Scalable Video Coding Based on Three-Dimensional Discrete Pseudo Cosine Transform

Eugeniy Belyaev[1], Timofey Sukhov[2], and Liu Kai[3]

[1] Saint-Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, 14 line 39, 199178 St. Petersburg, Russia
ebelyaev@vu.spb.ru

[2] Baltic State Technical University "Voenmeh" D.F.Ustinov, 1 Krasnoarmeiskaya street 1, 190005 St. Petersburg, Russia
tsuhov@mail.ru

[3] School of Computer Science and Technology, Xidian University, NO.2 Taibai South Road, MailBox 161, 710071 Xi'an, China
kailiu@mail.xidian.edu.cn

Abstract. This paper proposes a new spatial scalable and low-complexity video compression algorithm based on multiplication free three-dimensional discrete pseudo-cosine transform (3-D DPCT). Practical results which show the compression efficiency of the proposed algorithm in comparison with H.264/SVC standard are presented.

Keywords: 3-D DCT, scalable video coding, wireless video transmission.

Introduction

In the last few years a lot of wireless systems based on IEEE 802.11 [1], LTE [2], IEEE 802.16 [3], DVB-H [4] and other standards have appeared. One of the top task for such type of systems is real-time video transmission. Because the throughput of a wireless channels is less than raw video bit rate the additional video compression on the transmitter side is required. Taking into account high bit error ratios, packet losses and time-varying bandwidth of wireless systems, the *scalable video coding* (SVC) is more preferable as compression method for wireless video transmission.

The main idea of scalable coding is that video coder forms the bit stream from several layers: *base layer* and *enhancement layers*. The base layer is always coded independently and provide basic visual quality with low bit-rate. The enhancement layer provide refinement video quality and higher bit rate. For the next enhancement layer encoding reconstructed previous layers (that may include base layer) is needed. Therefore, in this case robust video transmission can be provided by using unequal error protection of different layers [5] and bandwidth adaptation can be easily achieved due to dropping the higher enhancement layers of the scalable video bit stream [6].

S. Balandin et al. (Eds.): ruSMART/NEW2AN 2010, LNCS 6294, pp. 448–459, 2010.

The most popular scalable video coding approach is based on extension of the H.264/AVC standard [7,9]. This extension includes temporal, spatial and quality scalability and provide high compression efficiency due to motion estimation and compensation exploiting the video source temporal redundancy and inter-layer prediction exploiting redundancy between different layers. But these methods require high computational complexity. Therefore, using of H.264/SVC codec's is difficult for real-time video compression and transmission.

In this paper we propose a scalable video coding approach based on three-dimensional discrete pseudo cosine transformation (3-D DPCT) as an alternative to H.264/SVC. This approach does not use motion estimation for exploiting of the video source temporal redundancy. Therefore, it has lower computational complexity and, in addition, it provides robust compression of the noised video sources. Therefore, proposed scalable video coding technique can be attractive for mobile video transmission.

The rest of this paper is organized as follows. Section 1 describes the multiplication free 3-D DPCT and division free quantization. Section 2 introduces video compression algorithm based on 3-D DPCT which is used for base layer coding. Section 3 presents spatial scalable video coding algorithm based on 3-D DPCT. Finally, comparison with H.264/SVC and conclusions are drawn in Section 4.

1 Implementation of Three-Dimensional Discrete Cosine Transform and Quantization

1.1 3-D DCT and Quantization

Traditionally, forward discrete cosine transform for cube $N \times N \times N$ is defined as

$$\begin{aligned} F(i,j,k) = \sqrt{\frac{8}{N^3}} c(i)c(j)c(k) \sum_{z=0}^{N-1} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x,y,z) \times \\ \times \cos\left[\frac{(2x+1)\pi i}{2N}\right] \cos\left[\frac{(2y+1)\pi j}{2N}\right] \cos\left[\frac{(2z+1)\pi k}{2N}\right], \end{aligned} \tag{1}$$

where $f(x,y,z)$ is luma or chroma value of color component with coordinates $x,y,z \in [0,...,N-1]$, $F(i,j,k)$ is transform coefficient with coordinates $i,j,k \in [0,...,N-1]$ and

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, k = 0 \\ 1, k \neq 0. \end{cases} \tag{2}$$

Inverse discrete cosine transform for cube $N \times N \times N$ is defined as

$$\begin{aligned} f(x,y,z) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \sqrt{\frac{8}{N^3}} c(i)c(j)c(k) F(i,j,k) \times \\ \times \cos\left[\frac{(2x+1)\pi i}{2N}\right] \cos\left[\frac{(2y+1)\pi j}{2N}\right] \cos\left[\frac{(2z+1)\pi k}{2N}\right]. \end{aligned} \tag{3}$$

After the 3-D DCT calculation, scalar quantization is used for each transform coefficient:

$$F_q(i,j,k) = \frac{F(i,j,k) + f \cdot q}{q}, \tag{4}$$

where q is a quantization step, $f \in [0, ..., 0.5]$.

Fast calculation of the 3-D DCT can be achieved by using the row-column-frame (RCF) approach based on 1-D DCT or by using 3D vector radix (3-D VR) approach based on partitioning an $N \times N \times N$ transform into smaller transforms, until $2 \times 2 \times 2$ transform is reached [8]. These approaches allow to calculate 3-D DCT by using $3N^3 \log_2 N$ and $\frac{14}{8} N^3 \log_2 N$ multiplications respectively and $9N^3 \log_2 -6N^3 + 6N^2$ additions.

Thus, traditional implementation of 3-D DCT and scalar quantization requires more than $N^3 \log_2 N$ multiplications and N^3 divisions.

1.2 Multiplication Free 3-D DPCT and Division Free Quantization

The further decrease of the computational complexity can be achieved by using multiplication free *three-dimensional discrete pseudo cosine transform* (3-D DPCT) and division free scalar quantization. It is necessary to notice that the same approach is used in H.264/AVC standard [9] and we propose to extend this techniques on a three-dimensional case.

Firstly, let us describe the division free implementation of the scalar quantization. In [10] it is proposed to quantize transform coefficient x in the following way:

$$X_q = \frac{x \cdot A(QP) + f \cdot 2^{20}}{2^{20}}, \tag{5}$$

and to calculate reconstructed coefficient as

$$x_r = \frac{X_q \cdot B(QP)}{2^{20}}, \tag{6}$$

where $f \in [0, ..., 0.5]$, $QP \in [0, ..., 31]$ is a quantization parameter, $A(QP)$ and $B(QP)$ are defined taking into account that $A(QP) \cdot B(QP) \approx 2^{40}$ (see Table 1). The rest of values of $A(QP)$ and $B(QP)$ for $QP > 5$ are defined taking into

Table 1. Division free scalar quantization

QP	0	1	2	3	4	5
$q(QP)$	2.5	2.8	3.2	3.5	4	4.5
$\frac{A(QP)}{676}$	620	553	492	439	391	348
$\frac{B(QP)}{676}$	3881	4351	4890	5481	6154	6914

account that $2A(QP+6) = A(QP)$ and $B(QP+6) = 2B(QP)$. Therefore, quantization can be written as [11]:

$$X_q = \frac{x \cdot A(q_M) + f \cdot 2^{20+q_E}}{2^{20+q_E}}, \tag{7}$$

and inverse quantization as

$$x_r = \frac{X_q \cdot B(q_M)}{2^{20-q_E}}, \tag{8}$$

where $q_M = QP\%6$ and $q_E = \left\lfloor \frac{QP}{6} \right\rfloor$. Thus, (7) and (8) corresponds to scalar quantization (4) with quantization step $q(QP)$ (see Table 1) and can be implemented without division.

Let us describe the multiplication free discrete cosine transform. For the sake of simplicity, let us consider the case of one-dimensional transform over a vector-column x of length 8. Then discrete cosine transform can be calculated as:

$$X = T \cdot x, \tag{9}$$

where

$$T = \begin{pmatrix} g & g & g & g & g & g & g & g \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ g & -g & -g & g & g & -g & -g & g \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{pmatrix}, \tag{10}$$

where $a = \frac{1}{2}\cos\left(\frac{\pi}{16}\right)$, $b = \frac{1}{2}\cos\left(\frac{3\pi}{16}\right)$ etc. Taking into account orthogonality properties $TT^T = I$, inverse discrete cosine transform can be calculated as:

$$x_r = T^T \cdot X. \tag{11}$$

For multiplication free implementation of discrete cosine transform in [12,13] it is proposed to use approximation of matrix T by matrix

$$H = \frac{1}{8}\begin{pmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & 10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & 4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{pmatrix}. \tag{12}$$

On the one hand, matrix H is close to matrix T. On the other hand, $H \cdot x$ and $H^T \cdot X$ can be calculated by using 32 additions and 10 shifts only (see program example in [14]). It is necessary to take into account that:

$$H^T \cdot D \cdot H = I, \tag{13}$$

where D is diagonal matrix. As a result, forward transform can be written as:

$$X = D \cdot H \cdot x = (H \cdot x) \otimes d, \tag{14}$$

where $d = D \cdot e$, e is a unit vector of length 8 and $\otimes$ means element-by-element multiplication. Inverse transform can be calculated as:

$$x_r = H^T \cdot X. \tag{15}$$

From (14) is follows that forward transform requires 8 multiplications. It is possible to avoid it by carrying over multiplications into quantization procedure (7). In this case quantization procedure is modified as following:

$$X_q = \frac{x \cdot A'(q_M, i) + f \cdot 2^{20+q_E}}{2^{20+q_E}}, \tag{16}$$

where $A'(q_M, i) = A(q_M) \cdot d_i$.

Extending the method described above, 3-D discrete pseudo cosine transform can be calculated by using the row-column-frame approach based on 1-D discrete pseudo cosine transform.

1.3 Transforms Comparison

Table 2 contains number of operations which is needed for fast calculation of 3-D DCT and 3-D DPCT. Figures 1–2 show rate-distortion comparison of these transforms by using different distortion metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). These results show that computational complexity of 3-D DPCT is significantly less than computational complexity of 3-D DCT. At the same time it has comparable rate-distortion performance. It means that three-dimensional discrete pseudo cosine transform is more preferable for real-time video compression systems.

Table 2. Transforms computational complexity for cube $8 \times 8 \times 8$

Transform	Multiplications	Divisions	Additions
3-D DCT (RCF [8]) and quantization	4608	512	11136
3-D DCT (3-D VR [8]) and quantization	2688	512	11136
3-D DPCT and modified quantization	512	0	6144

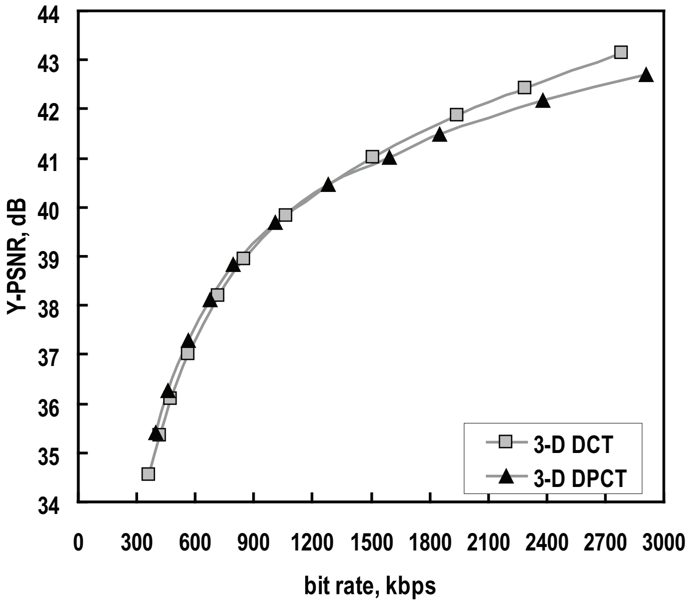

Fig. 1. Rate-distortion comparison of 3-D DCT and 3-D DPCT for test video sequence "Hall monitor" by using PSNR

2 Base Layer Coding by Using 3-D DPCT

Figure 3 shows main scheme of the base layer coding using proposed video compression algorithm based on 3-D DPCT. This scheme is typical for a single layer video compression based on 3-D DCT (see [15,16]) and can be described as follows.

First, a spatial-temporal filtration is used for input video frames denoising. Then, necessary number of video frames are accumulated in the *frame buffer* and divided into non-intersecting three-dimensional blocks (for example into cubes $8 \times 8 \times 8$).

Motion analyzer chooses compression mode for each cube as following. Let us define value of luma or chroma color component of current and previous cubes in base layer as $b(x, y, z)$ and $b'(x, y, z)$ respectively. Selection of the compression mode for cube $N \times N \times N$ is based on analyzing metrics M_1 and M_2 which are defined as:

$$M_1 = \max\{m_1(0,0), m_1(0, N/2), m_1(N/2, 0), m_1(N/2, N/2)\}, \quad (17)$$

where $m_1(a, b) = \max\limits_{z \in [0,\ldots,N-1]} \sum\limits_{x=a}^{a+\frac{N}{2}-1} \sum\limits_{y=b}^{b+\frac{N}{2}-1} |b(x, y, z) - b'(x, y, N-1)|$,

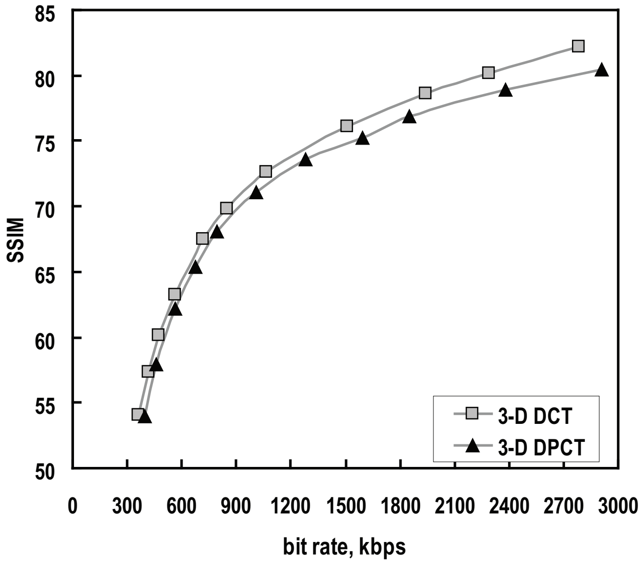

Fig. 2. Rate-distortion comparison of 3-D DCT and 3-D DPCT for test video sequence "Hall monitor" by using SSIM

$$M_2 = \max\{m_2(0,0), m_2(0,N/2), m_2(N/2,0), m_2(N/2,N/2)\}, \qquad (18)$$

where $m_2(a,b) = \max\limits_{z\in[1,\ldots,N-1]} \sum\limits_{x=a}^{a+\frac{N}{2}-1} \sum\limits_{y=b}^{b+\frac{N}{2}-1} |b(x,y,z) - b(x,y,0)|$.

If $T_1 \leq M_2 \leq T_2$, where T_1 and T_2 are algorithm parameters, then current cube is classified as a cube with slow motion. In this mode 3-D DPCT and division free quantization described in section 1.2 are used. If $M_1 < T_1$ and $M_2 < T_1$ then current cube is classified as a cube without motion (static cube). In this mode current cube is skipped and not transmitted. Otherwise, current cube is classified as cube with intensive motion. In this mode two-dimensional discrete pseudo cosine transform is used instead of 3-D DPCT. This approach allows to decrease visual artifacts for frames with scene changes. Thus, there are three compression modes for each cube:

1. Skip mode;
2. 2-D DPCT mode;
3. 3-D DPCT mode.

Rate controller chooses quantization step for each cube depending on required bit rate. Then, the quantized DCT coefficients are lossless compressed by using 3-D zig-zag scanning and context adaptive binary arithmetic coder with probability estimation by Virtual Sliding Window [17].

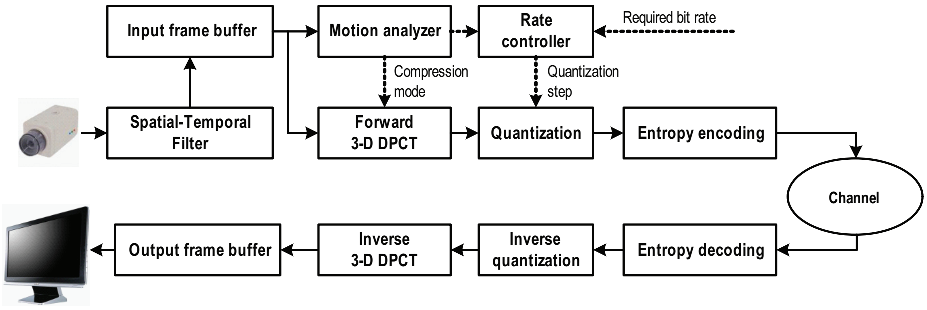

Fig. 3. Base layer video compression based on 3-D DPCT

3 Spatial Scalable Video Coding by Using 3-D DPCT

Spatial scalability allows to decode streams with the various frame resolution. Fig. 4 shows spatial scalable coding based on 3-D DPCT. For the sake of simplicity, let us consider a case of a spatial scalable coding with only two layers: base layer and enhancement layer. Then video compression algorithm can be described as follows. First, input frames are decimated by factor two per height and width. Decimated frames are accumulated in base layer frame buffer and compressed as described in section 2.

Then, enhancement layer is coded by using four cube compression modes:

1. Skip mode;
2. 3-D DPCT mode;
3. Inter-layer prediction and 3-D DPCT mode;
4. Inter-frame prediction and 3-D DPCT mode.

For increasing of enhancement layer compression efficiency we include two addition cube compression modes. On the one hand, redundancy between spatial layers is exploited by inter-layer prediction. In this case, reconstructed and scaled cube in base layer is used for prediction of the corresponding four cubes in enhancement layer. On the other hand, redundancy between frames inside enhancement layer is exploited by inter-frame prediction. In this case, last reconstructed 8×8 block of the corresponding previous cube is used for prediction of the current cube.

For fast compression mode selection in enhancement layer we propose a following heuristic approach. Let us define value of luma or chroma color component of current and previous cubes in enhancement layer as $e(x, y, z)$ and $e'(x, y, z)$ respectively. If current cube b in base layer is compressed in Skip mode, then four corresponding cubes in enhancement layer are compressed in Skip mode too. Otherwise, residual metrics R_1, R_2, R_3 for cube e are calculated:

$$R_1 = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} |e(x, y, z) - \hat{b}(x, y, z)|, \tag{19}$$

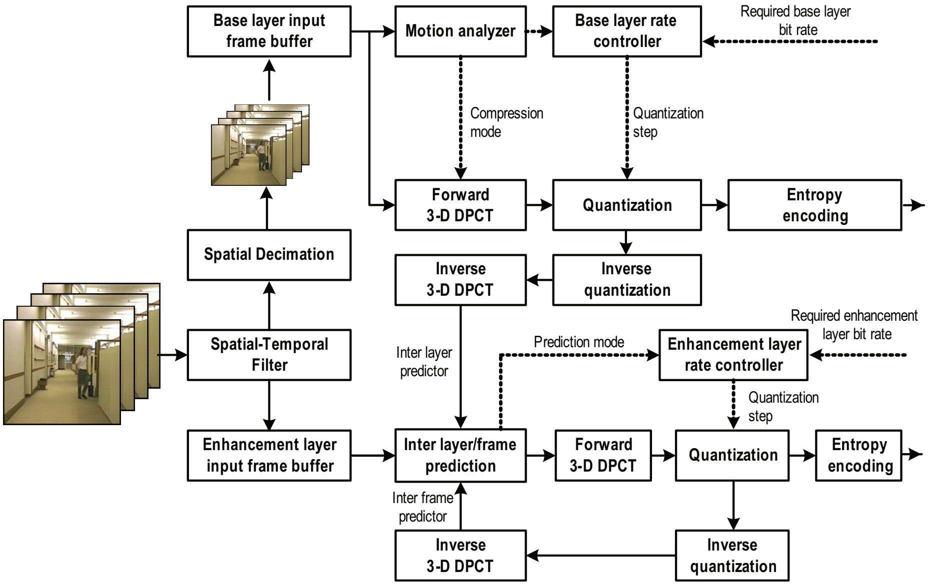

Fig. 4. Spatial scalable coding based on 3-D DPCT

where $\hat{b}(x, y, z)$ is luma or chroma color component of reconstructed and scaled corresponding cube from base layer,

$$R_2 = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} |e(x, y, z) - e'(x, y, N-1)|, \tag{20}$$

$$R_3 = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} |e(x, y, z)|. \tag{21}$$

If $R_1 < \min\{R_2, R_3\}$ then inter-layer prediction and 3D-DPCT compression mode is selected. If $R_2 < \min\{R_1, R_3\}$ then inter-frame prediction and 3D-DPCT compression mode is selected. Otherwise, 3D-DPCT without any prediction is used. The quantized DCT coefficients are lossless compressed by using 3-D zig-zag scanning, run-length coding and Huffman code tables.

4 Practical Results

For our experiments proposed spatial scalable video coding algorithm based on three-dimensional discrete pseudo cosine transform and the Joint Scalable Video Model (JSVM reference software v.9.19 [18]) which has formed spatial scalable video stream were used.

Practical results were obtained for test video sequences “Hall Monitor“ with 176×144 resolution for base layer and 352×288 resolution for enhancement layer

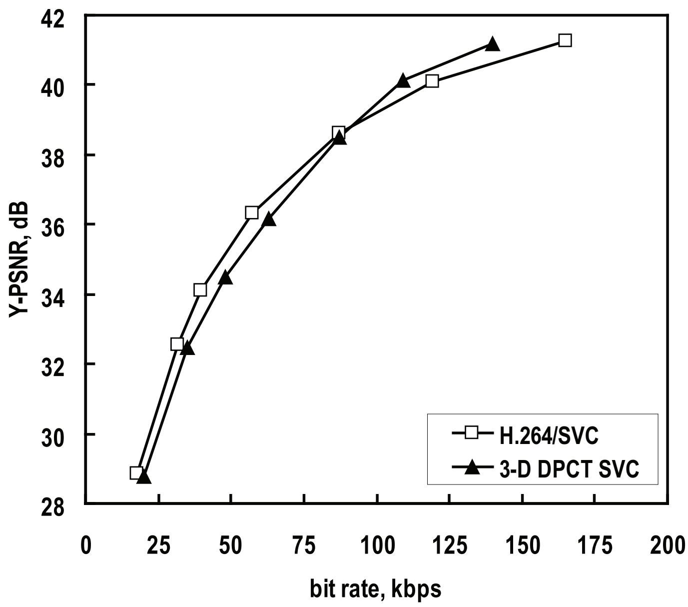

Fig. 5. Rate-distortion comparison of the base layer coding for test video sequence "Hall monitor", 176×144, 30fps

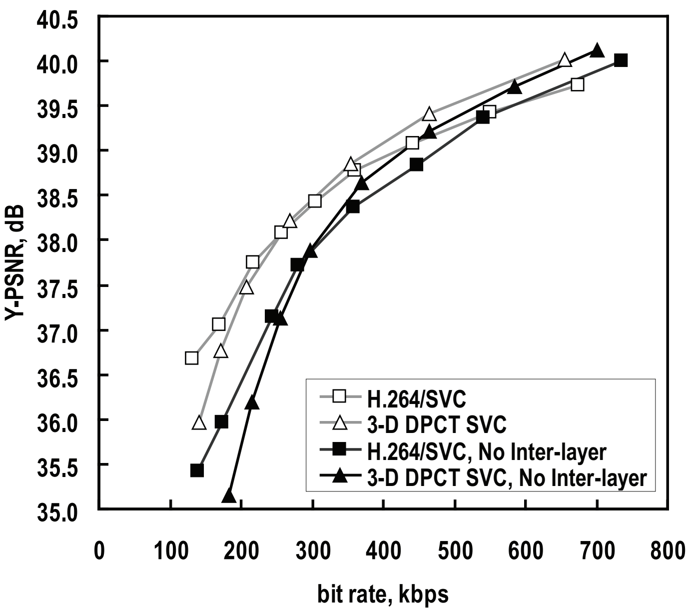

Fig. 6. Rate-distortion comparison of the enhancement layer coding for test video sequence "Hall monitor", 352×288, 30fps

and a frame rate of 30 fps. Figures 5–6 show relations between peak signal-to-noise ratio (PSNR) and bit rate for base and enhancement layers respectively. In addition, Figure 6 shows compression efficiency of inter-layer prediction in comparison to enhancement layer coding without inter-layer prediction (No Inter-layer).

Practical results show that proposed scalable video coding algorithm based on three-dimensional discrete pseudo cosine transform provide comparable compression efficiency in comparison with H.264/SVC. At the same time it has less computational complexity. Therefore, proposed video coding algorithm can be used as an alternative compression method for robust video transmission over wireless channels.

This work is supported by the Russian Foundation for Basic Research (projects 10-08-01071-a and 08-08-00403-a) as well as by the Branch of Nano- and Information Technologies of Russian Academy of Sciences (project 2.3).

References

1. IEEE P802.11n D1.0 Draft Amendment (March 2006)
2. Dahlman, E., Ekstrom, H., Furuskar, A., Jading, Y., Karlsson, J., Lundevall, M., Parkvall, S.: The 3G Long-Term Evolution - Radio Interface Concepts and Performance Evaluation. In: IEEE Vehicular Technology Conference (2006)
3. IEEE Std. IEEE 802.16-2004 (Revision of IEEE Std. IEEE 802.16 - 2001). IEEE Standard for Local and metropolitan area networks. Part 16: Air Interface Fixed Broad-band Wireless Access Systems, IEEE. 1 (2004)
4. ETSI EN 302 304 V1.1.1 (2004-11). Digital video broadcasting (DVB): transmission system for handheld terminals (DVB-H). European Telecommunications Standards Institute (2004)
5. Gallant, M., Kossentini, F.: Rate-distortion optimized layered coding with unequal error protection for robust Internet video. IEEE Transactions on Circuits and Systems for Video Technology (2001)
6. Fallah, Y.P., Mansour, H., Khan, S., Nasiopoulos, P., Alnuweiri, H.M.: A Link Adaptation Scheme for Efficient Transmission of H.264 Scalable Video Over Multirate WLANs. IEEE Transactions on Circuits and Systems for Video Technology 18(7) (2008)
7. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264 / AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology 17(9), 1103–1120 (2007)
8. Mekky, T., Boussakta, S., Darnell, M.: On the computation of the 3-D DCT. In: 10th IEEE International Conference on Electronics, Circuits and Systems, vol. 3, pp. 1141–1143 (2003)
9. Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10, AVC (2009)
10. Wiegand, T.: H. 26L Test Model Long-Term Number 9 (TML-9) draft0, ITU-T Q.6/SG16 Document TML9.doc, VCEG-N83d1 (2001)
11. Hallapuro, A., Karczewicz, M., Malvar, H.: Low complexity transform and quantization, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Doc. JVT-B038 (2002)
12. Malvar, H.S.: Low-complexity transform and quantization in H.264/AVC. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 598–603 (2003)

13. Gordon, S.: Simplified Use of 8x8 Transform, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Doc. JVT–I022 (2003)
14. Lin, Y.-K., Liao, Y.-Z., Chang, T.-S.: An area efficient design for integer transform in H.264/AVC FRext. In: The 17th VLSI Design/CAD Symposium (2006)
15. Chan, Y.-L., Siu, W.-C.: Variable Temporal-Length 3-D Discrete Cosine Transform Coding. IEEE Transactions on Image Processing 6(5), 758–763 (1997)
16. Servais, M.P., De Jager, G.: Video Compression using the Three Dimensional Discrete Cosine Transform. In: Proceedings of the South African Symposium on Communications and Signal Processing, pp. 27–32 (1997)
17. Belyaev, E., Gilmutdinov, M., Turlikov, A.: Binary arithmetic coding system with adaptive probability estimation by Virtual Sliding Window. In: Proceedings of the 10th IEEE International Symposium on Consumer Electronics, pp. 194–198 (2006)
18. JSVM 9.19 Software Package, `http://iphome.hhi.de/`