DOI: 10.21917/ijivp.2012.0075

DETERMINING OPTIMAL CUBE FOR 3D-DCT BASED VIDEO COMPRESSION FOR DIFFERENT MOTION LEVELS

J. Augustin Jacob[1] and N. Senthil Kumar[2]

[1]*Department of Electronics and Communication Engineering, Kamaraj College of Engineering and Technology, India*
E-mail: mailtoaugus@yahoo.co.in
[2]*Department of Electrical and Electronics Engineering, Mepco Schlenk Engineering College, India*
E-mail: nsk_vnr@yahoo.com

Abstract

This paper proposes new three dimensional discrete cosine transform (3D-DCT) based video compression algorithm that will select the optimal cube size based on the motion content of the video sequence. It is determined by finding normalized pixel difference (NPD) values, and by categorizing the cubes as "low" or "high" motion cube suitable cube size of dimension either [16×16×8] or[8×8×8] is chosen instead of fixed cube algorithm. To evaluate the performance of the proposed algorithm test sequence with different motion levels are chosen. By doing rate vs. distortion analysis the level of compression that can be achieved and the quality of reconstructed video sequence are determined and compared against fixed cube size algorithm. Peak signal to noise ratio (PSNR) is taken to measure the video quality. Experimental result shows that varying the cube size with reference to the motion content of video frames gives better performance in terms of compression ratio and video quality.

Keywords:

3D-DCT, PSNR, Video Compression

1. INTRODUCTION

Many video compression algorithms are based on reducing the spatial and temporal redundancy by prediction and motion compensation. However these algorithms are very hard to implement in hardware and no symmetry exists between encoding and decoding block. 2D-Discrete Fourier transform was used frequently in video compression algorithm. The reason is DCT has energy compaction property [1]. A video sequence can be viewed as three dimensional (3D) signals. If we apply 2D-DCT concentration of energy is achieved in spatial domain, by extending this DCT to the third dimension (temporal domain) similar concentration of energy is achieved. The 3D-DCT algorithm is stated in [2],[3],[4]. They constructed fixed cube size represented as, (($N_x \times N_y \times N_z$) where N_x and N_y represent the spatial length and N_z represent the temporal length. Usually the cube size will be [8×8×8]. Many fast algorithms are available for implementing the 3D-DCT algorithm stated in [5] also with the cheaper availability of memory, there is a possibility of replacing the standard video compression algorithm like MPEG (Motion Picture Expert Group) with the 3D-DCT algorithm. Motion detection algorithm was proposed by Chan and wan [6]. In that they considered the variable temporal cube. But they have not discussed about memory requirements if the cube is temporally variable.

An adaptive 3D-DCT algorithm was proposed by Borko and Kan [7]. In that blocking artifact is seen on the reconstructed Video sequence and also cube construction is uneven. We propose a new 3D-DCT algorithm that preprocesses the video sequence using motion estimation algorithm as shown in Fig.1. Based on level of motion in video frame dynamically cube size is chosen for encoding. The effectiveness is verified by performing rate vs. distortion analysis. The Block diagram of the proposed 3D-DCT encoding algorithm is shown in Fig.1. All the constructed cubes are processed sequentially through all the blocks as mentioned in Fig.1 to complete encoding. The same process is reversed to get the original video sequence.

2. THREE DIMENSIONAL DISCRETE FOURIER TRANSFORM ALGORITHM

Three dimensional Discrete Fourier transform is an extension of 2D-DCT (i.e.) by taking one more one dimensional DCT along the temporal domain will give the 3D-DCT. Mostly for video compression algorithm DCT is chosen because of its energy compaction property. The forward and inverse three dimensional DCT is given by,

$$F(X,Y,Z)=\sqrt{\frac{8}{N_x.N_y.N_z}}\, f_x f_y f_z \sum_{x=0}^{N_x}\sum_{y=0}^{N_y}\sum_{z=0}^{N_z} f(x,y,z) \frac{Cos(2x+1)X\pi}{2N_x}.\frac{Cos(2y+1)Y\pi}{2N_y}.\frac{Cos(2z+1)Z\pi}{2N_z} \tag{1}$$

where,

$$f_x f_y f_z = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } x,y,z=0 \\ 1 & \text{otherwise} \end{cases}$$

$$f(x,y,z)=\sqrt{\frac{8}{N_x.N_y.N_z}}\, f_x f_y f_z \sum_{x=0}^{N_x}\sum_{y=0}^{N_y}\sum_{z=0}^{N_z} F(X,Y,Z) \frac{Cos(2x+1)X\pi}{2N_x}.\frac{Cos(2y+1)Y\pi}{2N_y}.\frac{Cos(2z+1)Z\pi}{2N_z} \tag{2}$$

Many fast algorithms exist for finding 3D-DCT efficiently [5] and [8] that will accelerate the process of encoding.

3. MOTION ESTIMATION ALGORITHM

To estimate the motion level of the video sequence the input sequence is divided into cube of dimension [16×16×8] and then normalized pixel difference (NPD) between 1st and 8th frame is determined. Based on the NPD values the cube is categorized into "low" motion cubes and "high" motion cubes.

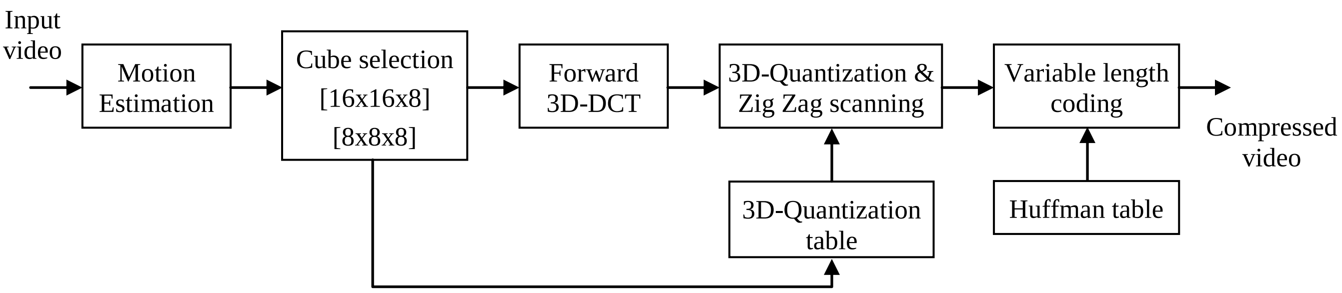

Fig.1. Block diagram of 3D-DCT base encoder

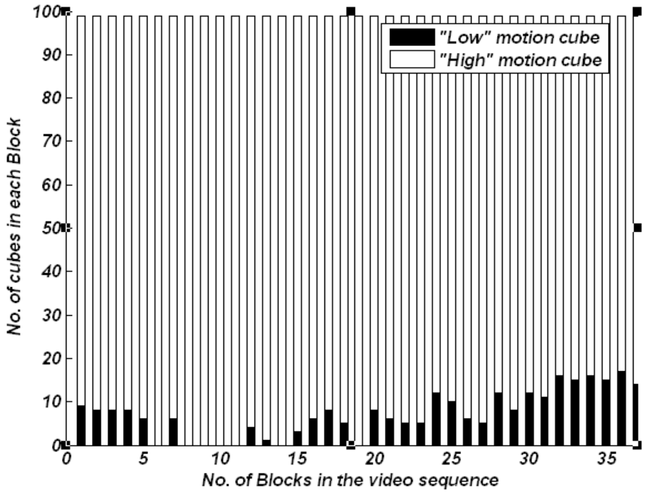

Fig.2. Distribution of motion level for coast guard sequence

Three test sequences having different motion level is taken for analysis. Each sequence is of size 176 × 144 and of type [4:2:0] YUV.

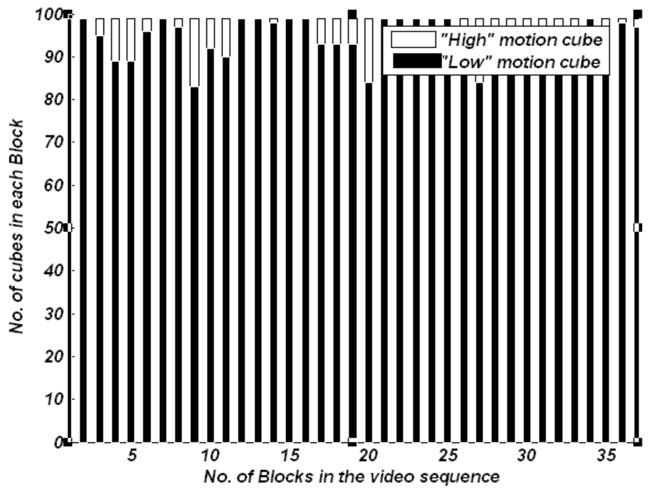

Fig.3. Distribution of motion level for akiyo sequence

For the above mentioned test sequence for a single block of frames there will be 99 cubes. According to the NPD values each cube is grouped under "low" motion cubes and "High" motion cubes. Fig.2 and Fig.3 shows the NPD values for the two test sequences. The distribution of motion level of the three test sequence is shown in Table.1. If we consider the entire frame length for "akiyo" sequence, out of 3663 cubes, 3491 cubes fall under "low" motion category and 172 cubes fall under "high" motion category. It can be clearly seen that majority of cubes in each block for "akiyo" sequence fall under the "low" motion category. So for encoding that block cube of dimension [16×16×8] is chosen. Because "low" motion cubes have highly correlated neighboring pixels values. If we take DCT for that block the entire cube can be represented with few values because of the energy compaction property. Similarly for "coastguard" sequence, out of 3663 cubes 265 cubes fall under "low" motion category and 3398 cubes fall under "high" motion category. In case of "coastguard" sequence majority of blocks fall under the "high" motion category so the chosen cube size is [8×8×8] it is shown in Fig.2.

Table.1. Distribution of motion level of three test sequence

Sl. No.	**Name of the test sequence**	**Number of cubes in each category**	
		"Low" motion category	**"High" motion category**
1	akiyo	3491	172
2	coastguard	265	3398
3	news	2205	1458

4. QUANTIZATION AND ZIG ZAG ORDERING

In the field of Video compression quantization plays an important role. For efficient encoding proper selection of values for the quantization table are needed. We cannot rely on the two dimensional quantization table that is designed for two dimensional discrete cosine transform. Hence generation of three dimensional quantization is essential. In that the entries are chosen by collecting the dynamic range of DC coefficients that are ranges from 2500 to 8000 and AC coefficients that are ranges from -1000 to 1000. It is analyzed in [9] and [10] that significant numbers of coefficients are concentrated on the major axes. It is stated by Eq.(3),

$$x + y + z \leq k \tag{3}$$

where, $k = 3,4 \ldots\ldots\ldots.. (x + y + z)$.

$F(1,1,1)$ is the DC coefficient and remaining values are AC coefficients. If the transformed value satisfies the condition given in Eq.(3) they are classified as significant coefficients. Detailed analysis was made as stated in [11], [12] and concluded with the values ranges from 8 to 16 for significant coefficients and 75 to 110 for the remaining coefficients. For Zig Zag ordering the same Eq.(3) is used and data are rearranged into one dimensional array that can be efficiently coded using Huffman coding algorithm.

5. EXPERIMENTAL RESULTS

The performance of the proposed algorithm is verified by finding the compression ratio and quality of reconstructed video sequence of the test sequence and the results are compared against the fixed cube size 3D-DCT based compression algorithm. The following experimental setup was considered for doing the rate vs. distortion analysis. Three test sequences with different motion levels are taken. Each test sequence has 296 frames of dimension 176 × 144 with frame rate of 25 frames/second. The quality of the reconstructed video sequence is measured by taking peak signal to Noise ratio (PSNR) as defined in Eq.(4) and Eq.(5),

$$\text{PSNR} = 10\log 10\left[\frac{255^2}{\text{Mean Squared error (MSE)}}\right] \quad (4)$$

$$\text{MSE} = \frac{1}{N_x N_y N_d}\sum_{d=0}^{N_d}\sum_{x=0}^{N_x}\sum_{y=0}^{N_y}[f(x,y,d)-\bar{f}(x,y,d)]^2 \quad (5)$$

where, $f(x, y, d)$ represent the original frame and $\bar{f}(x,y,d)$ represent the reconstructed frame and N_x, N_y represent the frame size, N_d represent the number of frames in the sequence. For rate vs. distortion analysis only the luminance component is taken. So in the graph PSNR is represented as Y-PSNR. Majority of blocks in the "akiyo" sequence are encoded with the cube size [16×16×8]. Because block wise majority of cubes are "low" motion cubes. Hence greater compression can be achieved. It is clearly seen from Fig.4 that proposed algorithm perform better than the fixed cube size algorithm. There is a noticeable improvement in the PSNR value ranges from 1 dB to 2 dB against the compression ratio 60:1. In case of "coastguard" sequence more number of block is encoded with the cube size [8×8×8] because it has more number of "high" motion cubes in each block. So there is no change in the PSNR value. In case of "News" sequence shown in Fig.5 even though majority of blocks are encoded with [16×16×8] cube. More number of coefficients is required to get the desired PSNR. For all the test sequence the proposed algorithm outperform the fixed cube size 3D-DCT algorithm and we get a compression ratio ranges between 15:1 to 60:1 without much degradation in the video quality.

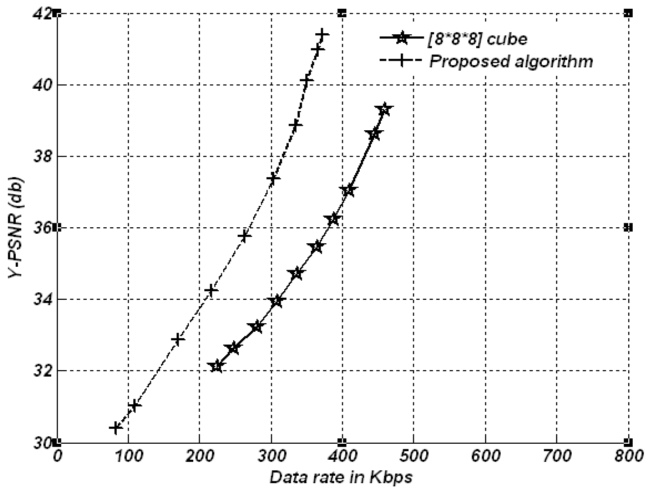

Fig.4. Rate vs. distortion plot for "akiyo" sequence

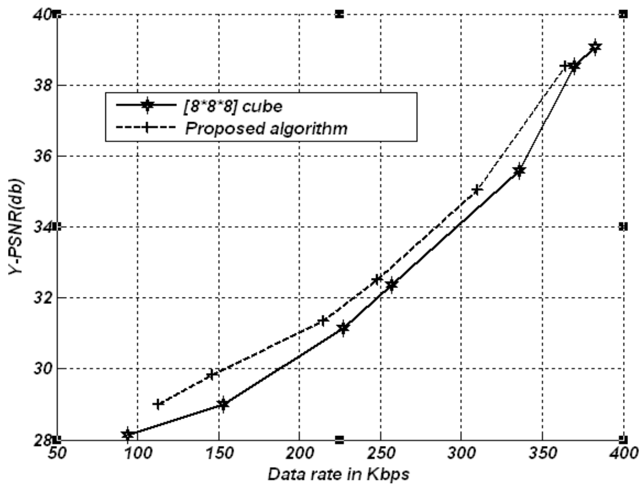

Fig.5. Rate vs. distortion plot for "News" sequence

6. CONCLUSION AND FUTURE WORK

We presented a new algorithm that will determine the cube size for 3D-DCT based video compression technique by analyzing the motion content of the video sequence. By finding the normalized pixel differences values and categorize as "low" and "High" motion suitable cube size is chosen. Experimental results reveal that we can achieve better compression with minimum distortion by selecting variable cube size algorithm instead of using standard [8×8×8] cube. We used the standard variable length coding method that is stated for 2D-DCT based video compression technique. Future work could be reducing the data rate further by designing an optimized coding technique suitable for 3D-DCT based video compression algorithm.

REFERENCES

[1] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform", *IEEE Transaction on Computers*, Vol. C-23, No. 1, pp. 90–93, 1974.

[2] R. Westwater and B. Furht, "Three dimensional DCT Video compression technique based on adaptive

quantizers", *Proceedings of the Second International Conference on Engineering of Complex Computer Systems*, pp. 189-198, 1996.

[3] M. Servais and G. de Jager, "Video compression using the Three-Dimensional Discrete cosine Transform (3D-DCT)", *Proceedings of the 1997 South African Symposium on Communications and Signal Processing*, pp. 27-32, 1997.

[4] Li Li and Zhengxin Hou, "Multi view Video compression with 3D-DCT", *ITI 5th International Conference on Information and Communications Technology*, pp. 59-61, 2007.

[5] S. Boussakta and H.O. Alshibami, "Fast Algorithm for the 3D-DCT-II", *IEEE Transactions on Signal Processing*, Vol. 52, No. 4, pp. 992-1001, 2004.

[6] 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.

[7] B. Furht, K. Gustafson, H. Huang and O. Marques, "An Adaptive Three – Dimensional DCT Compression Based on Motion Analysis", Proceedings *of the ACM Symposium on Applied Computing*, pp. 765-768, 2003.

[8] M.C. Lee, K.W. Chan and D.A. Adjeroh, "Quantization of 3D-DCT coefficients and scan order for video compression", *Journal of Visual Communication and Image Representation*, Vol. 8, No. 4, pp. 405 – 422, 1997.

[9] R.K.W. Chan and M.C. Lee, "3D – DCT quantization as a compression technique for video sequences", *Proceedings of International Conference on Virtual Systems and Multimedia*, pp. 188–196, 1997.

[10] Boon-Lock Yeo and Bede Liu, "Volume rendering of DCT-based compressed 3D scalar data", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 1, pp. 29-43, 1995.

[11] M. Bhaskaranand and J.D. Gibson, "Distribution of 3D-DCT co-efficients for video", *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 793-796, 2009.

[12] O. Alshibami, S. Boursakta and M. Darnell, "Three-dimensional algorithm for 3-D DCT-III", *Proceedings of Sixth International Symposium on Communication, Theory and Applications*, pp. 104–107, 2001.