A Complete Video Coding Chain Based on Multi-Dimensional Discrete Cosine Transform

Tomas FRYZA

Department of Radio Electronics, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic

fryza@feec.vutbr.cz

Abstract. *The paper deals with a video compression method based on the multi-dimensional discrete cosine transform. In the text, the encoder and decoder architectures including the definitions of all mathematical operations like the forward and inverse 3-D DCT, quantization and thresholding are presented. According to the particular number of currently processed pictures, the new quantization tables and entropy code dictionaries are proposed in the paper. The practical properties of the 3-D DCT coding chain compared with the modern video compression methods (such as H.264 and WebM) and the computing complexity are presented as well. It will be proved the best compress properties could be achieved by complex H.264 codec. On the other hand the computing complexity - especially on the encoding side - is lower for the 3-D DCT method.*

Keywords

Video signal processing, data compression, discrete cosine transform, multidimensional coding, digital signal processors.

1. Introduction

Video compression techniques have generally two objectives: to reduce the spatial redundancy among the picture elements and to reduce the temporal redundancy between successive frames, i.e. interframe coding. The main interframe coding principle is the predictive coding, which is used in all major standard video codecs, such as H.261, H.263, MPEG-1, MPEG-2 and MPEG-4 [7], [8], [16], [19]. Video compression with the 3-D DCT (Three-Dimensional Discrete Cosine Transform) contains both types of correlation reduction in a single transform coding. This transform is a block based method and requires storage of N picture frames but can provide an acceptable picture quality with high compression ratio [6], [15]. The basic structure of the 3-D DCT encoder and decoder is known and it was published for example in [15] or [21]. Several improvements mainly in quantization and entropy coding blocks for 2-D and 3-D coefficients were proposed in [1], [11] or [12]. Fast methods for DCT and Huffman coding algorithms were published as well in [5], [13] or [22]. Also the testing and comparison of the 3-D DCT properties of several image and video compression codecs could be found in literature, such as [2] or [21]. The contribution of the paper is to present a new Huffman dictionary, then the 3-D DCT implementation in digital signal processor and testing of the method with comparison to other codecs, mainly to H.264 and VP8 from the new WebM video container [20].

The paper is divided into four major parts. The main emphasis is focused on the structure of the entire coding chain with a description of each coding block (Section 2). In the second part, Section 3, the fast algorithms for 3-D DCT calculation are presented. The experimental verification of the coding chain along with other video codecs comparison is introduced in Section 4. The results discussion is depicted in Section 5.

2. Encoder Structure

The basic 3-D DCT encoder and decoder could be derived from the JPEG coder [18] and has only three main parts, as shown in Fig. 1: transform coding, frequency coefficients processing and entropy coding. The input of the 3-D DCT encoder f is formed as a video cube [15]. Video cube is a technical term for 3-D matrix, with dimension of N made up of small elements of particular frames with the dimensions of $N \times N$ pixels. The third (temporal) dimension of the cube is commonly of the same size as the first (horizontal) and the second (vertical) dimension. The reason is the way of fast implementing method of the 3-D algorithm and the entropy coding (process of implementation with variable video cube dimension will be described later). Another reason is the reachable compress ration: the smaller N in temporal dimension, the smaller compress ratio.

For a color sequence three independent coders are used: one for the luminance signal Y and two for the chrominance signals C_b and C_r, respectively. Each signal is transformed into frequency domain using the 3-D DCT forward transform. The values of the frequency coefficients D are subsequently reduced by quantization (D^Q) and in some cases by thresholding as well (D^{QT}). Non-zero coefficients are finally encoded by an entropy coder and formed into an output bit stream F (see Fig. 1).

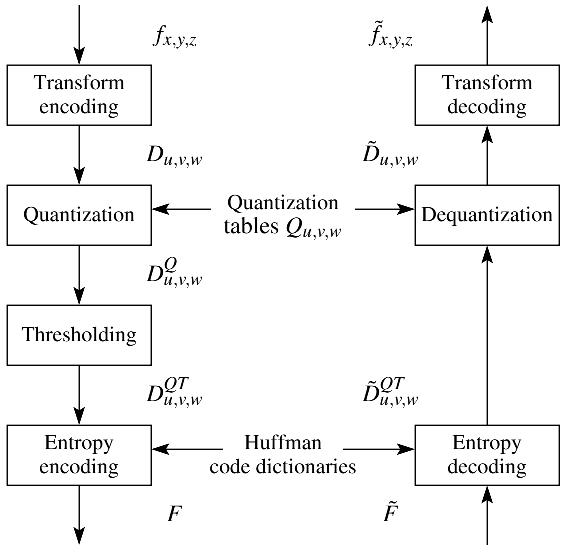


Fig. 1. Video coding chain based on 3-D discrete cosine transform.

For video cube dimensions of N, the forward 3-D DCT is defined in the following way [15]

$$D_{u,v,w} = \gamma_u \gamma_v \gamma_w \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} f_{x,y,z} \cdot \tag{1}$$
$$\cos \frac{\pi u(2x+1)}{2N} \cdot \cos \frac{\pi v(2y+1)}{2N} \cdot \cos \frac{\pi w(2z+1)}{2N}$$

where $D_{u,v,w}$ represents 3-D DCT coefficient of a picture element $f_{x,y,z}$ while $u,v,w = 0,1,\ldots,N-1$. The constants γ could be expressed as follows

$$\gamma_{u,v,w} = \begin{array}{lcl} \sqrt{1/N} & : & u,v,w = 0 \\ \sqrt{2/N} & : & u,v,w \neq 0. \end{array} \tag{2}$$

The coefficient with coordinates of $(0,0,0)$ is called the DC (Direct Current) coefficient and all others are called AC (Alternating Current) coefficients. It can be seen the total number of the 3-D DCT coefficients is equal to the number of input pixels. It can be proved that the number of bits for representing the frequency coefficients is higher than for representing pixels in the time domain. The adequate bit number for a real DC coefficient is equal to 13 bits and equal to 12 bits for an AC coefficient; i.e. symmetric $\mathrm{DC} \in \langle -4{,}095; 4{,}095 \rangle$ and $\mathrm{AC} \in \langle -2{,}047; 2{,}047 \rangle$. The effect of the transform coding is in reducing correlation between picture elements. The main energy of transformed values is centralized into the low frequency coefficients. It is known that fast varying sequences with high details have increased number of non-zero coefficients. An example of AC coefficient distribution of a slowly and fast varying sequences are shown in Fig. 2 and in Fig. 3, respectively.

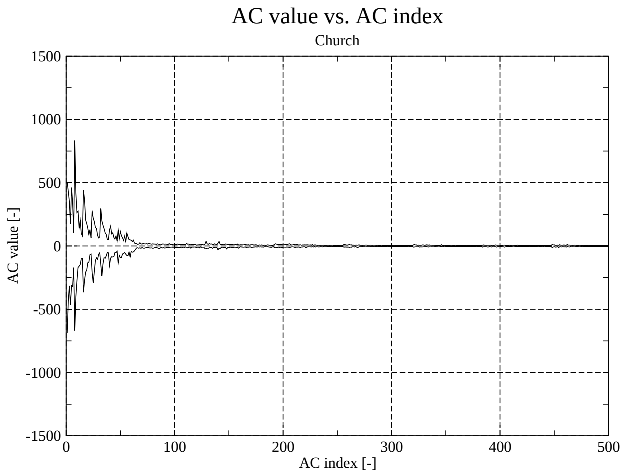


Fig. 2. Distribution of the AC coefficients for a slowly varying sequence.

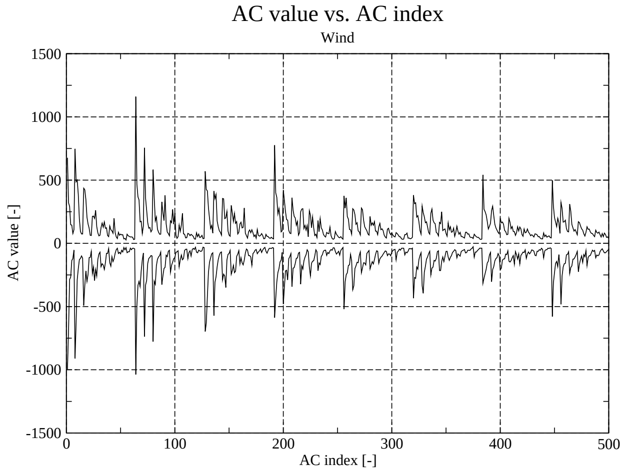


Fig. 3. Distribution of the AC coefficients for a fast varying sequence.

An inverse 3-D DCT transforms the coefficients from frequency to time domain and could be defined in the following form [15]:

$$\tilde{f}_{x,y,z} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{w=0}^{N-1} \gamma_u \gamma_v \gamma_w \cdot \tilde{D}_{u,v,w} \cdot \tag{3}$$
$$\cos \frac{\pi u(2x+1)}{2N} \cdot \cos \frac{\pi v(2y+1)}{2N} \cdot \cos \frac{\pi w(2z+1)}{2N}.$$

The notation $\tilde{D}$ corresponds to the frequency coefficients modifications during the quantization and thresholding process, respectively and the erroneous transmission of encoded sequences.

In the following text, the remaining coder blocks will be described. In Subsection 2.1 the quantization followed by thresholding are outlined. The entropy coding is introduced in Subsection 2.2.

2.1 Quantization and Thresholding

According to the encoder block scheme, the second part of the encoder includes a quantization. There are two objectives of this operation: the first is to reduce the dynamic range of the coefficients and the second is to decrease the number of insignificant coefficients, thus to increase the compress ratio. This aim can be described by the following equation

$$D^{Q}_{u,v,w} = \left\lfloor \frac{D_{u,v,w}}{Q_{u,v,w}} \right\rfloor \tag{4}$$

where $D^{Q}_{u,v,w}$ is a quantized 3-D DCT coefficient, $Q_{u,v,w}$ is a quantization value from interval $\langle 1; 255 \rangle$ and operation $\lfloor \cdot \rfloor$ represents rounding down. In general, the quantization causes lossy processing. The greater the quantization value, the higher is the impact in decompressed video sequence quality.

Another common way of reducing non-zero coefficients is thresholding. This operation can be expressed as follows

$$D^{QT}_{u,v,w} = \begin{cases} D^{Q}_{u,v,w} & : \quad |D^{Q}_{u,v,w}| \geq ThrL \\ 0 & : \quad |D^{Q}_{u,v,w}| < ThrL \end{cases} \tag{5}$$

where $D^{QT}_{u,v,w}$ is a 3-D DCT coefficient after thresholding and $ThrL$ is the threshold level. Evidently, the thresholding is a lossy operation as well. It can be proved that quantization, followed by the process of thresholding represents a main tool how to affect the quality of a video sequence and at the same time determine the final compression ratio [6].

2.2 Entropy Coding

The last part of the encoder is an entropy coding. Similarly to still picture compression [18], the Huffman coding is used. Therefore all coefficients from each block are read in a modified zig-zag way, where the major coefficients along axes u, v and w are read first. For the DC coefficients, differential coding is used. For the remaining AC coefficients only non-zero coefficient values are taken into account and are encoded together with a number of preceding zero coefficients, i.e. run length coding is used [11], [12], [13]. For 3-D DCT coefficients, the new code dictionaries were derived.

As mentioned above, the DC coefficient interval is from 0 to $\pm$4,095. The entire interval is divided into 13 subintervals, the first subinterval containing only value 0, the second one containing values -1 and 1, the third one values -3, -2, 2, 3, etc. Each DC coefficient is first identified by its interval and then encoded by its position within the interval. For each subinterval the unique codeword is suggested. Let the interval $|2 \div 3|$ be represented by the codeword `001`, then the entropy code for DC value -3 is `00100`, for -2 `00101`, etc.

As mentioned above, within one block (cube) the AC coefficients are reordered using the modified zig-zag scanning, followed by run-length coding and finished by an end-of-block identifier (EOB). Assuming that $N = 8$, one cube contains exactly 511 AC coefficients. Similarly to DC coefficients, the whole interval of possible values is divided into 11 subintervals and theoretically, run-length values vary between 0 and 510. Thus the proposed code dictionary for the AC coefficients contains $511 \times 11 + \text{EOB} = 5{,}622$ unique codewords. The outlook of the particular codewords for AC coefficients with run-length equal to 0 is shown in Tab. 1. According to the extensive code dictionary, the less common AC values are represented by wide codewords. For example, a codeword for subinterval $|256 \div 511|$ is 15 bits wide, for $|512 \div 1{,}023|$ 26 bits wide and for the subinterval of $|1{,}024 \div 2{,}047|$, 27 bits are needed. In output bit stream, each codeword is followed by the number identifies the coefficient value within the subinterval. It is apparent, provided various temporal dimensions in video cube, the Huffman code dictionary for AC coefficients and scanning order have to be modified.

Interval/Value	New codeword	Bits for new codeword + value
EOB	`001000`	6 + 0
$\lvert 1 \rvert$	`01`	2 + 1
$\lvert 2 \div 3 \rvert$	`000`	3 + 2
$\lvert 4 \div 7 \rvert$	`110`	3 + 3
$\lvert 8 \div 15 \rvert$	`1001`	4 + 4
$\lvert 16 \div 31 \rvert$	`10100`	5 + 5
$\lvert 32 \div 63 \rvert$	`0010011`	7 + 6
$\lvert 64 \div 127 \rvert$	`10111000`	8 + 7
$\lvert 128 \div 255 \rvert$	`00110001100`	11 + 8
$\lvert 256 \div 511 \rvert$	`00110101001..`	15 + 9
$\lvert 512 \div 1{,}023 \rvert$	`00110001101..`	26 + 10
$\lvert 1{,}024 \div 2{,}047 \rvert$	`00110001101..`	27 + 11

Tab. 1. Proposed codewords for AC coefficients with run-length equal to 0.

3. Transform Calculation

This section is dedicated to calculation of the transform coding. It can be shown that the transform coding is the most time-consuming operation of the entire coding chain [5]. Despite the multi-dimensional DCT is applied in the proposed coding chain, the subsequences of 1-D transforms are commonly used for real time processing [5], [8]. Two common dimensions of transform base are mainly considered: $N = 8$ and $N = 4$. The algorithm for evaluating 8-point discrete cosine transform is recapitulated in Subsection 3.1. Subsection 3.2 describes the algorithm for 4-point transform.

3.1 8-point Discrete Cosine Transform

Let $N = 8$. According to [8], 5 product and 29 sum operations have to be performed in order to evaluate eight 1-D coefficients. In every video cube with $N = 8$, the 1-D transform has to be repeated 192 times to obtain 512 frequency coefficients. Imagine a test grayscale video sequence with dimensions of 720×576 picture elements and length of 24 frames. The minimal number of arithmetical operations for encoding such sequence is 18,662,400 products and 108,241,920 sums [5].

3.2 4-point Discrete Cosine Transform

According to (1), one-dimensional 4-point forward discrete cosine transform could be expressed as follows

$$D_u = \gamma_u \cdot \sum_{x=0}^{3} f_x \cdot \cos \frac{\pi u(2x+1)}{8} \qquad (6)$$

where $u = 0, 1, \ldots, 3$. Applying the similar procedures mentioned in [8], the set of modified equations for fast calculations of 4-point 1-D DCT could be evaluated. Using the extension of 4 input samples in term $f_x = f_{7-x}$ and using the Discrete Fourier Transform, the 1-D DCT could be expressed by (7).

$$\gamma_u \cdot \sum_{x=0}^{3} f_x \cdot \cos \frac{\pi u(2x+1)}{8} = \frac{\Re\{F_u\}}{2 \cdot \cos(\frac{\pi u}{8})} \qquad (7)$$

where $\Re\{F_u\}$ represents a real part of Fourier coefficient. The left part of (7) is equal to the one-dimensional DCT. Therefore, 1-D transform could be evaluated via real parts of Fourier coefficients divided by the real constant $2 \cdot \cos(\frac{\pi u}{8})$. Like the γ_u values, the constant may be incorporated in the quantizer block as well. Hence, the only task is to enumerate the real parts of F_u. It could be done by a set of equations as defined below.

$$\begin{aligned} \Re\{F_0\} &= 2 \cdot (f_0 + f_1 + f_2 + f_3) \\ \Re\{F_1\} &= f_0 - f_3 + \cos \frac{2\pi}{8} \cdot (f_1 - f_3 - f_2 + f_0) \\ \Re\{F_2\} &= f_0 - f_2 + f_3 - f_1 \\ \Re\{F_3\} &= f_0 - f_3 - \cos \frac{2\pi}{8} \cdot (f_1 - f_3 - f_2 + f_0). \end{aligned} \qquad (8)$$

It can be seen that the number of necessary arithmetical operation for 4-point 1-D DCT is therefore 1 product and 9 sums. Nevertheless, according to the smaller blocks of input samples encoded in one moment, the total number of operations for transforming the test video sequence from Subsection 3.1 (720×576×24) is 7,464,960 products and 67,184,640 sums, i.e. a lower number.

4. Experimental Verification

4.1 Compression Properties

Six color video sequences with different characteristics were proposed for the simulation. Dimensions of all are 320×240 pixels and they contain 96 frames. Two extreme contents of video sequences were applied. First of them, titled Church, represents a slowly varying sequence with large areas of similar shades. The second extreme is represented by a fast varying sequence named Wind, where approximately 80% of the frame content is steadily changing. The other tested sequences represent real scenarios between these two extremes. The initial frames of selected video sequences are shown in Fig. 5(a)–Fig. 8(a).

Objective properties of all video sequences were enumerated in time domain by *SFM* (Spatial Frequency Measure) and in frequency domain by *SAM* (Spectral Activity Measure) parameter. According to [9] *SFM* and *SAM* for a luminance component of single video frame are defined as follows:

$$\begin{aligned} SFM &= \sqrt{R+C}, \\ R &= \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=1}^{N-1} \{f_{i,j} - f_{i,j-1}\}^2, \\ C &= \frac{1}{MN} \sum_{j=0}^{N-1} \sum_{i=1}^{M-1} \{F_{i,j} - f_{i-1,j}\}^2, \end{aligned} \qquad (9)$$

$$SAM = \frac{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |F_{i,j}|^2}{\prod_{i=0}^{M-1} \prod_{j=0}^{N-1} |F_{i,j}|^{\frac{2}{MN}}} \qquad (10)$$

where R represents row frequency and C is column frequency, $f_{i,j}$ are image/video pixels, and M and N are numbers of pixels in horizontal and vertical directions. $F_{i,j}$ are DFT coefficients of video frame. Calculated values for original video sequences, followed by values for 3-D DCT compressed sequences with bitrate of 1,000 kbits/s are shown in Tab. 2.

	Original Sequence		3-D DCT (1 Mbits/s)	
Sequence	*SFM* [-]	*SAM* [-]	*SFM* [-]	*SAM* [-]
Church	10.74	3,574	10.77	3,661
High Jump	27.91	144	27.81	144
Highway	26.65	290	26.57	291
Road	35.46	264	35.65	263
Sprint	24.63	378	24.25	388
Wind	46.28	114	40.79	152

Tab. 2. Comparison of test video sequences by *SFM* (Spatial Frequency Measure) and *SAM* (Spectral Activity Measure).

It is obvious for natural images and video sequences larger values of *SAM* imply smaller values of *SFM* parameter. Moreover, large values of *SAM* mean higher predictability and therefore these images and video sequences are in general easy to encode. On the other hand, sequences with small values of *SAM* parameter are difficult to encode. Verification of these hypotheses will be shown later in the text. From Tab. 2 it can be seen as well the changing of both parameters is insignificant when compress ratio (i.e. bitrate) does not cause perceptible degradation in images.

For evaluating the compression properties of the proposed video coding chain, a considerable number of simulation cases was performed. The main settings were altered within the quantizer block. With the aid of different quantization rates, a set of dissimilar compression ratios and picture qualities was achieved.

In order to compare the compression quality, two common criteria for objective evaluation were used: output bitrate and metric of *PSNR* (Peak Signal-to-Noise Ratio). A practical result of an output bit stream compared to picture quality for different sequences can be seen in Fig. 4. It can be seen that the slowly varying sequence Church (see Fig. 4(a)) presents a high picture quality even at low bitrates. On the other hand, the fast varying sequence Wind can be encoded at a bitrate of around 400 kbits/s only due to severe degradation of quality. Samples of encoded sequences by 3-D DCT with bitrates of 300 and 500 kbits/s can be seen in Fig. 5(c)–Fig. 8(c).

The second part of the 3-D DCT experimental verification was concentrated on a comparison with other video compression methods and codecs. For the first approach video formats MPEG-2, XviD, H.264 [10] and WebM [20] were chosen. All encoding processes were executed under Linux with help of freeware program FFmpeg from FFmpeg Team [4] with implementations of mpeg2video, libx264 and libvpx codecs. For the MPEG standard and H.264 the *BBIBBPBBPBBP* group of pictures were applied. Sixteen bitrates were used (200–2,400 kbits/s) for six video sequences. For objective quality testing, the *PSNR* (Peak Signal-to-Noise Ratio) metric of luminance component was measured. The results are graphically displayed in Fig. 4 and concrete values of the selected sequences are shown in Tab. 4.

The best proportion between bitrate and *PSNR* for most types of video sequences can be seen if the H.264 video format has been applied. Only for video sequences Church and Wind, i.e. for very slowly and fast varying sequences, the proposed 3-D DCT coder is characterized by excellent *PSNR* values. The remaining results of 3-D DCT are comparable with the XviD codec. The advantage of the 3-D DCT coder is its simple structure and single transform. The worst values in every test cases were achieved by the MPEG-2 standard and amazingly by the WebM as well.

4.2 Algorithm Complexity

The last part of experimental verification was focused on real time processing possibilities of evaluated transform coding. As mentioned in Section 3, the most time-consuming component of the coding scheme is the transform coding itself. According to that, only the calculation of 3-D DCT was considered as a criterion for real time processing.

The algorithms mentioned in Subsections 3.1 and 3.2 were programmed for Texas Instruments's floating-point digital signal processor TMS320C6713 in so-called linear assembly. The linear assembly is an interstage between high level C language and low level assembly code [3], [17].

The total numbers of the necessary CPU cycles needed for 8-point and 4-point 1-D DCT are shown in Tab. 3. The estimated times for encoding a grey scale video sequence with dimensions of 720×576×24 are shown as well. Note: the frequency of the clock signal was $f_{clk} = 150$ MHz. The parameters -o0, -o1, -o2 and -o3 correspond to the level of source code optimization. Parameter -o0 enables register level optimization, -o1 and -o2 start function level optimization and parameter -o3 corresponds to optimization on file level. The optimization could be done by the Code Composer Studio development software from Texas Instruments [14].

	$N = 8$		$N = 4$	
Param.	Cycles [-]	Time [s]	Cycles [-]	Time [s]
no opt.	10,284	4.00	1,054	3.28
-o0	10,284	4.00	1,054	3.28
-o1	5,206	2.02	702	2.18
-o2	2,144	0.83	417	1.30
-o3	2,144	0.83	417	1.30

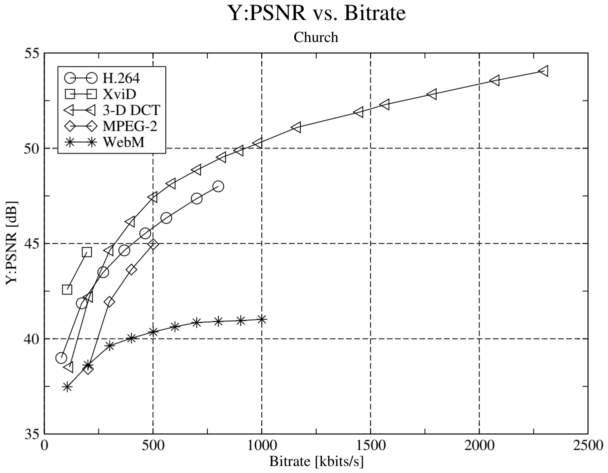
Tab. 3. CPU cycles for transforming 1 video cube and duration of transforming 1 sec. of video sequence.

It can be seen the only possibility of encoding a grey scale video sequence (with $f_s = 24$ frames/s) in real time in our case is using the 8-point fast algorithm followed by the optimization tools with maximal optimization level. The problem could be solved by applying a higher clock frequency f_{clk} or using a digital signal processor with parallel data processing, e.g. TMS320C6455 or TMS320C6457 (fixed-point processors with $f_{clk} = 1.2$ GHz).

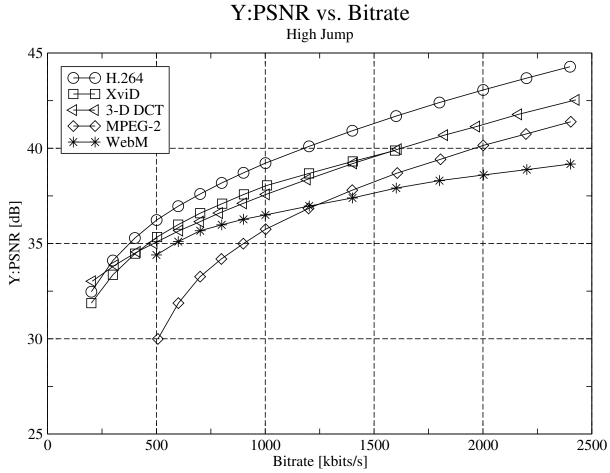
5. Conclusion

The paper was focussed on video signal compression and especially on application of multi-dimensional discrete cosine transform. The coding chain based on 3-D DCT was derived from coder structure for still pictures compression. The coding chain contains three main parts. The first part is the transform coding itself, where the input picture samples are transformed into frequency domain. The second part of the encoder is dedicated to the frequency coefficient modifications, i.e. to quantization and thresholding processes. In the experimental section the substantial effect of this part on the compression ratio and particularly on picture quality was shown. The last part of the encoder is the run-length coding followed by lossless entropy coding. The Huffman coding was used. Considering the specific characteristics of 3-D transform, the modified quantization tables and above all the new Huffman code dictionaries were developed and tested.

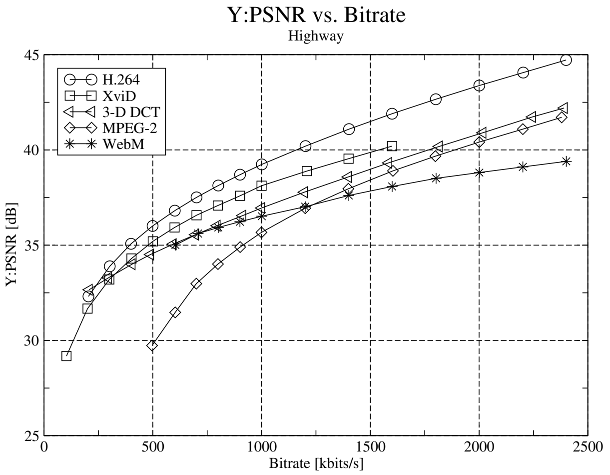
The mathematical complexity of the coder was discussed. The transform coding was evaluated by two methods with different values of N. The results from DSP implementation were presented for $N = 8$ and $N = 4$. It was shown the simpler 4-point algorithm could achieve better results only in the case when no optimizer tools were used. By selected DSP ($f_{CPU} = 150$ MHz), the real-time processing



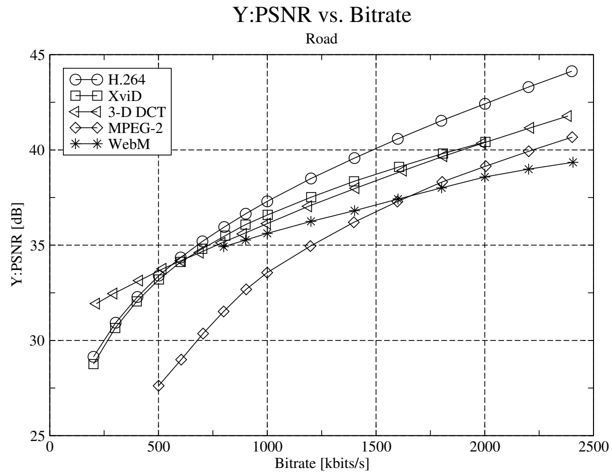
(a) Video sequence Church.



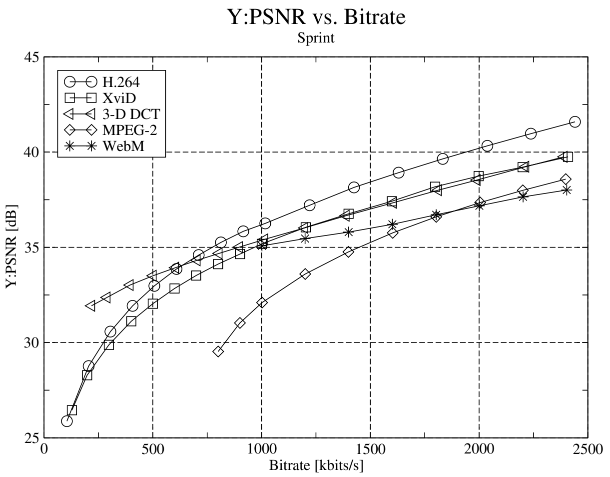
(b) Video sequence High Jump.



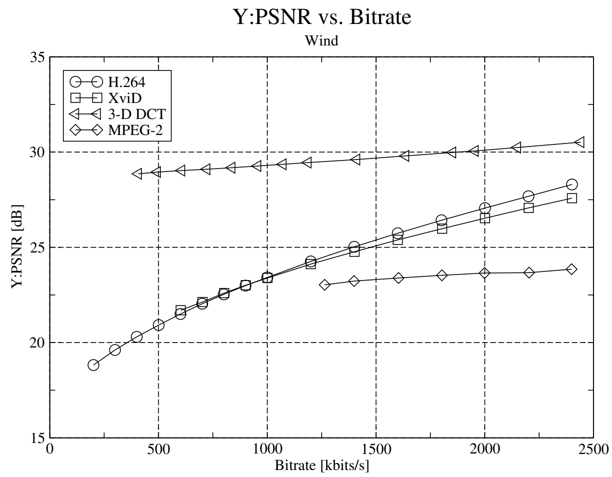
(c) Video sequence Highway.



(d) Video sequence Road.



(e) Video sequence Sprint.



(f) Video sequence Wind.

Fig. 4. Comparison of 3-D DCT with other video coding methods by *PSNR*.

Video Sequence	Bitrate [kbits/s]	Y : $PSNR$ [dB]				
		H.264	XviD	3-D DCT	MPEG-2	WebM
Church	300	43.49	–	44.64	41.94	39.63
	1,000	–	–	50.27	–	41.02
	2,200	–	–	54.06	–	–
High Jump	500	36.23	35.34	35.33	29.99	34.40
	1,000	39.22	38.05	37.58	35.75	36.50
	2,200	43.68	–	41.98	40.75	38.88
Sprint	500	32.98	32.04	33.51	–	–
	1,000	36.22	35.18	35.35	32.10	35.08
	2,200	40.87	39.21	39.20	37.99	37.65
Wind	500	20.92	–	28.97	–	–
	1,000	23.42	23.39	29.06	–	–
	2,200	27.68	27.07	30.30	23.67	–

Tab. 4. Comparison of 3-D DCT with other video coding methods by *PSNR*.

(a) Original sequence. **(b)** H.264. **(c)** 3-D DCT. **(d)** MPEG-2. **(e)** WebM.

Fig. 5. Sample frames of video sequence Church encoded by H.264, 3-D DCT, MPEG-2 and WebM at 300 kbits/s.

(a) Original sequence. **(b)** H.264. **(c)** 3-D DCT. **(d)** XviD. **(e)** WebM.

Fig. 6. Sample frames of video sequence High Jump encoded by H.264, 3-D DCT, XviD and WebM at 500 kbits/s.

(a) Original sequence. **(b)** H.264. **(c)** 3-D DCT. **(d)** XviD.

Fig. 7. Sample frames of video sequence Sprint encoded by H.264, 3-D DCT and XviD at 500 kbits/s.

(a) Original sequence. **(b)** H.264. **(c)** 3-D DCT. **(d)** XviD.

Fig. 8. Sample frames of video sequence Wind encoded by H.264, 3-D DCT and XviD at 500 kbits/s.

of 720×576 sequence could be provided only with 8-point DCT implementation followed by optimization. The coding results of real video sequences were presented in terms of output bitrate and picture quality. The *PSNR* metric was measured for the luminance component and bitrates were adjusted between 200 and 2,400 kbits/s. The different video codes and standards were compared as well. Namely H.264, XviD, MPEG-2 and WebM coders were used and the coding properties are described in the paper.

Future work will attend to detailed comparison of the 3-D DCT with other methods via more metrics. The computing demands of all blocks from encoder and decoder, followed by the implementation on modern DSP or FPGA will be performed as well. The compress properties of the 3-D DCT method could be unambiguously improved by variable video cube size and by addition of image post processing techniques, such as loop filter.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 230126. This work has been supported by the research program no. MSM 0021630513 of the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] ADJEROH, D. A., SAWANT, S. D. Error-resilient transmission for 3D DCT coded video. *IEEE Transactions on Broadcasting*, 2009, vol. 55, no. 2, p. 178–189.

[2] CHAN, Y-L., SIU, W-C. Variable temporal-length 3-D Discrete Cosine Transform coding. *IEEE Transactions on Image Processing*, 1997, vol. 6, no. 5, p. 758–763.

[3] CHASSAING, R. *Digital Signal Processing and Applications with the C6713 and C6416 DSK.* 1st ed. New Jersey: Wiley & Sons, Inc., 2005.

[4] *FFmpeg project.* [Online] Cited 2010-08-13. Available at: http://ffmpeg.org/.

[5] FRYZA, T., HANUS, S. Algorithms for fast computing of the 3D DCT transform. *Radioengineering*, 2003, vol. 12, no. 1, p. 23–26.

[6] FRYZA, T., HANUS, S. Dissimilarity detection in video sequences using variation in time domain. In *Radioelektronika 2004 Conference Proceedings.* Bratislava (Slovakia), 2004, p. 162–165.

[7] GHANBARI, M. *Standard Codecs: Image Compression to Advanced Video Coding.* 1st ed. London: IEE, 2003.

[8] GONZALEZ, R. C., WINTZ, P. *Digital Image Processing.* 2nd ed. Boston: Addison Wesley Publishing Company, 1987.

[9] GRGIC, S., MRAK, M., GRGIC, M. Picture quality measures in image compression systems. In *Conference Proceedings EUROCON 2003.* Ljubljana (Slovenia), 2003, p. 233–237.

[10] HUSKA, J., KULLA, P. Content adaptive true motion estimator for H.264 video compression. *Radioengineering*, 2007, vol. 16, no. 4, p. 68–75.

[11] LAKHANI, G. Optimal Huffman coding of DCT blocks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, vol. 14, no. 4, p. 522–527.

[12] LEE, M. C., CHAN, R. K. W., ADJEROH, D. A. Quantization of 3D DCT coefficients and scan order for video compression. *Journal of Visual Communication and Image Representation*, 1997, vol. 8, no. 4, p. 405–422.

[13] LU, W.-W., GOUGH, M. P. A Fast-adaptive Huffman coding algorithm. *IEEE Transactions on Communications*, 1993, vol. 41, no. 4, p. 535–538.

[14] QURESHI, S. *Embedded Image Processing on the TMS320C6000TM DSP, Examples in Code Composer StudioTM and MATLAB.* 1st ed. New York: Springer Science+Business Media, Inc., 2005.

[15] RAO, K. R., YIP, P. *Discrete Cosine Transform. Algorithms, Advantages, Applications.* 1st ed. San Diego: Academic Press, Inc., 1990.

[16] RICHARDSON, I. E. G. *Video Codec Design. Developing Image and Video Compression Systems.* 1st ed. New Jersey: John Wiley & Sons Ltd., 2002.

[17] TRETTER, S. A. *Communication System Design Using DSP Algorithms: With Laboratory Experiments for the TMS320C6701 and TMS320C6711.* 1st ed. New York: Springer, 2003.

[18] WALLACE, G. K. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1992, vol. 38, no. 1, p. xviii–xxxiv.

[19] WATKINSON, J. *The MPEG Handbook. MPEG-1, MPEG-2, MPEG-4.* 2nd ed. Burlington: Focal Press, 2004.

[20] *WebM project.* [Online] Cited 2010-06-23. Available at: http://www.webmproject.org/.

[21] WESTWATER, R., FURHT, B. Three-dimensional DCT video compression techniques based on adaptive quantizers. In *Second IEEE International Conference on Engineering of Complex Computer Systems.* Montreal (Canada), 1996, p. 189–198.

[22] WESTWATER, R., FURHT, B. *Real-Time Video Compression. Techniques and Algorithms.* 1st ed. Boston: Kluwer Academic Publishers, 1997.

About Author...

Tomas FRYZA was born in 1977 in Novy Jicin, Czech Republic. He received his M.Sc. and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering and Communication, Brno University of Technology in 2002 and in 2006, respectively. At present he is an assistant professor at the Department of Radio Electronics, Brno University of Technology. His research interests include digital and microprocessor technique, video signal compression and C/C++ language programming. He is IEEE member since 2003.