LOOK-UP-TABLE BASED DCT DOMAIN INVERSE MOTION COMPENSATION

Shizhong Liu and Alan C. Bovik

Laboratory for Image and Video Engineering, Dept. of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX 78712-1084, USA
Email: {sliu2, bovik}@ece.utexas.edu

ABSTRACT

DCT-based digital video coding standards such as MPEG and H.26x have been widely adopted for multimedia applications. Thus video processing in the DCT domain usually proves to be more ef£cient than in the spatial domain. To directly convert an inter-coded frame into an intra-coded frame in the DCT domain, the problem of *DCT domain inverse motion compensation* was studied in [1]. Since the data is organized block by block in the DCT domain, DCT domain inverse motion compensation is computationally intensive. In this paper, a look-up-table (LUT) based method for DCT domain inverse motion compensation is proposed by modeling the statistical distribution of DCT coef£cients in typical images and video sequences. Compared to the method in [1], the LUT based method can save more than 50% of computing time based on experimental results. The memory requirement of the LUT is about 800 KB which is reasonable. Moreover, the LUT can be shared by multiple DCT domain video processing applications running on the same computer.

1. INTRODUCTION

With the emergence of video compression standards such as MPEG and H.26x, compressed digital video bit streams are widely used for high ef£ciency of storage and transmission. However, the compression schemes based on a combination of Discrete Cosine Transform (DCT) and Motion Compensation (MC) do not lead to easy manipulation and composition of the compressed video. There are two general approaches for processing compressed video bit streams: spatial domain processing and DCT domain processing. In spatial domain methods, the video bit stream is £rst fully decompressed to the spatial domain, then processed in the spatial domain, and £nally re-compressed for storage or transmission. In DCT domain methods, the video bit stream is £rst partially decoded to the DCT domain, then processed in the DCT domain, and £nally re-encoded. DCT domain processing methods prove to be more ef£cient than spatial domain methods [1, 2] due to the following reasons: A) smaller data volume to be processed since DCT blocks are sparse; B) lower computational complexity due to elimination of the process of IDCT-DCT . In DCT domain video processing methods, DCT domain inverse motion compensation is usually used to convert an inter-coded frame into an intra-coded frame for video manipulation and composition. Since the data is organized block by block in the DCT domain, DCT domain inverse motion compensation has high computational complexity.

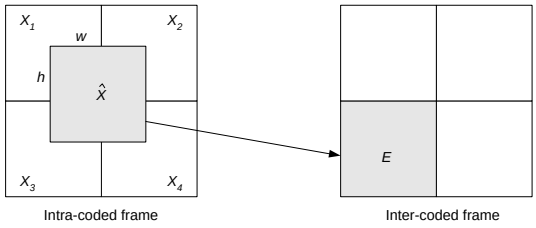

Fig. 1. DCT domain inverse motion compensation.

This problem was studied by Chang *et al.* [1]. The general setup is shown in Fig. 1, where $\hat{x}$ can be expressed as a superposition of the appropriate windowed and shifted versions of x_1, x_2 x_3 and x_4, *i.e.*,

$$\hat{x} = \sum_{i=1}^{4} q_{i1} x_i q_{i2} \tag{1}$$

where $q_{ij}, i = 1, \ldots, 4, j = 1, 2$ are sparse 8×8 matrices of zeros and ones that perform windowing and shifting operations. For example, for $i = 1$,

$$q_{11} = \begin{pmatrix} O & I_h \\ O & O \end{pmatrix} \quad q_{12} = \begin{pmatrix} O & O \\ I_w & O \end{pmatrix} \tag{2}$$

where I_h and I_w are identity matrices of dimension $h \times h$ and $w \times w$, respectively. The values h and w are determined by the motion vector of $\hat{x}$. By using the linear, distributive

This work was supported in part by Texas Instruments, Inc. and by the Texas Advanced Technology Program.

and unitary properties of DCT, it follows that

$$\hat{X} = \sum_{i=1}^{4} Q_{i1} X_i Q_{i2} \tag{3}$$

where $\hat{X}$, X_i, Q_{i1} and Q_{i2} are the DCT's of $\hat{x}$, x_i, q_{i1} and q_{i2}, respectively.

Merhav *et al.* [2] proposed a fast algorithm to compute (3) by factorizing the £xed matrices Q_{ij} into a series of relatively sparse matrices instead of fully pre-computing them. Hence, some of the matrix multiplications can be avoided by simple addition and permutation operations. Assuncao *et al.* [3] approximated the elements of Q_{ij} by binary numbers with a maximum distortion of $1/32$ so that all multiplications can be implemented by *shifts* and *additions*. They showed that in terms of operations *(shift, add)* required, their algorithm has only 28% of the computational complexity of the method proposed by Merhav *et al.* [2]. While all these methods utilize 2-D implementations, Acharya *et al.* [4] decomposed the problem into two separate 1-D problems. This decomposition proves to be more ef£cient than computing the combined operation. In (3), we have $Q_{12} = Q_{32} \stackrel{\text{def}}{=} Q_{x0}$, $Q_{22} = Q_{42} \stackrel{\text{def}}{=} Q_{x1}$ and $Q_{11} = Q_{21} \stackrel{\text{def}}{=} Q_{y0}$, $Q_{31} = Q_{41} \stackrel{\text{def}}{=} Q_{y1}$. By using the separable scheme, (3) can be implemented as follows:

$$G_0 = X_1 Q_{x0} + X_2 Q_{x1} \tag{4}$$

$$G_1 = X_3 Q_{x0} + X_4 Q_{x1} \tag{5}$$

$$\hat{X} = Q_{y0} G_0 + Q_{y1} G_1 \tag{6}$$

where G_0 and G_1 are two intermediate blocks generated by horizontally shifting and windowing $X_i, i = 1, 2, 3, 4$, respectively.

In this paper, a look-up-table (LUT) based method for DCT domain inverse motion compensation is proposed by modeling the statistical distribution of DCT coef£cients in typical images and video sequences. By pre-computing the multiplication results in $(4-6)$ for those DCT coef£cients with absolute value below a certain threshold, the LUT based method can save more than 50% of computing time, relative to Chang's algorithm [1]. The memory requirement of the LUT is about 800 KB which is reasonable based on the current computer memory capacity. Moreover, the LUT can be shared by multiple DCT domain video processing applications running on the same computer.

2. THE LUT BASED METHOD

2.1. Modeling distribution of DCT coef£cients

All DCT coef£cients in MPEG or H.26x coded images are quantized to integers with value ranging from -2048 to 2047. Since the DCT concentrates most of the signal energy into relatively few coef£cients, most AC coef£cients have small values. The distribution of AC coef£cients can be modeled as a Laplacian distribution with zero mean as follows [5, 6]:

$$p(x) = \frac{\lambda}{2} exp(-\lambda|x|) \tag{7}$$

where

$$\lambda = \frac{1}{E[|X|]}. \tag{8}$$

Let σ^2 be the variance of X, then $\sigma = \frac{\sqrt{2}}{\lambda}$. Given a positive threshold TH, one can have

$$\begin{aligned} P(|X| \le TH) &= \int_{-TH}^{TH} p(x)dx \\ &= \int_{0}^{TH} \lambda\, exp(-\lambda x)dx \end{aligned} \tag{9}$$

A few JPEG-coded images and I frames from several MPEG-coded video sequences are selected to estimate the value of λ according to (8). As a result, we obtain $\lambda \approx 0.0284$. If we set a threshold $TH = 2\sigma \approx 100$, then more than 94% of AC coef£cients have absolute value smaller than the threshold TH according to (9). This implies that a lot of computation can be saved by pre-computing the multiplication results for all those coef£cients having absolute value smaller than the threshold TH. In the following, the implementation of the LUT based method will be discussed.

2.2. LUT design

By using the separable approach [4], only 1-D case needs to be considered to build the LUT. So only two tables are needed to save all multiplication results with Q_{x0} and Q_{x1} in (4) and (5). In Fig. 1, w has 16 possible values including half-pixel resolution, *i.e.* $w = 0, 0.5, 1, \ldots, 7.5$. For 1-D case, each non-zero element in $\{X_i\}$ contributes to eight entries of G_0 or G_1. As a result, a four dimensional table is needed to save the pre-computed results, *i.e.* Table[*v*][*p*][*w*][*i*], where *v* represents the absolute value of DCT coef£cients, *p* represents the column position of DCT coef£cients in $\{X_i\}$, *w* is shown in Fig. 1 and *i* represents the column position of the pre-computed results in the target block. Both *p* and *i* have eight possible values each. *v* has 100 possible values since TH is 100. If four bytes are used to store each entry of the table, the size of table is

$$size = 4 \times v \times p \times w \times i = 4 \times 100 \times 8 \times 16 \times 8 = 400KB. \tag{10}$$

Hence, the total memory requirement for two tables is about $800KB$ when $TH = 100$. This is reasonable based on the current computer memory capacity. The vertical operations in (6) can be converted to the horizontal operations via matrix transposition so that the LUT can be reused.

2.3. DC coef£cients

In DCT-coded images, the distribution of DC coef£cients has a large mean value (*e.g.* 1000) and a larger variance relative to the distribution of AC coef£cients as shown in Fig. 2. Since the LUT is created by modeling the distribution of AC coef£cients, it doesn't apply to DC coef£cients because most DC coef£cients are much larger than the threshold TH. For adjacent DCT blocks, the DC coef£cients are highly correlated in typical images [7]. Therefore, the difference between adjacent DC coef£cients should have much smaller mean value and dynamical range than the DC coef£cient itself. With this observation, (4) can be rewritten as

$$\begin{aligned} G_0 &= X_1 Q_{x0} + X_2 Q_{x1} \\ &= \frac{X_1 + X_2}{2}(Q_{x0} + Q_{x1}) + \\ &\quad \frac{X_1 - X_2}{2}(Q_{x0} - Q_{x1}). \end{aligned} \tag{11}$$

Let $Q_+ = Q_{x0} + Q_{x1}$, Q_+ has the property:

$$Q_+(0,0) = 1; Q_+(0,j) = Q_+(j,0) = 0; j = 1, \ldots, 7.$$

This means that the summation of the DC components of X_1 and X_2 only contributes to the DC component of G_0. So we can just sum up both DC coef£cients from X_1 and X_2, then £ll it in the DC entry of G_0 without any further computation. The difference between the DC component of X_1 and that of X_2 has distribution similar to the distribution of AC coef£cients as shown in Fig. 3. In the selected JPEG-coded images, more than 70% of the difference values have absolute value below the threshold TH so that their multiplication results in (11) can be directly obtained from the LUT.

3. EXPERIMENTAL RESULTS

Both Chang's method [1] and the LUT based method were implemented. Four MPEG-coded video sequences with intensive motion activities were selected for our experiments, *i.e.*, "Foreman", "Coastguard", "Mobile" and "Stefan". All sequences are CIF resolution with 352 pixels and 288 lines. All P and B frames in the selected video sequences were converted into I frames by using both methods. In the LUT based method, for those DCT coef£cients with absolute value greater than the threshold TH, Chang's method was employed to compute the corresponding multiplication results. The time for reconstructing one P or B frame to an I frame for both methods was measured on a Windows NT workstation with 300 MHz CPU and 512 MB memory, respectively. The average time for converting one P or B frame into an I frame is tabulated in Table 1. On average, the LUT based method saved the computing time by more than 70% in "Forman", 60% in "Coastguard" and "Stefan" and 50% in "Mobile" according to the results in Table 1. In Fig. 4, the time for reconstructing each P frame to an I frame in the sequence "Mobile" is plotted for both methods. Fig. 5 shows the time for each B frame. As can be seen, the computing time in the LUT based method was almost constant for different P or B frames while the computing time in Chang's method changes dramatically. One reason is that for motion vectors with half-pixel precision, the complexity of Chang's method is doubled due to the interpolation, while it is constant in the LUT method because the results for half-pixel resolution are pre-computed and saved in the LUT.

Video sequence	Chang's method		The LUT based mehtod	
	P frame	B frame	P frame	B frame
"Forman"	0.3137	0.4738	0.0931	0.1423
"Coastguard"	0.2374	0.3417	0.0912	0.1190
"Mobile"	0.3487	0.4136	0.1462	0.2000
"Stefan"	0.2057	0.3667	0.0780	0.1416

Table 1. The average computing time to convert one P or B frame to an I frame (Unit: second)

4. CONCLUSION

In this paper, a LUT based method for DCT domain inverse motion compensation is proposed by modeling the distribution of DCT coef£cients in typical images and video sequences. Relative to Chang's method, more than 50% of computing time can be saved based on the experimental results. Compared to other fast algorithms [2, 3], the LUT based method is straightforward to implement and introduces no error. Moreover, for motion vectors with half-pixel precision, the algorithms in [2, 3] have the same problem as Chang's algorithm, *i.e.*, the computational complexity is doubled. However, the LUT method has no such problem since all corresponding results are pre-computed and saved in the LUT. This can help reduce the jerkiness in video manipulation and composition applications. The memory requirement of the LUT is about 800 KB which is reasonable based on the current computer memory capacity. In addition, the LUT can be shared by multiple video processing applications running on the same machine. For example, in a video editing server, usually multiple processes are running simultaneously to process different video bit streams. Only one LUT is needed for these parallel processes to conduct DCT domain inverse motion compensation.

5. REFERENCES

[1] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE*

J. on Selected Areas in Comm., vol. 13, pp. 1–11, Jan. 1995.

[2] N. Merhav and V. Bhaskaran, "Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 468–476, June 1997.

[3] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 953–967, Dec. 1998.

[4] S. Acharya and B. Smith, "Compressed domain transcoding of MPEG." Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS), Austin, TX, June 1998.

[5] K. A. Birney and T. R. Fischer, "On the modeling of DCT and subband image data for compression," *IEEE Trans. on Image Processing*, vol. 4, pp. 186–193, Feb. 1995.

[6] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coef£cient distributions for images," *IEEE Trans. on Image Processing*, vol. 9, pp. 1661–1666, Oct. 2000.

[7] T. Sikora and H. Li, "Optimal block-overlapping synthesis transforms for coding images and video at very low bitrates," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 157–167, Apr. 1996.

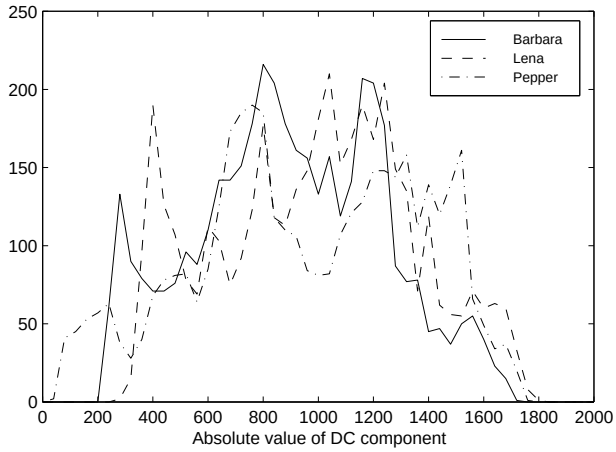

Fig. 2. Histograms of DC coef£cients in images with the Bin size 40.

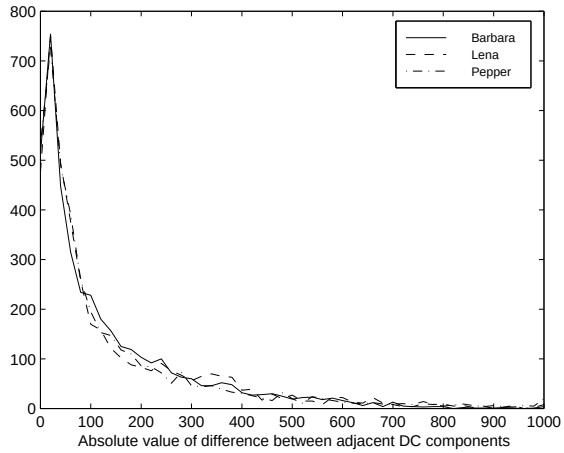

Fig. 3. Histograms of difference between adjacent DC coef£cients in images with the Bin size 20.

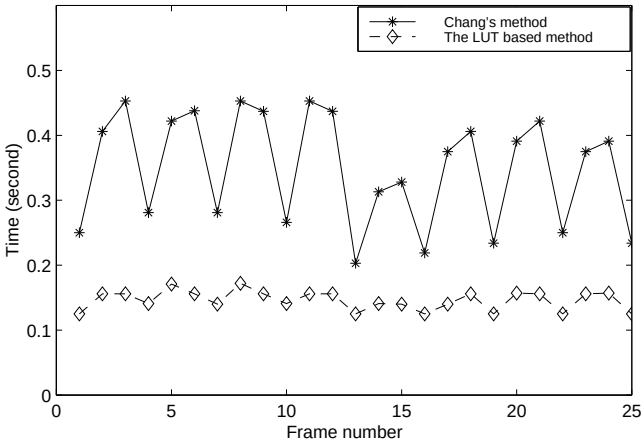

Fig. 4. The computing time for reconstructing each P frame to I frame in the video sequence "Mobile".

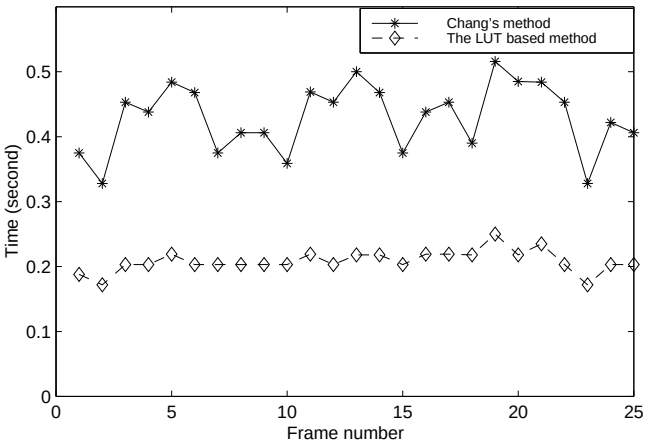

Fig. 5. The computing time for reconstructing each B frame to I frame in the video sequence "Mobile".