Efficient down-up sampling using DCT kernel for MPEG-21 SVC

Il-hong Shin and HyunWook Park
Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea
hwpark@athena.kaist.ac.kr

ABSTRACT

Down-up sampling of images is an essential process for spatial scalability of the video coding standard. We propose an efficient down-up sampling method in spatial domain using DCT kernel. The computational complexity of the proposed method is reduced by taking advantages of various symmetries of the combined DCT kernel. The proposed down-up sampling method is compared to that of MPEG-21 SVM 3.0 [1]. The proposed method outperforms the performance of MPEG-21 SVM 3.0 with slightly increased addition operation.

1. INTRODUCTION

Recently, MPEG-21 SVM 3.0 was proposed to provide scalable video model for standardization. It inherited most building blocks of H.264 with some improved features for scalability such as MCTF (motion compensated temporal filtering) [1]. For spatial scalability, 11 taps and 6 taps filters were exploited in SVM 3.0 for down and up sampling, respectively. Although their computational complexity and down-up sampling performance are proper to be exploited in scalable video coding, more improvement must be made for rate-distortion performance of scalable video coding, *i.e.,* the performance of spatial scalability depends on the adopted down-up sampling method.

Many researches were performed for image resizing by using DCT kernel employing characteristics of DCT [2-6]. The down-up sampling using DCT kernel provides more improved visual quality and PSNR than those of simple bilinear interpolation method. The simple down-up sampling method using DCT kernel is the sequential operation of IDCT and DCT with bilinear interpolation in spatial domain. Although fast IDCT and DCT method is used for down-up sampling with consideration of zero coefficients in high frequency band, their computational complexity is higher than those of spatial domain filtering in SVM 3.0.

In this paper, we introduce an efficient down-up sampling method using DCT and IDCT kernels, which are combined for efficient calculation.

The proposed down-up sampling method exploits various symmetries inherited from DCT kernel, which is described in section 2. Experimental results and analysis of the proposed method are given in section 3, and section 4 concludes the paper.

2. Down-up sampling in spatial domain using DCT kernel

Down sampling in spatial domain using DCT kernel is expressed by combination of DCT and IDCT kernels as follows:

$$D_{\frac{N}{2}\times\frac{N}{2}} = T^{t}_{\frac{N}{2}\times\frac{N}{2}} \times \left(\begin{bmatrix} T^{u}_{\frac{N}{2}\times N} \\ O_{\frac{N}{2}\times N} \end{bmatrix} \times B_{N\times N} \times \begin{bmatrix} T^{u,t}_{\frac{N}{2}\times N} & O_{N\times\frac{N}{2}} \end{bmatrix} \right) \times T_{\frac{N}{2}\times\frac{N}{2}} \quad (1)$$
$$= V^{D}_{\frac{N}{2}\times N} \times B_{N\times N} \times H^{D}_{N\times\frac{N}{2}}$$

where T denotes 1D DCT kernel, and $B_{N\times N}$ and $D_{\frac{N}{2}\times\frac{N}{2}}$ are the original image of $N\times N$ and down-sampled image of $\frac{N}{2}\times\frac{N}{2}$, respectively. T^{u} represents upper kernels of DCT from row 1 to $\frac{N}{2}$, and the superscript, t, means transpose of the matrix. In eq. (1), $O_{\frac{N}{2}\times N}$ is the zero matrix of $\frac{N}{2}\times N$. $V^{D}_{\frac{N}{2}\times N}$ and $H^{D}_{\frac{N}{2}\times N}$ denote the vertical and horizontal down-sampling kernels, respectively.

In the similar way, the upsampled image $U_{2N\times 2N}$ is obtained as follows:

$$U_{2N\times 2N} = T^{t}_{2N\times 2N} \times \left(\begin{bmatrix} T_{N\times N} & O_{N\times N} \\ O_{N\times N} & O_{N\times N} \end{bmatrix} \times B_{N\times N} \times \begin{bmatrix} T^{t}_{N\times N} & O_{N\times N} \\ O_{N\times N} & O_{N\times N} \end{bmatrix} \right) \times T_{2N\times 2N} \quad (2)$$
$$= V^{U}_{2N\times N} \times B_{N\times N} \times H^{U}_{N\times 2N}$$

0-7803-9134-9/05/$20.00 ©2005 IEEE

It shows simple zero padding in the high frequency band. The element of the vertical up-sampling kernel, $V^U_{2N\times N}$ can be expressed as follows:

$$
\begin{aligned}
V^U_{2N\times N}(n_1,n_2) &= \sum_{k=0}^{N-1} p(k)\cdot\cos\Big(\frac{\pi k\cdot(2n_1+1)}{4N}\Big)\cdot\cos\Big(\frac{\pi k\cdot(2n_2+1)}{2N}\Big)\\
&= \sum_{k=0}^{N-1}\frac{p(k)}{2}\cdot\left(\cos\Big(\frac{\pi k\cdot(4n_2-2n_1+1)}{4N}\Big)+\cos\Big(\frac{\pi k\cdot(4n_2+2n_1+3)}{4N}\Big)\right) \qquad (3)\\
&= \sum_{k=0}^{N-1} p(k)\bullet cc(n_1,n_2,k)\\
&\textit{where } p(0)=\frac{1}{N},\ p(k)=\frac{2}{N}\ \textit{ for } 1\le k\le N-1
\end{aligned}
$$

where n_1 and n_2 are the row and column indices of the vertical up-sampling kernel, respectively. From eq. (3), it is shown that the up-sampling kernel is symmetric, *i.e.,* $V^U_{2N\times N}(n_1,n_2)=V^U_{2N\times N}(2N-n_1-1,N-n_2-1)$. This symmetry can be exploited for reducing multiplications of up-sampling kernel as follows:

$$
\begin{bmatrix} a & d\\ b & c\\ c & b\\ d & a\end{bmatrix}\times\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}\dfrac{\begin{bmatrix}a-d\\ b-c\end{bmatrix}}{2}\times(x_1-x_2)+\dfrac{\begin{bmatrix}a+d\\ b+c\end{bmatrix}}{2}\times(x_1+x_2)\\ \dfrac{\begin{bmatrix}b+c\\ a+d\end{bmatrix}}{2}\times(x_1+x_2)-\dfrac{\begin{bmatrix}b-c\\ a-d\end{bmatrix}}{2}\times(x_1-x_2)\end{bmatrix} \qquad (4)
$$

Eq. (4) shows basic method for reducing multiplications, where left matrix shows similar symmetry of up-sampling kernel in eq. (3). As shown in eq. (4), the required number of multiplications were reduced to half of that of the original one. Using eq. (4), we can decompose up-sampling matrix as follows:

$$
F^1_{N\times\frac{N}{2}}=\begin{bmatrix} V(0,0)-V(0,N-1) & \cdots & V(0,\frac{N}{2}-1)-V(0,\frac{N}{2})\\ \vdots & \ddots & \vdots\\ V(N-1,0)-V(N-1,N-1) & \cdots & V(N-1,\frac{N}{2}-1)-V(N-1,\frac{N}{2})\end{bmatrix}/2 \qquad (5)
$$

$$
F^2_{N\times\frac{N}{2}}=\begin{bmatrix} V(0,0)+V(0,N-1) & \cdots & V(0,\frac{N}{2}-1)+V(0,\frac{N}{2})\\ \vdots & \ddots & \vdots\\ V(N-1,0)+V(N-1,N-1) & \cdots & V(N-1,\frac{N}{2}-1)+V(N-1,\frac{N}{2})\end{bmatrix}/2 \qquad (6)
$$

where superscript and subscript of $V^U_{2N\times N}$ are omitted for simplicity in eqs. (5) and (6). Also, the input image $B_{N\times N}$ is decomposed to be computed by eq. (4) as follows:

$$
B^1_{\frac{N}{2}\times N}=\begin{bmatrix} B(0,0)-B(N-1,0) & \cdots & B(0,N-1)-B(N-1,N-1)\\ \vdots & \ddots & \vdots\\ B(\frac{N}{2}-1,0)-B(\frac{N}{2},0) & \cdots & B(\frac{N}{2}-1,N-1)-B(\frac{N}{2},N-1)\end{bmatrix} \qquad (7)
$$

$$
B^2_{\frac{N}{2}\times N}=\begin{bmatrix} B(0,0)+B(N-1,0) & \cdots & B(0,N-1)+B(N-1,N-1)\\ \vdots & \ddots & \vdots\\ B(\frac{N}{2}-1,0)+B(\frac{N}{2},0) & \cdots & B(\frac{N}{2}-1,N-1)+B(\frac{N}{2},N-1)\end{bmatrix} \qquad (8)
$$

Therefore, vertical filtering using eqs. (4)-(8) can be re-written as follows:

$$
V^U_{2N\times N}\times B_{N\times N} = \begin{bmatrix} F^1_{N\times\frac{N}{2}}\times B^1_{\frac{N}{2}\times N}+F^2_{N\times\frac{N}{2}}\times B^2_{\frac{N}{2}\times N}\\ R_{row}(F^2_{N\times\frac{N}{2}}\times B^2_{\frac{N}{2}\times N}-F^1_{N\times\frac{N}{2}}\times B^1_{\frac{N}{2}\times N})\end{bmatrix} \qquad (9)
$$

where R_{row} denotes the row inversion of matrix. The horizontal up-sampling procedure can be performed in the similar way using eqs. (4)-(9). In addition to this reduction of multiplications, more reduction is still possible if characteristics of F^1 and F^2 are considered as follows:

$$
\begin{aligned}
F^1(n_1,n_2) &= V^U_{2N\times N}(n_1,n_2)-V^U_{2N\times N}(n_1,N-n_2-1)\\
&= \sum_{k=0}^{N-1} p(k)\cdot(1-(-1)^k)\cdot cc(n_1,n_2,k)\\
F^2(n_1,n_2) &= V^U_{2N\times N}(n_1,n_2)+V^U_{2N\times N}(n_1,N-n_2-1) \qquad (10)\\
&= \sum_{k=0}^{N-1} p(k)\cdot(1+(-1)^k)\cdot cc(n_1,n_2,k)\\
&\textit{for } 0\le n_1<N,\ 0\le n_2<\frac{N}{2}
\end{aligned}
$$

where $cc(n_1,n_2,k)$ is defined in eq. (3). In eq. (10), F^2 filter is symmetric, *i.e.,* $F^2(N-n_1-1,\frac{N}{2}-n_2-1)=F^2(n_1,n_2)$. By using this symmetry, more decomposition of F^2 filter is possible, where F^2 filter is decomposed into $F^{2,1}$ and $F^{2,2}$ filters by using eq. (5)-(6). The decomposition of F^2

filter can be possible until F^2 filter has 2 rows and 1 column.

For computational efficiency of F^1 filter, we can describe the F^1 filter as follows:

$$F^1_{N\times\frac{N}{2}} = \begin{bmatrix} A_{\frac{N}{2}\times\frac{N}{4}} & B_{\frac{N}{2}\times\frac{N}{4}} \\ C_{\frac{N}{2}\times\frac{N}{4}} & D_{\frac{N}{2}\times\frac{N}{4}} \end{bmatrix} \quad (11)$$

where $A_{\frac{N}{2}\times\frac{N}{4}}$, $B_{\frac{N}{2}\times\frac{N}{4}}$, $C_{\frac{N}{2}\times\frac{N}{4}}$, and $D_{\frac{N}{2}\times\frac{N}{4}}$ are the sub-matrices of F^1. The $A_{\frac{N}{2}\times\frac{N}{4}}$ and $D_{\frac{N}{2}\times\frac{N}{4}}$ have relationship as follows:

$$\begin{aligned} &A(2n_1, n_2) - D(N-2-n_1-\frac{N}{2}\cdot n_2, \frac{N}{2}-1-n_1) = \\ &A(2n_1+1, n_2) - D(N-2-n_1-\frac{N}{2}\cdot n_2+1, \frac{N}{2}-1-n_1) \end{aligned} \quad (12)$$

Also, The $B_{\frac{N}{2}\times\frac{N}{4}}$ and $C_{\frac{N}{2}\times\frac{N}{4}}$ have relationship as follows:

$$\begin{aligned} &B(2n_1, n_2) - C(2n_2, n_1) = \\ &C(2n_2+1, n_1) - B(2n_1+1, n_2) \end{aligned} \quad (13)$$

Thanks to these relations, F^1 filter is decomposed further. For example, when N is 4, the F^1 filter can be decomposed as follows:

$$\begin{aligned} F^1_{4\times2} &= \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \\ f_{2,0} & f_{2,1} \\ f_{3,0} & f_{3,1} \end{bmatrix} = \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \\ f_{0,1} & f_{0,0} \\ f_{1,1} & f_{1,0} \end{bmatrix} - \begin{bmatrix} \multicolumn{2}{c}{O_{2\times2}} \\ f_{0,1}-f_{2,0} & f_{0,0}-f_{2,1} \\ f_{1,1}-f_{3,0} & f_{1,0}-f_{3,1} \end{bmatrix} \quad (14) \\ &= F^{1,s}_{4\times2} - F^{1,c}_{4\times2} \end{aligned}$$

where $f_{i,j}$ is the (*i-j*)-th element of $F^1_{4\times2}$. In eq. (14), $f_{0,0}-f_{2,1} = f_{1,0}-f_{3,1}$ and $f_{0,1}-f_{2,0} = -(f_{1,1}-f_{3,0})$ are obvious from eqs. (12) and (13). Therefore, F^1 is decomposed to $F^{1,s}_{4\times2}$ and $F^{1,c}_{4\times2}$. Also decomposition of $F^{1,s}_{4\times2}$ is possible by using eqs. (5) and (6), where it can be represented as $F^{1,s,1}_{4\times2}$ and $F^{1,s,2}_{4\times2}$. Eq. (14) shows reduction of multiplications by using the decomposed matrices $F^{1,s}_{4\times2}$ and $F^{1,c}_{4\times2}$. When N is 4, the vertical up-sampling in eq. (9) can be simplified as follows:

$$V^U_{8\times4} \times B_{4\times4} = \begin{bmatrix} V^1_B + V^2_B \\ R_{row}(V^2_B - V^1_B) \end{bmatrix} \quad (15)$$

where the matrices V^1_B and V^2_B are defined as follows:

$$\begin{aligned} V^1_B &= \begin{bmatrix} F^{1,s,1}_{2\times1} \times B^{1,1}_{1\times4} + F^{1,s,2}_{2\times1} \times B^{1,2}_{1\times4} \\ F^{1,s,2}_{2\times1} \times B^{1,2}_{1\times4} - F^{1,s,1}_{2\times1} \times B^{1,1}_{1\times4} \end{bmatrix} - F^{1,c}_{4\times2} \times B^1_{2\times4} \\ V^2_B &= \begin{bmatrix} F^{2,1}_{2\times1} \times B^{2,1}_{1\times4} + F^{2,2}_{2\times1} \times B^{2,2}_{1\times4} \\ R_{row}(F^{2,2}_{2\times1} \times B^{2,2}_{1\times4} - F^{2,1}_{2\times1} \times B^{2,1}_{1\times4}) \end{bmatrix} \end{aligned} \quad (16)$$

The horizontal up-sampling filter can be made in the similar way to the vertical up-sampling filter. Although the matrix size of down-sampling is different to that of up-sampling, the calculation procedure of down-sampling is identical to that of up-sampling. Therefore, the computational complexities of down and up-sampling are the same.

Table 1 shows comparison of the computational complexity of SVM 3.0, fast DCT&IDCT, and the proposed up-sampling methods.

Per pixel	Multiplication	Addition
SVM 3.0	1.5	3.75
Fast DCT&IDCT	2.1875	4.4375
Original matrix	6	4.5
Proposed	1.5	5.25

Table 1. The computational complexity of up-sampling method (N=4 was applied)

We use the fastest DCT&IDCT [4] considering the zero coefficients. The proposed method outperforms the fast DCT&IDCT method with slightly increased number of additions, since the combined kernels provide more efficient calculation. The computational complexity of the proposed method has slightly increased addition operation to that of SVM 3.0.

3. Experimental results

We implemented the proposed down-up sampling method in SVM 3.0 video codec [1]. For an impartial comparison, the proposed down-up sampling kernels were converted to the fixed-point 16 bit integer type. Although the PSNR loss is expected in the fixed-point conversion, the loss is negligible. The value of 32 is multiplied to the kernel and then rounded to the nearest integer for converting the proposed kernels. We can further reduce the number of multiplication by using shift operation in the integer implementation.

We used two video sequences of Football and Foreman which have a spatial resolution of 352×288 (CIF) and 15 Hz frame rate. The base layer were encoded with 176×144 (QCIF) after down-sampling using SVM 3.0 and the proposed method. The enhancement layer uses the CIF resolution with up-sampling of the proposed method and SVM 3.0 for inter-layer prediction [1]. Although rate control does not exist in SVM 3.0, we set nearly the same bitrate in base layer for an impartial comparison. The GOP size and search range were 16 and 96 in integer pel, respectively.

Fig. 1 shows the rate-PSNR curves of the proposed and SVM 3.0 methods with 15 Hz. The temporal scalability will be given inherently by using MCTF [1], where GOP size as 16 provides four levels of temporal scalability. The high, middle, and low represent the bitrates of base layer. Football and Foreman have high (564, 228), middle (254, 88), and low (82, 32) kbps. If the base layer has the high bitrate, prediction in the enhancement layer will be efficient. Thus, the high bitrates shows extreme condition of experimental setting to show maximum performance of the proposed method. As shown in Fig. 1, the proposed method outperforms SVM 3.0 in high and middle bitrates. The maximum improvements of PSNR are about 1.2 dB and 0.5 dB in high and middle bitrates of base layer, respectively. Of course, when base layer was encoded with low bitrate, the rate-PSNR curves showed nearly the same performance.

The visual quality of reconstructed frames seems good in comparison with that of SVM 3.0.

4. Conclusions

We proposed very efficient down-up sampling method for spatial scalability in SVM 3.0. The symmetric property in the combined DCT kernel provides efficient filtering method, which enables to operate properly in SVC decoder. The experimental result shows much PSNR improvements, when the base layer is encoded with high bitrates. The proposed method could be applied to MPEG-21 SVC standardization.

7. REFERENCES

[1] ISO/IEC/JTC1, "Scalable video model 3.0," ISO/IEC JTC1/WG11 Doc. N6716, Nov. 2004.

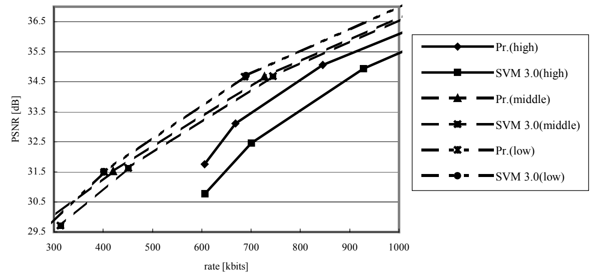
[2] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 11, pp. 461-474, Apr. 2001.

[3] H. W. Park, Y. S. Park, and S. K. Oh, "L/M fold image resizing in block-DCT domain using symmetric convolution," *IEEE Trans. Image Processing,* vol. 12, pp. 1016-1034, Sept. 2003.

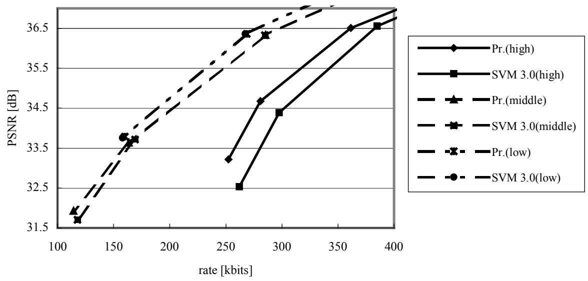
[4] C. Loeffer, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11-multiplications," *in IEEE Int. Conf. Accoustics, Speech and Signal Processing,* vol. 2, pp. 988-991, May. 1989.

[5] J. Mukherjee and S. K. Mitra, "Image resizing in the compressed domain using subband DCT," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 12, pp. 620-627, July. 2002.

[6] H. Shu and L. P. Chau, "An efficient arbitrary downsizing algorithm for video transcoding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 14, pp. 887-891, June. 2004.



(a) Football, 15 Hz



(b) Foreman, 15 Hz

Fig 1. Rate-PSNR curves of the SVM 3.0 and proposed method for (a) Football and (b) Foreman.