Integrated Compressed Domain Resolution Conversion with De-interlacing for DV to MPEG-4 Transcoding

Haruhisa KATO and Yasuyuki NAKAJIMA
KDDI R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-shi Saitama 356-8502, JAPAN
Email: hkato@kddilabs.jp and nakajima@kddilabs.jp

Takashi SANO
Tokyo University of Science
1-3 Kagurazaka Shinjuku-ku Tokyo 162-8601, JAPAN
Email: t-sano@tsm.kddilabs.jp

***Abstract*— In this paper we describe a fast picture size conversion with de-interlacing for Digital Video (DV) to MPEG-4 transcoding. By an integration of matrices operation of resolution conversion with de-interlacing and by exploiting local symmetries in the matrices, very fast conversion can be obtained. The speed up factor of the conversion is up to 1.6 and 5 for multiplication and addition, respectively, when compared with baseband domain conversion. The transcoding experiments also show that the proposed method can achieve almost the same PSNR performance as that of baseband domain conversion.**

I. INTRODUCTION

DV has been widely used as a convenient digital video recorder recently. DV data can be directly transferred without degradation to PC via IEEE 1394. However, since the coding bitrate of DV is set to 25Mbps, DV can be stored only for 25 minutes even with DVD with a capacity of 4.7G bytes. Moreover, the current bandwidth of the Internet is not enough for transmitting DV. For these reasons, DV to MPEG transcoding becomes very important task to obtain high quality video at much lower coding bit rate than DV.

As for transcoding technologies, a number of research works have been conducted in order to achieve fast and high quality transcoding when compared with baseband domain cascaded decoder-encoder system (for review, [1] for example). In the past, we proposed a transcoding algorithm from DV to MPEG-2 [2]. In this algorithm, we proposed a 2*4*8 DCT to 8*8 DCT conversion in compressed domain and fast motion estimation employing a hierarchical 2-step search using DCT information obtained from DV data. Recently, DV to MPEG-4 transcoding including picture size conversion (such as 601 to SIF) also becomes important for portable video and Internet video applications. This paper proposes a fast resolution conversion with de-interlacing method for DV to MPEG-4 transcoding. The proposed method can accelerate integrated conversion with almost degradation-free image quality.

In the following, we firstly investigate conventional conversion methods. Then in Section III, we depict the proposed method followed by its experimental results in Section IV.

II. CONVENTIONAL RESIZING METHOD IN CODED DOMAIN

Picture resizing method in coding data domain has been proposed in [3] by changing the base of DCT and IDCT. In this method, since IDCT is performed using only lower frequency components of DCT coefficients, the computational complexity decreases not only by integrating resolution conversion and IDCT in advance but also by reducing the degree of DCT when compared with baseband domain conversion. Especially, the resolution conversion method to half size in the coded domain [4] extracts the 4*4 lower frequency DCT coefficients from the four neighboring blocks, then 4*4 IDCT is performed to obtain four 4*4 sub-blocks, and they are combined into 8*8 blocks.

The reference [5] interpreted DCT decimation as basis vectors re-sampling, and presented a compressed data domain approach for DCT decimation. It has also proposed fast operations equivalent to baseband conversion algebraically using conversion matrix symmetry. However, this fast conversion method assumes that the conversion matrix itself has symmetry and other method should be exploited when only local symmetry exists. The reference [6] also proposed resizing in the coded data domain, however an interlaced frame may become blurred since this method uses averaging both fields. In this regard, de-interlacing algorithm in the code domain is proposed in [7]. This method prepares the de-interlacing matrix converted by DCT, and applies it to DCT coefficients. However, intensive computational complexity is expected and therefore a fast algorithm is desired.

III. PROPOSED RESOLUTION CONVERSION ALGORITHM FOR DV TO MPEG-4 TRANSCODING

A. Integration of matrix operation

DV is always coded with an interlaced frame while MPEG-4 is often coded with a progressive frame. Therefore, when DV (ITU-R601) to MPEG-4 (e.g., SIF) transcoding is performed, picture size conversion as well as de-interlacing are required. Although there are various methods for resizing, care must be taken for interlace to progressive conversion. For example, the temporal luminance change between fields is often large

0-7803-8554-3/04/$20.00

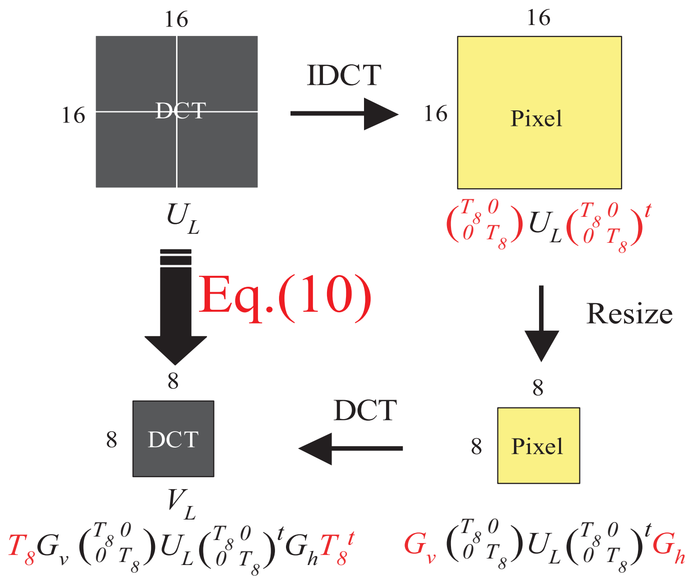


Fig. 1. 1/2 resolution conversion

in a panning scene. In such a case, picture resizing using average operation may become blurred severely. Therefore, field dropping is often used as a vertical resizing is often used for ITU-R601 to SIF conversion. In the following, we describe a fast compressed domain resizing method with de-interlacing.

A set of four 4*4 sub-matrices X_{00}, X_{01}, X_{10} and X_{11} for an 8*8 DCT luminance matrix X_L are first defined. Similarly, sub-matrices are also defined for neighboring three 8*8 DCT matrices Y_L, Z_L, and W_L. Then a combined matrix U_L for four 8*8 DCT blocks can be written as Eq. (1) using the above sub-matrices.

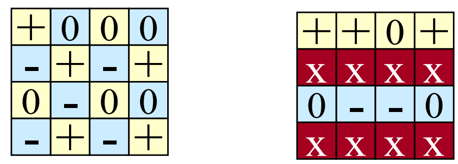
$$U_L = \begin{pmatrix} X_{00} & X_{01} & Y_{00} & Y_{01} \\ X_{10} & X_{11} & Y_{10} & Y_{11} \\ Z_{00} & Z_{01} & W_{00} & W_{01} \\ Z_{10} & Z_{11} & W_{10} & W_{11} \end{pmatrix} \tag{1}$$

Then the target resized 8*8 DCT matrix V_L can be derived by Eq. (2).

$$V_L = T_8 G_h \begin{pmatrix} T_8^t & 0 \\ 0 & T_8^t \end{pmatrix} U_L \begin{pmatrix} T_8 & 0 \\ 0 & T_8 \end{pmatrix} G_w T_8^t \tag{2}$$

Here, the matrix T_n is $n*n$ DCT matrix, and t denotes transposition. G_h and G_v are picture size conversion matrices horizontally and vertically, respectively. G_h and G_v can be arbitrary designed according to filtering algorithm. Therefore, in the case of 601 to SIF conversion, for example, G_h^{ave} and G_v^{sub} can be defined as the bilinear resizing matrix for horizontal direction and sub-sampling matrix for vertical direction respectively. Fig. 1 shows an example of both compressed and baseband domain resolution conversion to half size.

As all the matrices except for input matrix U_L can be calculated beforehand, the conversion operation complexity can be reduced further as described in the following. In the above resizing example (horizontally bilinear resizing and vertically subsampling), the right hand and left hand of U_L



(a) A & A' (b) K & K''

Fig. 2. Relation between sub-matrices

in Eq. (2) can be rewritten as

$$\begin{pmatrix} T_8 & 0 \\ 0 & T_8 \end{pmatrix} G_w^{ave} T_8^t = \begin{pmatrix} A & B & A' & B' \\ C & D & C' & D' \end{pmatrix}^t \tag{3}$$

$$T_8 G_h^{sub} \begin{pmatrix} T_8^t & 0 \\ 0 & T_8^t \end{pmatrix} = \begin{pmatrix} K & L & K'' & L'' \\ M & N & M'' & N'' \end{pmatrix}. \tag{4}$$

where A, B, C, D, K, L, M and N are 4*4 sub-matrices. The sub-matrices A and A' alternate positive and negative as shown in Fig.2(a) where "0", "+" and "-" mean zero, same, positive / negative reversed, respectively. For example, $a_{00} = a'_{00}$ and $a_{10} = -a'_{10}$ in the figure. By observing the relationship in Fig. 2(a), the operator " $'$ " can be expressed by Eq. (5)

$$A' = H_4 A H_4 \tag{5}$$

where H_4 is the 4*4 matrix which can be defined by Eq. (6).

$$\begin{aligned} h_{i,i} &= \begin{cases} 1 & (i = 0, 2) \\ -1 & (i = 1, 3) \end{cases} \\ h_{i,k} &= \quad 0 \qquad (i \neq k) \end{aligned} \tag{6}$$

As for matrices K and K'', coefficients in the 0th row, the 2nd row, and the odd rows (1st and 3rd) are the same, positive / negative reverse, and completely different, respectively. Fig. 2(b) shows the relation between the sub-matrices K and K'' where "x" means completely different. For example, $k_{00} = k''_{00}$, $k_{10} \neq k''_{10}$, and $k_{21} = -k''_{21}$ in the figure. These relations can be also described in Eq. (7).

$$(E_4 + H_4)K = J_4(E_4 + H_4)K'' \tag{7}$$

where J_4 is the 4*4 matrix which can be defined by Eq. (8).

$$\begin{aligned} j_{i,i} &= \begin{cases} 1 & (i = 0, 3) \\ -1 & (i = 1, 2) \end{cases} \\ j_{i,k} &= \quad 0 \qquad (i \neq k) \end{aligned} \tag{8}$$

B. Fast algorithm for conversion matrix using local symmetry

Since only lower frequency components are remained in resizing in the horizontal direction in interlaced frames, we apply a low pass filter for input luminance matrix U_L to simplify calculations. Matrix $\hat{U}_L$ in Eq. (9) shows the low-pass filtered matrix U_L defined in Eq. (1).

$$\hat{U}_L = \begin{pmatrix} X_{00} & 0 & Y_{00} & 0 \\ 0 & 0 & 0 & 0 \\ Z_{00} & 0 & W_{00} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{9}$$

It is noted that aliasing noise caused by simple down-sampling can be also avoided by this simple frequency domain filtering. Although no low pass filter is required for the vertical direction since field dropping is conducted, we performed filtering for both direction in order to show basic step of calculation reduction. If no filtering is applied in the vertical direction, Eq. (9) is modified as shown later.

Here, we also use $\hat{V}_L$ instead of V_L for the resized DCT matrix with low pass filtered input matrix U_L. Then $\hat{V}_L$ can be expressed by Eq. (10) when Eq. (3), (4) and (9) are substituted for Eq. (2).

$$\hat{V}_L = \begin{pmatrix} P_K A^t + Q_K A'^t & P_K C^t + Q_K C'^t \\ P_M A^t + Q_M A'^t & P_M C^t + Q_M C'^t \end{pmatrix} \tag{10}$$

$$\begin{aligned} P_K &= (KX_{00} + K''Z_{00}) \\ P_M &= (MX_{00} + M''Z_{00}) \\ Q_K &= (KY_{00} + K''W_{00}) \\ Q_M &= (MY_{00} + M''W_{00}) \end{aligned} \tag{11}$$

As can be seen in Eq. (10), there are many common terms such as P_K and Q_K. Therefore, the calculation load can be reduced by calculating Eq. (11) in advance.

The 4*4 sub-matrix $\hat{V}_{00}$ at the upper left of $\hat{V}_L$ in Eq.(10) can be rewritten as Eq. (12).

$$\hat{V}_{00} = (KX_{00} + K''Z_{00})A^t + (KY_{00} + K''W_{00})A'^t \tag{12}$$

Then this equation can be rewritten as the following equation by using Eq. (5) and $H_4^2 = E_4$ if the Eq. (10) are multiplied by the matrix $(E_4 \pm H_4)$.

$$\begin{aligned} \hat{V}_{00}(E_4 \pm H_4) =& \{K(X_{00} \pm Y_{00}H) \\ &+ K''(Z_{00} \pm W_{00}H)\}A^t(E_4 \pm H_4) \end{aligned} \tag{13}$$

Eq. (13) can be separately calculated for even / odd columns using the sparse matrix. Similarly, other 4*4 sub-matrices $\hat{V}_{01}$, $\hat{V}_{10}$ and $\hat{V}_{11}$ of Eq. (10) can be also obtained if matrix A is replaced with matrix C. Then Eq. (10) can be simplified by calculating $A^t(E_4 \pm H_4)$ and $C^t(E_4 \pm H_4)$. Here we define the parts of Eq. (13) as $\hat{V}_{00}^+$ and $\hat{V}_{00}^-$ in Eq. (14) and (15).

$$\hat{V}_{00}^+ = K(X_{00} + Y_{00}H_4) + K''(Z_{00} + W_{00}H_4) \tag{14}$$

$$\hat{V}_{00}^- = K(X_{00} - Y_{00}H_4) + K''(Z_{00} - W_{00}H_4) \tag{15}$$

Then Eq. (16) can be obtained by multiplying $(E_4 + J_4)(E_4 + H_4)$ to both terms of Eq. (14). Here, we use relationship of $J_4^2 = E_4$.

$$\begin{aligned} (E_4 + J_4)(E_4 + H_4)\hat{V}_{00}^+ =& (E_4 + J_4)(E_4 + H_4)K \\ &(X_{00} + Y_{00}H_4 + Z_{00} + W_{00}H_4) \end{aligned} \tag{16}$$

Since Eq. (16) is multiplied by $(E_4 + J_4)(E_4 + H_4)$, only the 0th row remains. On the other hand, Eq. (17) is obtained by multiplying $(E_4 - J_4)(E_4 + H_4)$ to both terms of Eq. (14). In this case, only the 2nd row remains by applying a similar process.

$$\begin{aligned} (E_4 - J_4)(E_4 + H_4)\hat{V}_{00}^+ =& (E_4 - J_4)(E_4 + H_4)K \\ &(X_{00} + Y_{00}H_4 - Z_{00} - W_{00}H_4) \end{aligned} \tag{17}$$

Since only even rows in matrices K and K'' have local symmetry, the odd rows are calculated as follows:

$$\begin{aligned} (E_4 - H_4)\hat{V}_{00}^+ =& (E_4 - H_4)K(X_{00} + Y_{00}H_4) \\ &+ (E_4 - H_4)K''(Z_{00} + W_{00}H_4) \end{aligned} \tag{18}$$

Although the conversion matrices $(E_4 - H_4)K$ and $(E_4 - H_4)K''$ are sparse, computational complexity can not be reduced in Eq. (18) when compared with odd rows of Eq. (14). Eq. (15) required for the matrix $\hat{V}_{00}(E_4 - H_4)$ can be obtained in similar fashion. The 0th row, the 2nd row, and the odd rows of the matrix $\hat{V}_{00}^-$ can be calculated from Eq. (19), (20), and (21), respectively.

$$\begin{aligned} &(E_4 + J_4)(E_4 + H_4)\hat{V}_{00}^- \\ &= 4K_{0th}(X_{00} - Y_{00}H_4 + Z_{00} - W_{00}H_4) \end{aligned} \tag{19}$$

$$\begin{aligned} &(E_4 - J_4)(E_4 + H_4)\hat{V}_{00}^- \\ &= 4K_{2nd}(X_{00} - Y_{00}H_4 - Z_{00} + W_{00}H_4) \end{aligned} \tag{20}$$

$$\begin{aligned} &(E_4 - H_4)\hat{V}_{00}^- \\ &= 2K_{odd}(X_{00} - Y_{00}H_4) + 2K''_{odd}(Z_{00} - W_{00}H_4) \end{aligned} \tag{21}$$

where K_{0th}, K_{2nd} and K_{odd} are described as Eq. (22).

$$\begin{aligned} K_{0th} &= (E_4 + J_4)(E_4 + H_4)K/4 \\ K_{2nd} &= (E_4 - J_4)(E_4 + H_4)K/4 \\ K_{odd} &= (E_4 - H_4)K/2 \\ K''_{odd} &= (E_4 - H_4)K''/2 \end{aligned} \tag{22}$$

Similar approach can be applied for other matrices, $\hat{V}_{01}$, $\hat{V}_{10}$ and $\hat{V}_{11}$ by replacing the matrix K with the matrix M. Since there are many terms which are common in calculating the sub-matrix, the amount of operations is reduced by sharing calculation results.

When no low pass filtering is applied in the vertical direction, Eq. (9) can be replaced with Eq. (23) and similar procedure to the above can be applied for the calculation reduction.

$$\bar{U}_L = \begin{pmatrix} X_{00} & 0 & Y_{00} & 0 \\ X_{10} & 0 & Y_{10} & 0 \\ Z_{00} & 0 & W_{00} & 0 \\ Z_{10} & 0 & W_{10} & 0 \end{pmatrix} \tag{23}$$

In order to distinguish these methods, we define the former method (LPF for both directions) as the proposed (LPF_HV) method and the latter (LPF for horizontal only) as the proposed (LPF_H).

IV. Results

A. Comparison of computational complexity

The computational complexity comparison for resolution conversion with de-interlacing is shown in Tab. I. Here, we compare the proposed methods (LPF_H and LPF_HV) with a conventional method and baseband domain conversion.

The conventional method performs resolution conversion by applying fast 4*4 IDCT (28 multiplications and 66 additions [8]) to the lower frequency 4*4 of 8*8 DCT coefficients.

TABLE I

OPERAND OF RESOLUTION CONVERSION

	Multiplications	Additions
Proposed (LPF_H) conversion	840 (95.5%)	888 (37.2%)
Proposed (LPF_HV) conversion	496 (56.4%)	488 (20.5%)
Conventional conversion	576 (65.5%)	512 (21.5%)
Baseband conversion	880 (100.0%)	2384 (100.0%)

(times / block)

Note that this method extracts low frequency component of interlaced block DCT, de-interlacing is not achieved.

The baseband conversion uses four fast 8*8 IDCT operations (128 multiplications and 430 additions [8] for each DCT) with a bilinear resizing in the horizontal direction (1 addition per 4 pixels) and field dropping, then it is followed by one 8*8 DCT operation.

As can be seen from the table, both the proposed methods can achieve fast de-interlacing and resizing in both multiplication and addition operations when compared with the baseband domain conversion. Since recent processor can achieve almost the same latency for both multiplication and addition, total number of multiplication and addition operations becomes more important than the mere number of multiplication operation. As for the conventional method, a little bit faster operation can be achieved in the proposed (LP_HV), although de-interlacing is not achieved as stated earlier.

The other method [5] uses a decomposition method so that the conversion matrices become sparser to reduce the computation. However, it is restricted only when the reduction matrix used for resolution conversion satisfies Eq. (6). Therefore, when a resizing matrix G is asymmetrical, the conventional method cannot be applied since the conversion matrix is inseparable. On the other hand, the proposed method can calculate even if there is only local symmetry as shown in Eq. (22). In this case, the proposed method can simplify calculation partially using matrices H and J.

B. Comparison of DV to MPEG-4 transcoding quality

We also evaluated DV (720*480, 30fps, 150frames) to MPEG-4 (360*240, 30fps, M=1, N=30) transcoding performance. As is discussed earlier, for interlaced 601 to progressive SIF conversion, field dropping usually provides better quality than that of averaging. Therefore, we decoded DV and then dropped one field of the decoded frame as a vertical resizing method with bilinear resizing for horizontal direction and used it as an original for PSNR calculation. As for transcoding scheme, we use the proposed method in [2] and apply different picture size conversion methods discussed in the above. The PSNR results are shown in Fig. 3. In the figure, large degradation is observed in the conventional method since only low frequency components of interlaced DCT is used and therefore no de-interlacing is performed. As for the transcoding using the proposed (LPF_HV) method, although much better PSNR characteristics than the conventional method can be obtained,

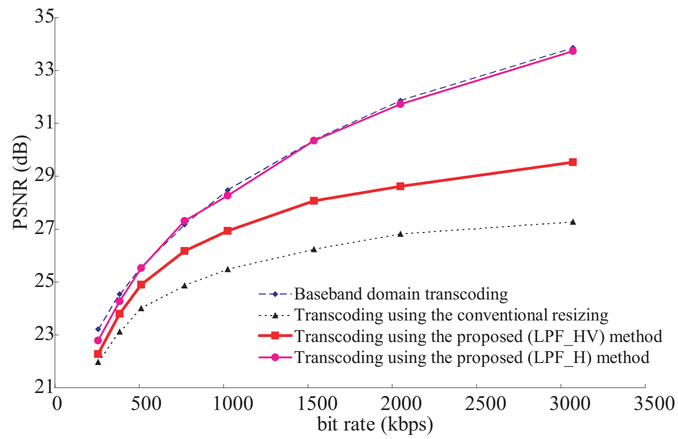


Fig. 3. PSNR of DV to MPEG-4 transcoding using baseband domain resizing

up to a few dB degradation is still observed when compared with the baseband domain conversion and transcoding due to LPF in the vertical direction which results in cutting high frequency components which should be preserved in interlaced block. In the transcoding using the proposed (LPF_H) method, it can be seen that almost the same PSNR characteristics as baseband domain conversion is achieved.

V. CONCLUSIONS

We proposed a fast resolution conversion with de-interlacing in the compressed domain for DV to MPEG-4 transcoding. By integrating DCT matrix operation and exploiting local symmetry, up to about 40% multiplication and 80% addition operations can be reduced when compared with baseband domain operation. The experimental results also confirm very high transcoding capabilities with the proposed method.

ACKNOWLEDGMENT

The authors thank Mr. Nagata for his constructive and helpful discussion.

REFERENCES

[1] A. Vetro, C. Christopoulos, and H. Sun, “Video transcoding architectures and techniques: An overview,” *IEEE Signal Processing Magazine*, vol. 20, pp. 18–29, Mar 2003.

[2] H. Kato, T. Sano, and Y. Nakajima, “A fast DV to MPEG-2 IBP-picture conversion algorithm,” *IEEE Trans. on Consumer Electronics*, vol. 49, pp. 433–440, May 2003.

[3] Y. R. Lee, C. W. Lin, and C. C. Kao, “A DCT-domain video transcoder for spatial resolution downconversion,” *Lecture Notes in Computer Science*, vol. 2314, pp. 207–218, 2002.

[4] R. Dugad and N. Ahuja, “A fast scheme for downsampling and upsampling in the DCT domain,” in *Proc. Int’l Conf. Image Processing*, pp. 909–913, 1999.

[5] R. Dugad and N. Ahuja, “A fast scheme for image size change in the compressed domain,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 461–474, 2001.

[6] P. Yin, M. Wu, and B. Lui, “Video transcoding by reducing spatial resolution,” in *Proc. IEEE Int’l Conf. Image Processing*, pp. 972–975, 2000.

[7] J. Song and B. Yeo, “Fast extraction of spatially reduced image sequences from MPEG-2 compressed video,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1100–1114, 1999.

[8] F. Kamangar and K. R. Rao, “Fast algorithms for the 2-D discrete cosine transform,” *IEEE Trans. Computers*, vol. C-31, no. 9, pp. 899–906, 1982.