, while H.264 uses a 4-tap IT to do so. As a result, one of the indispensable steps in the transform domain MPEG-2 to H.264 transcoding is to convert MPEG-2 DCT coefficients to IT coefficients, i.e. DCT-to-IT conversion.

The role of DCT-to-IT conversion in MPEG-2 to H.264 transcoder is equal to the transform, such as DCT, in video encoder. There have been many fast DCT algorithms proposed to implement the video encoder efficiently. In order to implement the MPEG-2 to H.264 transcoder efficiently in the transform domain, we should to speed up the process of DCT-to-IT conversion. In this paper, we focus on speeding up this process by taking advantage of the property that most of the 8x8 DCT blocks in real video sequences only have nonzero coefficients in the upper left 4x4 quadrant. The rest of the paper is organized as follows. Section 2 reviews the existing methods. A fast method for the DCT-to-IT conversion is presented in Section 3. Simulation results are given in Section 4, and conclusions are described in Section 5.

2. Overview of the existing methods

The conventional MPEG-2 to H.264 pixel domain transcoder first needs to decode an input MPEG-2 video to pixel level, and then encode decoded image in the H.264 format. So, the DCT coefficients are first converted to pixels data through inverse DCT and then transformed to IT coefficients through IT, i.e. cascading an inverse 8-tap DCT with four 4-tap IT (this process is called pixel domain approach thereinafter). Obviously, this method cannot be adopted in the transform domain MPEG-2 to H.264 transcoder, since the input MPEG-2 video sequence is already decoded to the pixel data.

Through manipulating the pixel domain coefficients conversion operations in the matrix multiplication form, Jun xin et al. [7] and Bo shen [8] have derived two different transform kernel matrices to perform DCT-to-IT conversion directly in transform domain. After comparing, we find that there are more 64 operations and 2 computing stages in Bo shen's method than Jun xin's to perform the DCT-to-IT conversion. In this paper, Jun xin's matrix is selected as the foundation to simplify the calculating process instead of Bo shen's.

3. Fast method for DCT-to-IT conversion

Since the energy distribution of DCT blocks obtained from the real MPEG-2 video sequences mainly concentrates on the low frequency region, it is beneficial to exploit this property to speed up the DCT-to-IT conversion. The detail of our proposed fast method is described in the following.

3.1. Classifying the 8×8 DCT blocks

Usually, the MPEG-2 to H.264 transcoder needs to perform an entropy decoding to extract the motion vectors, header information, coding mode and quantized DCT coefficients from the input MPEG-2 video bit stream. Particularly, the raster scan order of each non-zero DCT coefficients within an 8×8 block is also available to the transcoder. With such information, we can classify the MPEG-2 8×8 DCT blocks into three types as following.

1. **Low pass block**

If the non-zero coefficients of an 8×8 DCT block are only located in the upper left 4×4 quadrant, this DCT block is denoted as a low pass block.

2. **Zero block**

The zero block is an 8 × 8 DCT block whose coefficients are all equal to zero. For example, all 8×8 DCT blocks in a skipped macroblock are indicated as zero blocks.

3. **Normal block**

All other 8×8 DCT blocks are marked as normal block.

3.2. Simplifying Jun xin`s calculating process for low pass blocks

Jun xin's method [7] can be explicitly used for the DCT-to-IT conversion of low pass block. However, it is not efficient since it does not utilize the coefficients distribution property of the low pass block. In the following, we propose a simplified calculating process to process the DCT coefficients in a low pass block.

We first discuss the one dimension case. The two-dimensional case will be a repeated application for very row and then for very column of an 8×8 DCT block. Let z be an 8-dimensional column vector, and a vector Z be the 1D transform of z. Considering that the elements z[5]~z[8] equal to zero, the calculating steps described in [7] can be simplified as:

$$
\begin{aligned}
m1 &= a\times z[1] \\
m2 &= b\times z[2] - c\times z[4] \\
m3 &= g\times z[3] \\
m4 &= f\times z[2] + h\times z[4] \\
m6 &= -l\times z[2] + m\times z[4] \\
m7 &= j\times z[3] \\
m8 &= p\times z[2] - q\times z[4]
\end{aligned}
$$

$$
\begin{aligned}
Z[1] &= m1 + m2 \\
Z[5] &= m1 - m2 \\
Z[2] &= m3 + m4 \\
Z[6] &= m4 - m3 \\
Z[3] &= m6 \\
Z[7] &= -m6 \\
Z[4] &= m7 - m8 \\
Z[8] &= m7 + m8
\end{aligned}
$$

Where the const values a…. q have the same value in [7]. The corresponding flow-chart is depicted in Figure 1.

The simplified calculating process totally needs only 11 multiplications and 11 additions. Furthermore, considering that the 2D transform of low pass block needs only four 1D column transforms and eight 1D row transforms since that the last 4 column vectors in the low pass block are all zero vector. So, It follows that two-dimension case needs 132 (= 12 × 11) multiplications and 132 (=12×11) additions, for total of 264 operations. Thus, the simplified calculating process saves 62.5% of operations (i.e. 440 operations) than using Jun xin's method directly. This simplified calculating process is referred as simplified S-transform method hereafter.

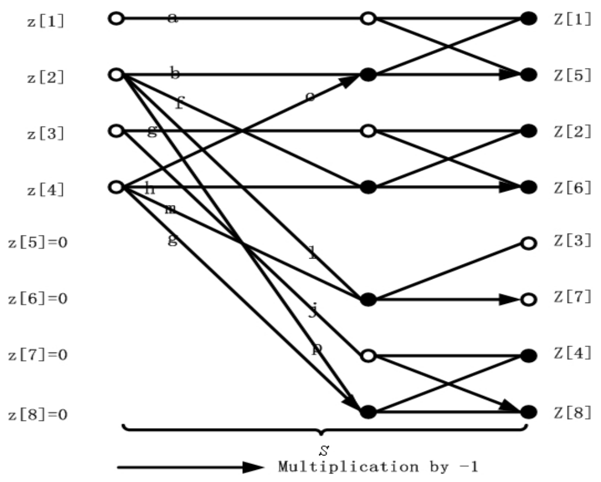

Figure 1. Flow-chart for calculating low pass block

3.3. Fast method for DCT-to-IT conversion

Finally, in order to reduce the computation complexity as much as possible, we propose a fast method that operates in two steps as following.

Step 1. The decoded MPEG-2 DCT 8×8 blocks are first classified into three types as mentioned in subsection 3.1.

Step 2. We adopt different algorithm to perform DCT-to-IT conversion according to the type of DCT 8×8 block. That is the Jun xin's method is used for normal blocks and our proposed simplified S-transform method is used for low pass blocks. Obviously, the zero blocks does not need to process because their corresponding IT coefficients are all zero.

3.4. Complexity comparison

Note that only the non-zero DCT coefficients are encoded in MPEG-2 bit-stream. The computation used to judge the type of block is very small and can be ignored when compared with the computation of DCT-to-IT conversion. The operations and calculating stages in different methods for DCT-to-IT conversion are tabulated in Table 1.

In the worst case, where all DCT blocks are not low pass blocks, our proposed method is the same as Jun xin's method, and in the best case, where all 8×8 DCT blocks are low pass blocks, our proposed method can saves 440 operations for each 8×8 DCT block compared with Jun xin's method. Obviously, the practical reduction of computation is highly depending on the percentage of low pass blocks in MPEG-2 video sequences. The distortion and speed performance of the proposed method are quantitatively evaluated in next section.

Table 1. DCT-to-IT conversion operations

Method	Pixel domain	Bo shen	Jun xin	Simplified S-transform
Add	672	352	352	132
Mul	256	352	352	132
Shift	64	64	0	0
Sum of operations	992	768	704	264
Calculating Stages	6	4	2	2

4. Experimental results

4.1. Simulation conditions

In order to evaluate our proposed method, we adopt MPEG Software Simulation Group (MSSG) MPEG-2 software decoder [9] to decode the input MPEG-2 test videos. A flag, i.e., low pass block flag, which denotes whether an 8×8 DCT block is a low pass block or not, is adopted in our simulations. The decoded 8×8 DCT block (**X**) and the associated block type information (e.g. low pass block flag) are first sent to three processing systems, which adopt the pixel domain method, the Jun xin's method, and our proposed method respectively. Each of the three processing systems converts **X** to the IT blocks. In order to avoid the influences of H.264 coding tools, such as intra prediction and variable block size motion compensation, the IT coefficients are directly subjected to H.264 quantization, inverse quantization and inverse IT process, which are the same processes as in the reference software H.264/AVC JM8.2 [10], to get the reconstructed pixels data. The H.264 re-quantization QP factors ranging from 0 to 51, corresponding to the full H.264 QP range. The block diagram of simulation setting is shown in Figure 2.

The first 90 frames of each of test sequences in CIF resolution (352 × 288) are pre-coded to different MPEG-2 test videos at 2.5Mbps/s, 3Mbps/s, 3.5Mbps/s and 4Mbps/s respectively. These MPEG-2 test videos are all intra-coded at a frame rate of 30 fps. All simulations are performed using Windows XP, Intel P4 2.8GHz CPU and 1GB memories, Visual C++ 6.0 Compiler. The performance is measured with the mean

Peak Signal to Noise Ratio (PSNR) between the reconstructed frame sequence and the corresponding original uncompressed frame sequence. The computational complexity is measured with the mean runtime of the all DCT block in one frame needed to convert to IT coefficients. Furthermore, the mean runtime is obtained from the average value of three repeated simulations.

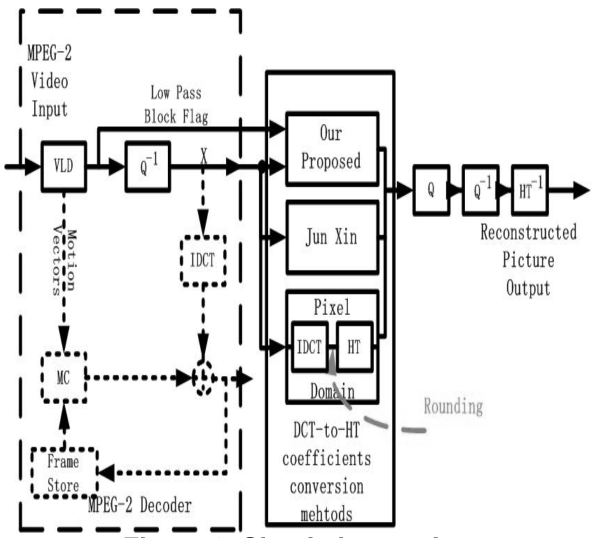

Figure 2. Simulation setting

Extensive simulations and performance comparison have been done with different motion characteristic sequences. Due to the limit of pages, we only show the results of FOREMAN and the results of other sequences are omitted.

4.2. Our proposed method vs. pixel domain method

Figure 3 shows the PSNR difference between our proposed method and pixel domain method. Just as the Jun xin's method, our proposed method outperforms the pixel domain method. The superior quality of our proposed method is owing to that the rounding operation is taken place for the result IT coefficients. On the contrary, for pixel domain method, the saturation and mismatch control (the standard MPEG-2 decoding steps following inverse DCT to reconstruct pixel data), which act as the rounding operation, are taken place for the intermediate results.

4.3. Our proposed method vs. Jun xin's method

Figure 4 shows the PSNR difference between our proposed method and Jun xin's method. The difference is nearly equal to zero, with the maximum difference less than 8×10^{-4}dB. This minuscular difference is resulted from that the value zero of the 64th DCT coefficient in one low pass block is usually set to value one after the standard decoding operations: saturation and mismatch control, but this block is still considered as low pass block and the value of the 64th coefficient is ignored in our proposed simplified S-transform method.

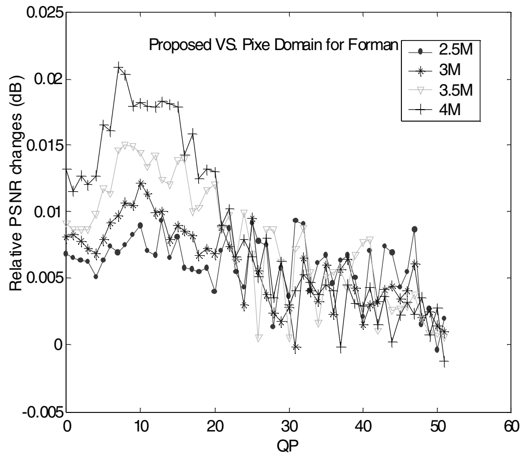

Figure 3. Relative PSNR difference of our proposed method vs. pixel domain method for FOREMAN

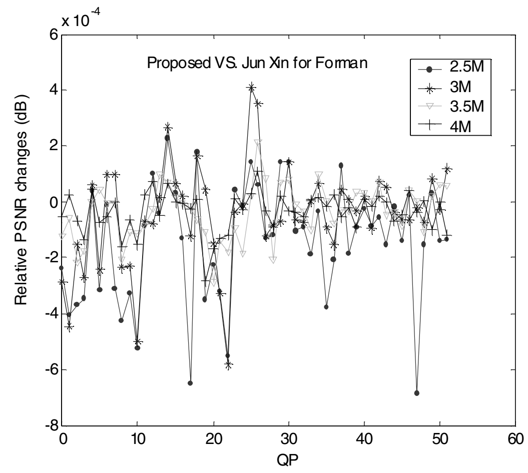

Figure 4. Relative PSNR difference of our proposed method vs. Jun xin's method for FOREMAN

4.4. Complexity comparison

As expected, the lower the bit-rate is, the more of the low pass blocks are in video sequences. In order to meet the requirement of target bit-rate, the video encoder needs to adopt high quantization parameter, which may cause more DCT coefficients to be quantized to zero. In our simulations, the overall percentages of low pass block for FOREMAN sequence are 72.8%, 63.8%, 55.6% and 48.4% corresponding to the bit rate 2.5 Mb/s, 3 Mb/s, 3.5 Mb/s and 4 Mb/s respectively.

Table 2. Runtime for DCT-to-IT conversion

Test sequence	Bit-rate (Mb/s)	Runtime for DCT-to-IT conversion (ms)		
		Pixel domain	Jun xin	Our method
FORE MAN	2.5	12.88	12.10	7.76 (35.9%)
	3	12.88	12.05	8.32 (31.0%)
	3.5	12.90	12.26	8.95 (27.0%)
	4	12.80	12.31	9.40 (23.6%)

Table 2. tabulates the actual runtime requirements (in terms of microseconds used) for each method at different input bit-rates. The relative improvements of our proposed method compared with Jun xin's approach are shown inside the parenthesis for the ease of comparison. Our proposed method achieves the lowest runtime and saves about 23.6%-35.9% computation compared with Jun xin's method. Obviously, the complexity reduction is proportion with the percentage of low pass blocks.

5. Conclusion

In this paper, we derive a fast method to reduce the computational complexity of the DCT-to-IT conversion. We analyze the operations involved in our proposed fast method. The efficiency of our proposed fast method is also quantitatively evaluated. Our simulation results show that the proposed method leads to about 23.6%–35.9% saves in computational complexity, while obtains almost negligible PSNR differences (with the maximum value less than 8×10^{-4}dB) compared with Jun xin's method. In our simulations, only the intra-coded MB is used to test. Considering that the DCT coefficients decoded from inter-coded MBs of MPEG-2 video are computed from the motion compensated residual, there would be more low pass block in inter-coded frames than in intra-coded frames. So, we can say that our proposed method would be more efficient in the inter-coded frames transcoding.

6. Acknowledgement

This work is supported by National Nature Science Foundation of China under grant number 60302028.

7. References

[1] Thomas Wiegand, Gary Sullivan. "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITUT Rec. H.264 | ISO/IEC 14496-10 AVC)," JVT-G050, Pattaya, Thailand, Mar. 2003.
[2] ISO/IEC JTC11/SC29/WG11, "Generic Coding of Moving Pictures and Associated Audio Information: Video", ISO/IEC 13818-2. May 1994.A. N. Netravali and B. G. Haskell, Digital Pictures, 2nd ed., Plenum Press: New York, 1995, pp. 613-651.
[3] A. Vetro, C. Christopoulos, and H.Sun. "Video Transcoding Architectures and Techniques: An Overiew". IEEE Signal Processing Magazine, Vol 20, no.2, pp.18-29, March. 2003.
[4] Hari Kalva. "Issues in H.264/MPEG-2 Video Transcoding". CCNC 2004. First IEEE, 5-8 Jan. 2004.
[5] Hari Kalva, Branko Petljanski, and Borko Furht. "Complexity Reduction Tools for MPEG-2 to H.264 Video Transcoding." WSEAS Transactions on Information Science & Applications, Vol. 2, Issues, Marc. 2005, pp. 295-300.
[6] Chen Chen, Ping-Hao Wu, and Chen, H, "Transform-domain intra prediction for H.264," Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on 23-26 May 2005 Page(s):1497-1500 Vol 2
[7] J. Xin, A. Vetro and H.Sun, "Converting DCT coefficients to H.264/AVC transform coefficients," IEEE Pacific-Rim Conference on Multimedia (PCM), Lecture Notes in Computer Science, ISSN: 0302-9743, Vol. 3332/2004 pp. 939, November 2004.
[8] Bo Shen, "From 8-tap DCT to 4-tap integer-transform for MPEG to H.264 transcoding," Proc. IEEE International Conf. on Image Processing (ICIP04), Oct. 2004.
[9] MPEG-2 video decodec v12, available online at http://www.mpeg.org/MPEG/MSSG.
[10] H.264/AVC Reference Software JM8.2, available online at http://bs.hhi.de/~suehring/tml/download