Direct 3-D DCT-to-DCT Resizing Algorithm for Video Coding

[a]Jin Li, [a]Moncef Gabbouj, [a]Jarmo Takala and [b]Hexin Chen
[a]*Faculty of Computing and Electrical Engineering, Tampere University of Technology, Finland*
[b]*School of Communication Engineering, Jilin University, Changchun, China*
[a]*FirstName.LastName@tut.fi*[1] [b]*chx@jlu.edu.cn*

Abstract

This paper proposes a direct DCT-to-DCT resizing algorithm for 3-D DCT based video coding. An $8 \times 8 \times 8$ *cube is resized to three modes along temporal dimension: a single* 8×8 *block, a downsized* $8 \times 8 \times 4$ *cube and two* $8 \times 8 \times 4$ *cubes. The mode selection is based on the local motion activity and determined after 2-D DCT on each block. In addition, the proposed resizing scheme directly resizes the DCT blocks in the transform domain using the DCT-to-DCT algorithm. Compared to traditional resizing approaches the proposed algorithm does not require the inverse transform and the computations in the spatial domain, thus is superior to other methods in terms of complexity. The proposed model is evaluated with the baseline video codec and the reference codec in the literature. Experimental results show a promising performance in terms of both coding efficiency and computational complexity. Potential applications could be for portable digital devices with low-power processors and other areas with real-time requirement.*

1. Introduction

Most today's video coding standards such as MPEG-4 and H.264 rely on the motion estimation/ compensation to exploit the temporal correlations among inter frames. However, this requires a large number of computations and significant computational power for hardware. Therefore, it is not appropriate for most portable devices such as mobile phones and digital cameras. Since for these types of applications, low complexity implementation and low power consumption are still the most critical issues.

An alternative approach is to use the three-dimensional discrete cosine transform (3-D DCT). A 3-D DCT video codec extends the 2-D DCT to the temporal dimension and can remove the temporal redundancy instead of utilizing motion estimation/compensation. Therefore, it is able to reach adequate compression efficiency with much less complexity.

One of the major problems in a 3-D DCT video codec is that if only utilizing fixed-length transform regardless the level of motion activity, the compression efficiency is usually inferior to the standardized video codec such as MPEG-4 and H.264. To solve this problem, several 3-D DCT algorithms [1]-[5] have been developed. These techniques utilize variable cube size of 3-D DCT based on local motion activity. All the methods deal with sequences containing static background well, but the coding efficiency maintains the same for those sequences with drastic motion. In addition, since the motion analysis is always performed in the spatial domain, the complexity is highly increased. In 2007, we proposed a resizing 3-D DCT scheme [6] to improve the compression efficiency and even simplify the complexity for sequences with low motion activity.

In what follows we describe a direct DCT-to-DCT resizing algorithm in the 3-D DCT domain. The $8 \times 8 \times 8$ cubes are resized into three modes based on the level of local motion activity. Compared to our previous work, this technique presents a higher coding efficiency and a lower complexity. Therefore, it is more suitable for low-power processors in portable digital devices and for the real-time applications.

The rest of this paper is organized as follows. The direct DCT-to-DCT resizing algorithm is presented in Section 2. In Section 3 the proposed method is implemented based on the 3-D DCT video coding. The experimental results are presented in section 4. Finally, we conclude the paper in section 5.

2. Proposed DCT-to-DCT resizing method

2.1 1-D DCT-to-DCT algorithm

For an input vector $\boldsymbol{X} = \{x_{0,} x_{1,} \dots , x_{N-1,}\}$, the DCT output vector $\boldsymbol{Y} = \{y_{0,} y_{1,} \dots , y_{N-1,}\}$ is given by the relation

$$\boldsymbol{Y} = \boldsymbol{T_N}\boldsymbol{X} \tag{1}$$

where the $N \times N$ transform matrix for N-point DCT is written as

$$T_N(i,j) = \sqrt{\frac{2}{N}} \begin{cases} 1/\sqrt{2}, & i = 0 \\ cos\dfrac{(2j+1)i\pi}{2N}, & otherwise \end{cases} \tag{2}$$

Let $\boldsymbol{B}_{N/2}$ denotes the type-IV DCT matrix, which is defined as

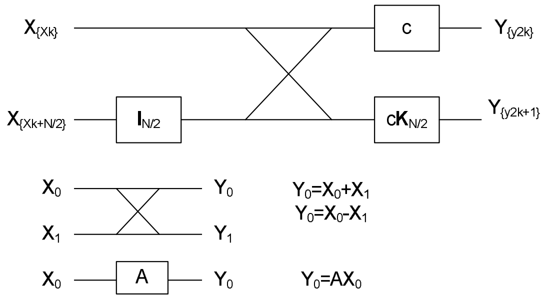


Fig.1. The signal flow graph of the 1-D DCT-to-DCT algorithm.

$$B_{N/2}(i,j) = \sqrt{\frac{2}{N/2}}cos\frac{(2i+1)(2j+1)\pi}{2N} \qquad (3)$$

Then, one of the two transform matrix [7] for the 1-D DC- to-DCT computation, $\boldsymbol{K}_{N/2}$, is

$$\boldsymbol{K}_{N/2} = \boldsymbol{B}_{N/2}\boldsymbol{T}_{N/2}^{-1} \qquad (4)$$

where $\boldsymbol{T}_{N/2}^{-1}$ is the inverse matrix of $\boldsymbol{T}_{N/2}$.

The other transform matrix, I_N, is defined as

$$I_N = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 \end{pmatrix}_{N\times N}$$

Let vectors $\boldsymbol{F}_{\{x_k\}}$ and $\boldsymbol{F}_{\{x_{k+N/2}\}}$, $k = 0,1,\dots,\frac{N}{2}-1$, denotes the two subsequences of $N/2$-point DCT coefficients, respectively, and $\boldsymbol{Y}_{\{y_{2k}\}}$ and $\boldsymbol{Y}_{\{y_{2k+1}\}}$, $k = 0,1,\dots,\frac{N}{2}-1$, denote the even-indexed subsequence and the odd-indexed subsequence of N-point DCT coefficients, respectively. We can calculate the 1-D DCT-to-DCT as

$$\boldsymbol{Y}_{\{y_{2k}\}} = c\left(\boldsymbol{F}_{\{x_k\}} + \boldsymbol{I}_{N/2}\boldsymbol{F}_{\{x_{k+N/2}\}}\right) \qquad (5)$$

$$\boldsymbol{Y}_{\{y_{2k+1}\}} = c\boldsymbol{K}_{N/2}\left(\boldsymbol{F}_{\{x_k\}} + \boldsymbol{I}_{N/2}\boldsymbol{F}_{\{x_{k+N/2}\}}\right) \qquad (6)$$

where $c = 1/\sqrt{2}$.

The signal flow graph for the 1-D DCT-to-DCT is shown in Fig.1.

2.2. The proposed resizing algorithm

The outline of resizing scheme [8] for 1-D signals is shown in Fig.2. Let $\mathbf{B}_1$ and $\mathbf{B}_2$ denote the $N/2$-point DCT of two consecutive $N/2$--point blocks $\mathbf{b}_1$ and $\mathbf{b}_2$, and let $\boldsymbol{B'}$ be the resulting resized $N/2$-point block. The traditional resizing scheme can be described as follows

1) Take the $N/2$-point DCT for block $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$, respectively;

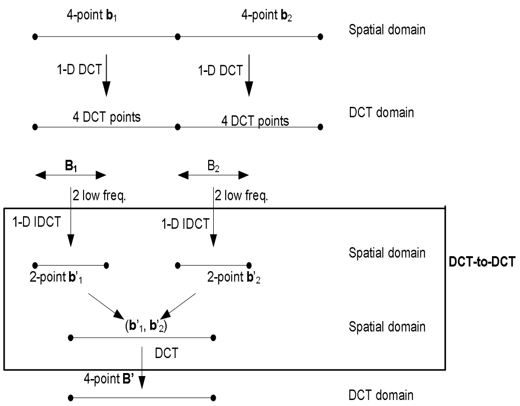


Fig.2. The traditional resizing algorithm at $N = 8$ and the proposed method replacing the computations inside the rectangle for the DCT-to-DCT algorithm.

2) Take the $N/4$-point inverse DCT for the $N/4$-point low frequency coefficients in $\boldsymbol{B_1}$, and the same as $\boldsymbol{B_2}$;
3) Combine the two $N/4$-point blocks $\boldsymbol{b_1'}$ and $\boldsymbol{b_2'}$ into one block $(\boldsymbol{b_1'},\boldsymbol{b_2'})$, then take its $N/2$-point DCT. Finally, the two $N/2$-point blocks $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are resized into a $N/2$-point DCT block $\boldsymbol{B}'$.

In the proposed resizing scheme, the DCT block $\boldsymbol{B}'$ is directly calculated from the two blocks $\boldsymbol{b_1'}$ and $\boldsymbol{b_2'}$ using the DCT-to-DCT algorithm in (6) and (7). Thus, the inverse transform and the computations in the spatial domain in step 2) and 3) are saved. The signal flow graph of traditional resizing method and the proposed resizing algorithm is shown in Fig.2.

Take $N = 8$ for instance, the DCT-to-DCT computation for the two blocks $\boldsymbol{b_1'}$ and $\boldsymbol{b_2'}$ only requires 6 multiplications (MUL) and 9 additions (ADD) [7]. While the inverse transform and the computations in the spatial domain in the traditional resizing methods takes 8 MUL and 14 ADD. Therefore, the proposed method is superior to other approaches in terms of complexity.

The resizing method relies on two principles. One, if the consecutive blocks are similar enough, the resizing operation does not produce significant high-frequency coefficients into the resulting block. Two, if the quantization is going to remove the high frequency of the blocks, the truncation of high frequency information with downsizing operation is justified.

3. Implementation of proposed method

In a 3-D DCT based codec, a video sequence is divided into a number of M×N×L cubes, where M×N is an image block of pixels, and L is the number of successive frames. The forward 3-D DCT is then defined as

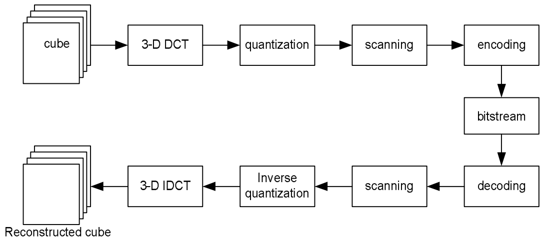


Fig.3. Block diagram of typical 3-D DCT based video codec

$$F(u,v,w) = C(u,L)C(v,N)C(w,M)\sum_{x=0}^{L-1}\sum_{y=0}^{N-1}\sum_{z=0}^{M-1} f(x,y,z)\times \quad (7)$$
$$\frac{\cos(2x+1)u\pi}{2L}\times\frac{\cos(2y+1)v\pi}{2N}\times\frac{\cos(2z+1)w\pi}{2M}$$

where

$$C(k,P) = \begin{cases} \sqrt{1/P} & k=0 \\ \sqrt{2/P} & otherwise \end{cases}$$

The 3-D DCT can be computed by taking one-dimensional transform separately in each of the three dimensions. Although a number of transforms are required, the computational complexity – even without taking the motion estimation into account – is superior to 2-D DCT video codec. A typical 3-D DCT based codec is described in Fig.3.

In the proposed method, three modes with variable size of cubes are used to perform the transformation. The mode decision and the resizing algorithm are determined in the process of temporal transform, thus the 2-D DCT can be first performed for each block as

$$F(u,v,t) = C(u,8)C(v,8)\sum_{x=0}^{7}\sum_{y=0}^{7} p(x,y,t) \quad 0 \le t \le 7 \quad (8)$$
$$\frac{\cos(2x+1)u\pi}{16}\times\frac{\cos(2y+1)v\pi}{16}$$

where $p(x,y,t)$ is the pixel value at position (x,y,t) in the cube and $F(u,v,t)$ is the transformed coefficients.

3.1 Mode 1: 2-D DCT

The selection between 3-D DCT and the approximation algorithm 2-D DCT is made by computing Normalized Pixel Difference (NPD) of DC coefficient and a few AC coefficients and by comparing NPD with the threshold T_1. The NPD is calculated between the first and the other seven time-dependent 8×8 blocks within the cube as

$$NPD(t) = \frac{1}{4}\sum_{u=0}^{1}\sum_{v=0}^{1}\left|\frac{F(u,v,t)-F(u,v,0)}{Q_1(u,v)}\right| \quad (9)$$

$$\forall t \in \{1,2,\ldots,7\}$$

where $Q_1(u,v)$ is the quantization for the 2-D DCT block.

If one of the NPD values exceeds the threshold, which means that the cube is not constant enough to be approximated, the 3-D DCT is selected. Otherwise, only the first block is encoded. In the decoding side, the other seven blocks are retrieved by duplicating the first decoded.

3.2 Mode 2: Resizing of 3-D DCT

Mode 1 is only suitable for the cubes with very low motion activity. For other cubes, use of the resizing algorithm can remove efficiently the temporal correlations. The resizing of 3-D DCT is described as follows

1) We continue the transform along temporal dimension on the first four blocks and the last four blocks, respectively.

$$F_1\,(u,v,w) = C(w,4)\sum_{t=0}^{3} F(w,v,t)\frac{\cos(2t+1)w\pi}{8} \quad (10)$$

$$F_2(u,v,w) = C(w,4)\sum_{t=4}^{7} F(w,v,t)\frac{\cos(2t+1)w\pi}{8} \quad (11)$$

2) As aforementioned, two conditions have to be satisfied to ensure that the resizing algorithm improve the compression efficiency. This is done by comparing NPD between the two cubes and checking the high-frequency information.

$$NPD = \frac{1}{8}\sum_{u=0}^{1}\sum_{v=0}^{1}\sum_{w=0}^{1}\left|\frac{F_1^Q(u,v,w)-F_2^Q(u,v,w)}{Q_2(u,v,w)}\right| \quad (12)$$

$$\forall u,v \in \{0,1,\ldots,7\}, 0 \le w \le 3 \; and \; i = 1,2$$

and

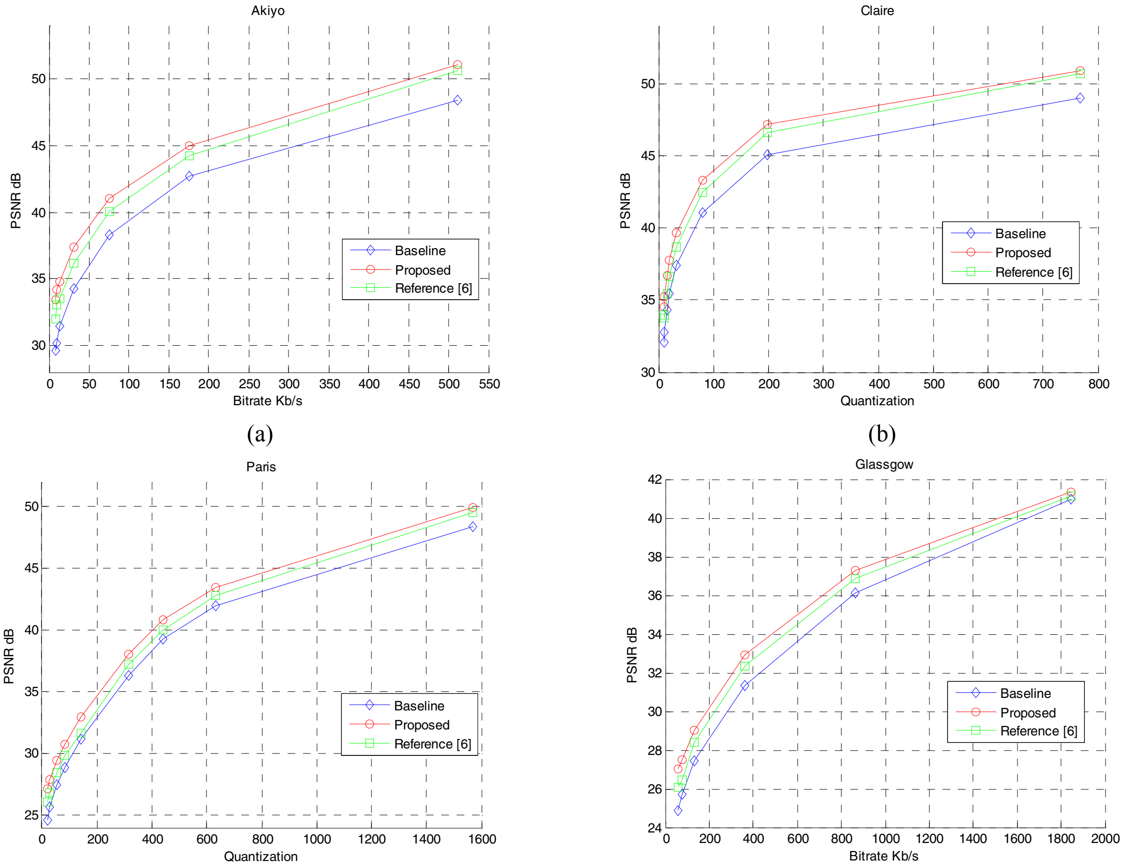
$$F_i(u,v,w)/Q_2(u,v,w) = 0 \quad (13)$$

$$\forall u,v \in \{0,1,\ldots,7\},\; i=1,2, w=2,3$$

3) If both NPD is smaller than the threshold T_2 and (13) is fulfilled, we apply the proposed resizing algorithm in the temporal dimension to the first blocks of each cube. Therefore, the $8 \times 8 \times 8$ cube is downsized to a new $8 \times 8 \times 4$ transformed cube.

3.3 Mode 3: 8× 8×4 3-D DCT

If mode 1 and 2 are not fulfilled, it indicates that the cube contains high local motion activity. In this case, smaller cube size in temporal dimension is usually better for coding efficiency since the temporal pixels are not highly correlated. Therefore, if one of the two conditions in (7) and (8) is not fulfilled, we directly encode the two $8 \times 8 \times 4$ cubes F_1^Q and F_2^Q instead of the $8 \times 8 \times 8$ cube.



(a) (b) (c) (d)

Fig. 4 The PSNR versus Bitrate for (a) Akiyo, (b) Claire, (c) Paris and (d) Glasgow based on three different codec: the baseline codec, the proposed codec and the reference codec [6]

4. Experimental results

The proposed 3-D DCT model was tested against the baseline 3-D DCT codec and the reference codec [6]. Video sequences with various motion activities were encoded and decoded. The Peak Signal to Noise Ratio ($PSNR$) versus bitrates curves are plotted based on the experimental results. The quantization strategy in accordance with [9] is defined as

$$q\overrightarrow{[k]} = \lfloor Q_p(1 + k_1^p + k_2^p + k_3^p) \rfloor \qquad (14)$$

where $q\overrightarrow{[k]}$ is the quantization step for 3-D DCT coefficient at position $\vec{k}[k_1, k_2, k_3]$, Q_p is the quantization parameter, and p is the parameter to control the rate of delay in the quantization volume. Here, it is fixed to 0.3. For the 2-D DCT block, the value of k_3 is zero.

The selection of the thresholds T_1 and T_2 is empirically determined. In the experiments, different thresholds are tested using various video materials. Finally, the constant thresholds of $T_1 = 2$ and $T_2 = 8$ usually gives the best results.

Experimental results show that the proposed model achieves the best coding performance than the baseline codec and the reference codec. Best improvement can be expected for sequences with low motion activity at low bitrates. Fig.4 shows the PSNR versus Bitrates curves of the four sequences. Based on the obtained results, the proposed model significantly improves the compression efficiency over the baseline codec. In addition, since the proposed model allows Mode 1 to be adaptively determined relying on the quantization, it is more efficient in mode selection and outperforms the reference codec in terms of compression performance.

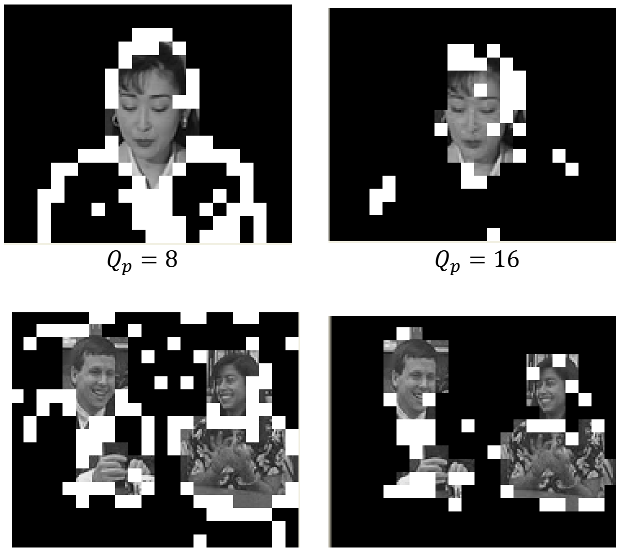
Table I shows the utilization ratio of the three modes in the proposed codec. Since the mode decision is partially dependent on quantization, the utilization ratio adaptively changes with quantization. Fig.5 shows the utilization of the three modes in frame 63 of Akiyo and 136 of Paris at different quantization. The utilization ratio of the three modes is adaptive to the change of quantization. More cubes are usually processed as the first mode with the increasing of quantization, thus the proposed model gives a higher coding efficiency at low bitrates.

Based on our previous analysis, the first mode of 2-D DCT requires the least complexity and the second mode of the resized transform is most complicated. Therefore, the overall complexity of the transformation in the reference codec and the proposed model is determined by the utilization ratio of the three modes. Experiments show that the proposed algorithm achieves a lower complexity than the reference codec. The required

Table 1 Utilization Ratio of Proposed Modes (%)

Sequence	**Mode type**	**Q_p - quantization**			
		4	**8**	**16**	**32**
Akiyo	1	57.30	78.19	83.21	89.87
	2	6.48	9.32	8.66	7.90
	3	36.22	12.49	8.13	2.23
Claire	1	9.15	45.94	77.09	85.65
	2	36.79	34.49	8.57	6.14
	3	54.06	19.5	14.35	8.21
Paris	1	6.05	23.49	60.01	67.88
	2	21.45	44.38	13.69	9.31
	3	72.50	32.12	26.30	22.81
Glasgow	1	3.37	13.97	29.93	33.89
	2	9.82	15.96	10.98	15.47
	3	86.81	70.07	59.08	50.64

number of MUL and ADD is compared in Fig.6. for the sequences of Akiyo and Glasgow. Based on the experimental results, the two codecs have almost the same complexity at high bitrates, but the proposed model outperforms the reference codec at low bitrates.



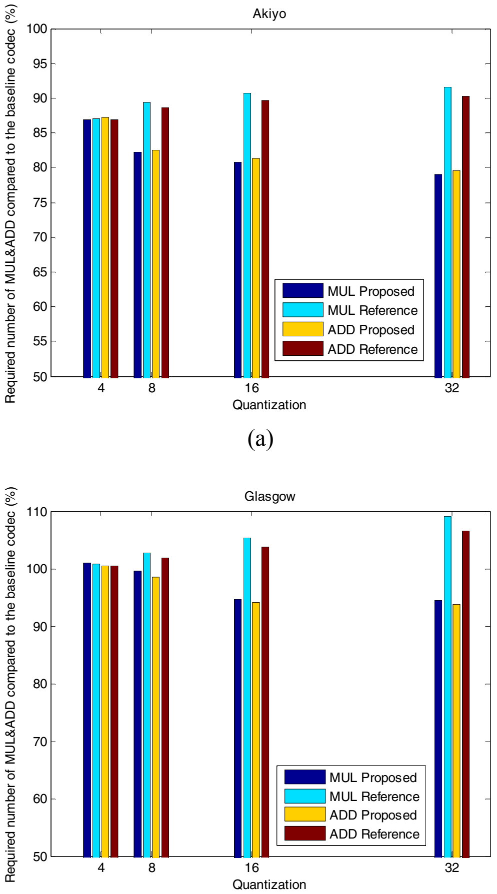
$Q_p = 8$ $Q_p = 16$

$Q_p = 8$ $Q_p = 16$

Fig. 5 Utilization of proposed modes in the frame 63 of Akiyo and the frame 136 of Paris, Black: Mode 1, White: Mode 2 and Other area: Mode 3

5. Conclusions

An adaptive DCT-to-DCT resizing algorithm is proposed for the 3-D DCT video coding to improve the coding efficiency and reduce the complexity. The proposed model adaptively resizes the DCT cube depending on the local motion activity. Experiments show that the proposed algorithm achieves the best results in terms of coding performance. In addition, since the inverse transform and the computations in the spatial domain are not required in the proposed resizing approach, the complexity is reduced. Although the compression efficiency is still lower than H.264, the encoding process of 3-D DCT is much faster. This makes the 3-D DCT video codec especially suitable for low-power processors. Potential applications could be for portable digital devices and the real-time applications.



(a)

(b)

Fig. 6 The MUL and ADD comparison among the proposed model, the reference codec and the baseline codec for (a) Akiyo and (b) Glasgow.

6. Acknowledgement

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence Program (2006 - 2011), the Academy of Finland Grant 117065 and by the Chinese Science & Technology Ministry Grant 2005DFA10300.

7. References

[1] N.P. Sgouros, S.S. Athineos, P.E. Mardaki and *etc.*, "Use of an Adaptive 3D-DCT Scheme for Coding Multiview Stereo Images," IEEE *Proceedings of ISSPIT*, pp.180-185, 2005.

[2] Y.L. Chan and W.C. Siu, "Variable Temporal-length 3-D Discrete Cosine Transform Coding," IEEE *Transactions on Image Processing*, VOL 6, No. 5, pp. 758-763, 1997.

[3] J.J. Koivusaari, and J.H. Takala, "Simplified Three-Dimensional Discrete Cosine Transform Based Video Codec," SPIE *Proc. of Multimedia on Mobile Devices*, pp. 11-21, 2005.

[4] R. Dugad and N. Ahuja, "A Fast Scheme for Image Size Change in The Compressed Domain," IEEE *Transactions on Circuits and Systems for Video Technology*, pp.461-474, 2001.

[5] J.J. Koivusaari, J.H. Takala, and M. Gabbouj, "Image Coding Using Adaptive Resizing in The Block-DCT Domain," SPIE *Proc. of Multimedia on Mobile Devices II*, pp. 1-9, 2006.

[6] J. Li, J. Takala, M. Gabbouj and H. Chen, "Use of Adaptive Resizing in 3-D DCT Domain for Video Coding," *Picture Coding Symp.*, Lisbon, Portugal, Nov.7-9, 2007.

[7] Q. Dai, X. Chen and C. Lin, "Fast Algorithms for Multidimensional DCT-to-DCT Computation Between a Block and Its Associated Subblocks," *IEEE Trans. Circuits and Syst. for Video Tech.,* Vol.13, No.7, pp.717-725, 2003.

[8] Y.S. Park and H.W. Park, "Arbitrary-Ratio Image Resizing Using Fast DCT of Composite Length for DCT-based Transcoder," IEEE *Transactions on Image Processing* 15**,** pp.494-500, 2006.

[9] N. Bozinovic, and J. Konrad, "Motion Analysis in 3D DCT Domain and Its Application to Video Coding," EURASIP *Signal Process., Image Commun.*, 2005.