Algorithms for Fast Computing of the 3D-DCT Transform

Tomáš FRÝZA, Stanislav HANUS

Dept. of Radio Electronics, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic

fryza@feec.vutbr.cz, hanus@feec.vutbr.cz

Abstract. *The algorithm for video compression based on the Three-Dimensional Discrete Cosine Transform (3D-DCT) is presented. The original algorithm of the 3D-DCT has high time complexity. We propose several enhancements to the original algorithm and make the calculation of the DCT algorithm feasible for future real-time video compression.*

Keywords

Video compression, 3D-DCT, complexity, DFT, multi-dimensional transforms, optimization.

1. Introduction

The Three-Dimensional Discrete Cosine Transform (3D-DCT) is used for video compression. The main principle is the correlation not only between points inside one frame but also between successive frames.

The input of the 3D-DCT encoder is so-called video cube, which is a cube (dim. $n \times n \times n$) made up of small elements of particular frames with the dimensions $n \times n$.

The 3D-DCT encoder and decoder are symmetrical and each has only three parts, as shown in Fig. 1. They are the forward or inverse 3D-DCT calculations, the quantization or dequantization and, finally, the variable- length coding or decoding. The main objective of this article is to describe the methods of fast computing of the Three-Dimensional Discrete Cosine Transform.

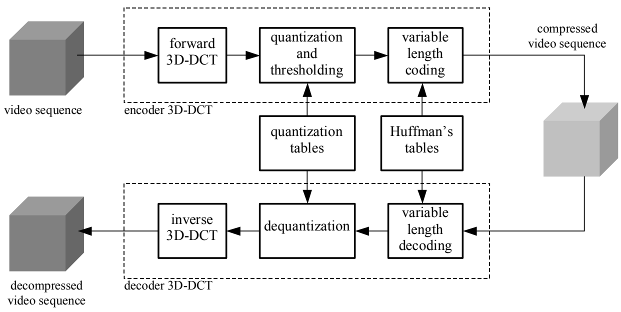

Fig. 1. Block diagram of 3D-DCT encoder and decoder.

Let n=8. Then, by Westwater [1] the Forward Three-Dimensional Discrete Cosine Transform (3D-FDCT) is defined in the following way:

$$F_{u,v,w} = \frac{1}{8} \cdot C_u C_v C_w \cdot \sum_{x=0}^{7}\sum_{y=0}^{7}\sum_{z=0}^{7} s_{x,y,z} \cdot D(x,u) \cdot D(y,v) \cdot D(z,w), \tag{1}$$

where $s_{x,y,z}$ is the intensity of picture element (pixel), $F_{u,v,w}$ is the 3D-DCT coefficient and $C_u=C_v=C_w=0.7071$ for $u=v=w=0$ while $C_u=C_v=C_w=1$ for all $u>0$, $v>0$ and $w>0$. We construct a DCT transform base as follows:

$$D(a,b) = \cos\frac{(2a+1)b\pi}{16}, \tag{2}$$

where parameters $a \in \{x, y, z\}$ and $b \in \{u, v, w\}$.

In a similar way we define the Inverse Three-Dimensional Discrete Cosine Transform (3D-IDCT) of the form:

$$s_{x,y,z} = \frac{1}{8} \cdot \sum_{u=0}^{7}\sum_{v=0}^{7}\sum_{w=0}^{7} C_u C_v C_w \cdot F_{u,v,w} \cdot D(x,u) \cdot D(y,v) \cdot D(z,w). \tag{3}$$

2. Methods of Calculation

Using the 3D-DCT transform in video compression has a serious disadvantage in that the calculation of coefficients by equations (1) and (3) is very demanding in terms of time. The number of necessary operations for 512 3D-DCT coefficients (i.e. one video cube) is 262,144 products and 175,616 sums. The disadvantage mentioned above can be avoided by dividing the three-dimensional transform into a succession of a two-dimensional and a one-dimensional transform or three one-dimensional transforms.

In section 2.1, the application of scalar product in order to simplify the calculation is described. In section 2.2, the Discrete Fourier Transform (DFT) is used. Finally, all methods are discussed.

2.1 Application of Scalar Product

Here, we divide the 3D-DCT transform into a succession of a 2D-DCT followed by a 1D-DCT. The equation for the Two-Dimensional Discrete Cosine Transform (2D-DCT) is defined as follows:

$$\mathbf{G}_w = \mathbf{H}_{DCT} \cdot \mathbf{S}_w \cdot \mathbf{H}_{DCT}^T , \tag{4}$$

where $\mathbf{G}_w$ is the two-dimensional coefficients matrix (8×8), $\mathbf{S}_w$ is the matrix of picture elements, w=0…7, and $\mathbf{H}_{\mathrm{DCT}}$ is the base of the 1D-DCT Transform.

The remaining One-Dimensional Discrete Cosine Transform can be written in the following form:

$$\mathbf{L}_u = \mathbf{H}_{DCT} \cdot \mathbf{I}_u^T , \tag{5}$$

where $\mathbf{L}_u$ is the matrix of three-dimensional coefficients (8×8), $\mathbf{I}_u$ is the semi-transformed matrix of two-dimensional coefficients $\mathbf{G}_w$, and u=0…7. The number of fundamental mathematical operations is reduced to 12.288 products and 10.752 sums for one video cube 8×8×8, i.e. for 512 coefficients.

2.2 Application of Discrete Fourier Transform

In this section the application of the 1D, 2D and 3D Discrete Fourier Transforms to 3D-DCT calculation is presented. We use the definition of 1D Discrete Fourier Transform and we derive the final form (6). Because of the length of the derivation we omit it here.

$$\frac{F'_u}{E_u} = \sum_{x=0}^{7} s_x \cdot D(x,u), \tag{6}$$

where the right side of equation is equal to 1D-DCT, variable u=0...7, constant E_u is given by:

$$E_u = 2 \cdot \cos\frac{u\pi}{16} \tag{7}$$

and the real part of DFT F_u' has the form:

$$F'_u = \sum_{x=0}^{7} s_x \cdot [\cos\frac{2\pi ux}{16} + \cos\frac{2\pi u(x+1)}{16}]. \tag{8}$$

As we can see, the 1D-DCT Transform could be calculated using only the real part of the Discrete Fourier Transform divided by the real constant. The numerator of the left side of equation (8) can be modified and rewritten into the following equation system.

$$\begin{aligned}
F'_0 &= 2(s_0 + s_1 + s_2 + s_3 + s_4 + s_5 + s_6 + s_7) \\
F'_1 &= s_0 - s_7 + B_1C_1 + B_2C_2 - B_3(C_2 - C_3) \\
F'_2 &= s_0 - s_3 - s_4 + s_7 + B_1 \cdot C_4 \\
F'_3 &= s_0 - s_7 - B_1C_1 - B_4C_3 - B_3(C_2 - C_3) \\
F'_4 &= s_0 - s_1 - s_2 + s_3 + s_4 - s_5 - s_6 + s_7 \\
F'_5 &= s_0 - s_7 - B_1C_1 + B_4C_3 + B_3(C_2 - C_3) \\
F'_6 &= s_0 - s_3 - s_4 + s_7 - B_1 \cdot C_4 \\
F'_7 &= s_0 - s_7 + B_1C_1 - B_2C_2 + B_3(C_2 - C_3),
\end{aligned} \tag{9}$$

where constants B_i and C_i are as described below

$$\begin{aligned}
B_1 &= 0{,}7071 \\
B_2 &= 0{,}9239 + 0{,}3827 \\
B_3 &= 0{,}9239 \\
B_4 &= 0{,}9239 - 0{,}3827
\end{aligned} \tag{10}$$

$$\begin{aligned}
C_1 &= s_1 + s_2 - s_5 - s_6 \\
C_2 &= s_2 + s_3 - s_4 - s_5 \\
C_3 &= s_0 + s_1 - s_6 - s_7 \\
C_4 &= s_0 + s_1 - s_2 - s_3 - s_4 - s_5 + s_6 + s_7 .
\end{aligned} \tag{11}$$

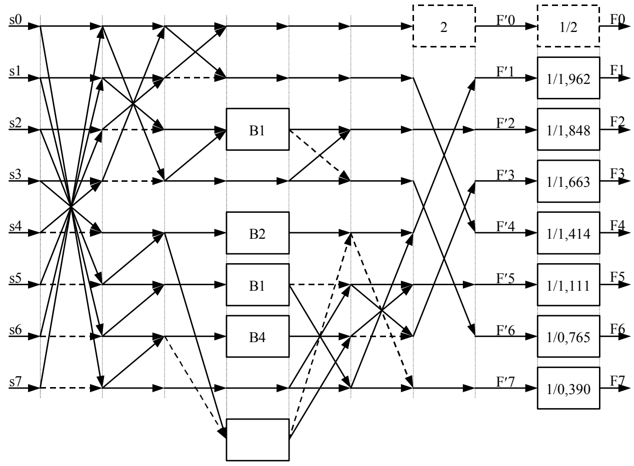

Fig. 2. Flow diagram of calculation of the Forward 1D-DCT based on the Discrete Fourier Transform.

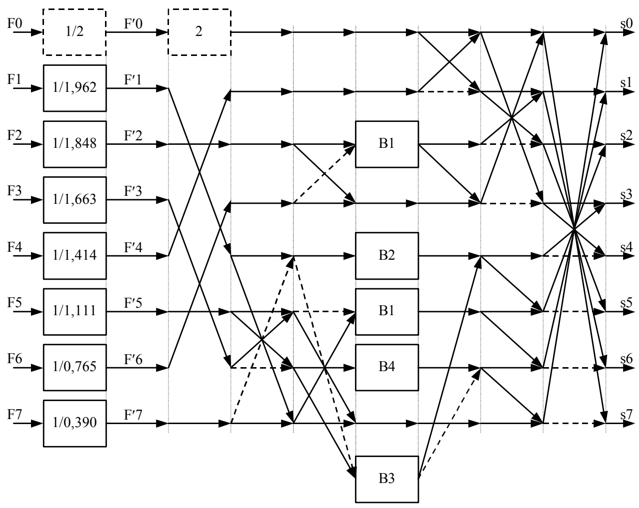

Fig. 3. Flow diagram of calculation of the Inverse 1D-DCT based on the Discrete Fourier Transform.

The situation is shown in Fig. 2, where the solid line represents the sum and the dashed line the difference of two numbers. We can see that the number of products for the calculation of eight one-dimensional DFT coefficients is only 5. We also need 29 sums.

Dividing the real part of the Discrete Fourier Transform F'_u by the real constant can be implied in the second section of the 3D-DCT encoder, which is the quantization section.

Similarly we can derive the equation system for the Inverse 1D-DCT Transform as shown Fig. 3. The number of necessary arithmetic operations is equal to that for the Forward 1D-DCT Transform described above.

If we express the 3D-DCT Transform by the above procedure, we obtain 960 products (3.64.5) and 5.568 (3.64.29) sums.

Analogous to the one-dimensional transform, we can express the 2D-DCT and 3D-DCT Transforms in the following forms:

$$\frac{F'_{u,v}}{E_{u,v}} = \sum_{x=0}^{7}\sum_{y=0}^{7} s_{x,y} \cdot D(x,u)D(y,v), \tag{12}$$

$$\frac{F'_{u,v,w}}{E_{u,v,w}} = \sum_{x=0}^{7}\sum_{y=0}^{7}\sum_{z=0}^{7} s_{x,y,z} \cdot D(x,u)D(y,v)D(z,w), \tag{13}$$

where the right sides of equations are equals to 2D-DCT and 3D-DCT, respectively, while variables u,v,w=0...7, constants $E_{u,v}$ and $E_{u,v,w}$ are given by:

$$E_{u,v} = 4 \cdot \cos\frac{(u+v)\pi}{16}, \tag{14}$$

$$E_{u,v,w} = 32 \cdot \cos\frac{(u+v+w)\pi}{16} \tag{15}$$

and the real parts of the Two-Dimensional and Three-Dimensional Discrete Fourier Transforms $F'_{u,v}$ and $F'_{u,v,w}$ are given by formulas (16) and (17).

$$\begin{aligned} F'_{u,v} = \sum_{x=0}^{7}\sum_{y=0}^{7} s_{x,y} \cdot \\ [\cos\frac{(ux+vy)2\pi}{16} \\ + \cos\frac{(ux-v(y+1))2\pi}{16} \\ + \cos\frac{(-u(x+1)+vy)2\pi}{16} \\ + \cos\frac{(-u(x+1)-v(y+1))2\pi}{16}]. \end{aligned} \tag{16}$$

$$\begin{aligned} F'_{u,v,w} = \sum_{x=0}^{7}\sum_{y=0}^{7}\sum_{z=0}^{7} s_{x,y,z} \cdot \\ [\cos\frac{(ux+vy+wz)2\pi}{16} \\ + \cos\frac{(ux+vy-w(z+1))2\pi}{16} \\ + \cos\frac{(ux-v(y+1)+wz)2\pi}{16} \\ + \cos\frac{(ux-v(y+1)-w(z+1))2\pi}{16} \\ + \cos\frac{(-u(x+1)+vy+wz)2\pi}{16} \\ + \cos\frac{(-u(x+1)+vy-w(z+1))2\pi}{16} \\ + \cos\frac{(-u(x+1)-v(y+1)+wz)2\pi}{16} \\ + \cos\frac{(-u(x+1)-v(y+1)-w(z+1))2\pi}{16}]. \end{aligned} \tag{17}$$

The solutions of formulas (16) and (17) are equation systems with 64 and 512 equations, respectively. Using equations for the 2D and 3D Discrete Fourier Transforms we further reduce the number of arithmetic operations (the number of products in particular). Here from (16), we require 840 products and c. 7,500 sums for 512 coefficients. For the 3D DFT we expect a partial decrease in products and a considerable increase in source code complexity.

3. Simulation Results

A comparison of all methods (except the 3D Discrete Fourier Transform) is given in Tab. 1.

Parameter	Definition	Use of Scalar Product	Use of 1D DFT	Use of 2D DFT
Number of products	262,144	12,288	960	840
Number of sums	175,616	10,752	5,568	7,500
Calculation time [s]	-	15.5	15.9	48.1
Complexity	2	2	5	10

Tab. 1. Comparison of computing methods.

As the testing procedure we use a video sequence with a frame size of 640x480 pixels and the duration of 8 frames. All executions have been done in Matlab on a PC with CPU PIII/733 MHz. We observed four parameters: number of products and sums, calculation time, and source code complexity (as a subjective opinion).

According to Tab. 1, the Scalar algorithm needs 4.7 % of products and 6.1 % of sums comparing to the definition of the 3D-DCT. Also, the 1D DFT algorithm needs only 0.4 % of products and 3.2 % of sums! Calculation times have been measured in Matlab and their values correspond to a way of programming. Because of this reason, the less-operations methods could be more demanding in terms of time.

The choice of the method is given by the required application. For the implementation of the 3D-DCT algorithm in Matlab we will probably choose the method based on the scalar product, in view of the low complexity. On the other hand, in the evaluation of the 3D-DCT Transform on a DSP (digital signal processor) our objective will be to have a small number of products and sums. Consequently, we will use the application of the 1D (or 2D) Discrete Fourier Transform.

4. Conclusion

We examined the Three-Dimensional Discrete Cosine Transform (3D-DCT) for the compression of video signals. First we introduced an encoder and decoder based on the 3D-DCT Transform. Subsequently we defined equations of the Forward and the Inverse 3D-DCT Transforms.

The main objective in this article has been to simplify the calculation of the 3D-DCT, i.e. to decrease the number of operations necessary for a computation of 3D-DCT coefficients and execution time. We have proposed three alternatives: using the scalar products, using the One-Dimensional Discrete Fourier Transform and, finally, the application of the multi-dimensional DFT (named Two and Three Dimensional Discrete Fourier Transforms).

References

[1] WESTWATER, R., FURHT, B. *Real-Time Video Compression. Techniques and Algorithms*. Boston: Kluwer Academic Publishers, 1997. ISBN 0-7923-9787-8.

[2] CHROMÝ, I. *Compression of Digital Video Signals*. Postgraduate thesis. Brno: Department of Radio Electronics FEI VUT, 1999.

[3] RILEY, M.J., RICHARDSON, I.E.G. *Digital Video Communications*. Boston – London: Artech House, 1997. ISBN 0-89006-890-9.

[4] FRÝZA, T. Three-Dimensional Discrete Cosine Transform 3D-DCT and methods of calculation. In *Proceedings of 8th conference Student EEICT 2002*, vol. I., p. 57-59.

[5] FRÝZA, T. *Compression of Video Signals by 3D-DCT Transform*. Diploma Thesis. Brno: Department of Radio Electronics FEKT VUT, 2002.

[6] BURG, A., KELLER, R. A Real-Time Video Compression System. In *Proceedings of MoMuC2000*, Tokyo, 2000.

About Authors...

Tomáš FRÝZA was born in 1977 in Nový Jičín (Czech Republic). He received the Ing. (M.Sc.) degree in electrical engineering from the Faculty of Electrical Engineering and Communication, Brno University of Technology (FEKT VUT) in 2002. At present he is a Ph.D. student at the Institute of Radio Electronics, FEEC VUT in Brno.

Stanislav HANUS was born in Brno, Czechoslovakia, in 1950. He received the Ing. (M.Sc.) and CSc. (Ph.D.) degrees from the Brno University of Technology. He is Associate Professor at the Institute of Radio Electronics, Faculty of Electrical Engineering and Communication in Brno. His research is concentrated on Mobile Communications, Television Technology and Circuit Theory.

Call for Papers

Utilization of MATLAB in Radio Engineering Computations

A Special Issue of

Radioengineering

October 15, 2003 Submission of a paper

A paper is requested being submitted via e-mail to **raida@feec.vutbr.cz**. The manuscript has to be prepared in MS Word for Windows in Radioengineering Publishing Style. The example document with proper styles is available at:

http://www.feec.vutbr.cz/UREL/RADIOENG/

Due to the very short publication time, extreme care has to be devoted to the formal processing of the paper. A paper of an inadequate form might be rejected from that reason. An author is encouraged to submit a PDF version of the paper together with the word document in order to simplify handling the paper.

October 31, 2003 Notification of acceptance

Two independent reviewers review a paper submitted. Reviews are distributed to the authors via e-mail. If paper is accepted for publication, authors are asked to revise the paper in accordance with review comments.

November 15, 2003 Submission of a revised paper

The revised paper has to be delivered to the editors via e-mail in the form of the MS Word document.