Augustin Jacob JACOB[1], Senthilkumar NATARAJAN[2]

Kamaraj College of Engineering and Technology (1), MEPCO Schlenk Engineering College (2)

An approach to adaptive 3D-DCT based motion level prediction algorithm for improved 3D-DCT video coding

Abstract. *This paper reports an adaptive three dimensional discrete cosine transform (3D-DCT) based motion level prediction algorithm which determines the optimal cube for 3D-DCT based compression technique by analyzing the motion content of the video sequence. Irrespective of the motion levels in the video sequence the generally used cube size is [8 × 8 × 8], but the proposed algorithm reported in this paper will adaptively choose the cube size in relation to the motion level of video sequence. The effectiveness of the algorithm can be verified by performing Rate Vs distortion comparison with different motion level sequences. Peak Signal to Noise Ratio (PSNR) has been taken as a measure of distortion. Experimental results reveals that without any motion compensation technique ,the proposed 3D-DCT algorithm which adaptively selects the cube size relative to the motion content of video sequence gives better performance in terms of reduction in the data rate and speed up the encoding process compared to the existing 3D-DCT based video compression algorithm.*

Streszczenie. *W artykule opisano algorytm kompresji video bazujący na adaptacyjnej trójwymiarowej dyskretnej transformacie kosinusowej i przewidywaniu poziomu ruchu. W sposób adaptacyjny dobierany jest rozmiar sześcianu. Weryfikacji dokonano przez porównanie szybkości względem zniekształceń dla różnych sekwencji ruchu.* ***Algorytm poprawy kodowania 3D DCT video bazujący na przewidywaniu poziomu ruchu****.*

Keywords: Video signal processing, 3D-Discrete cosine transforms, Peak signal to noise ratio, Video compression.
Słowa kluczowe: kodowanie video, dyskretna transformata kosinusowa, kompresja video

doi:10.12915/pe.2014.12.22

1. Introduction

Most video compression algorithms rely on reducing the spatial and temporal redundancy by motion compensation and prediction [1,2,3]. However these algorithms have complex motion estimation and prediction algorithms and no symmetry exists between encoding and decoding block. This made implementing this algorithm in hardware is more complex. Frequently used video compression algorithms use 2D-Discrete cosine transform [4, 5] in order to achieve compression, because it possesses the decorrelation and energy compaction property. A video sequence often can be viewed as three dimensional signals. By applying 2D-DCT concentration of energy in spatial domain can be achieved and many fast algorithms are available to implement 2D-DCT [6]. Taking one dimensional DCT along the temporal domain will give the similar concentration of energy in the temporal domain. 3D-DCT is an alternate approach for traditional video compression techniques. Several research efforts were made related to 3D-DCT algorithm in [7,8,9,10] stating the compression ratio and its complexity over other traditional techniques. In that they constructed standard cube size as [$N_r \times N_c \times N_d$], N_r and N_c represent the spatial length and N_d represents the temporal length, $N_r = N_c = N_d = 8$ irrespective of the motion content in the video sequence. Chan and Wan proposed a motion detection algorithm using the variable temporal length cube [11]. Also no detailed study has been made in finding the maximum allowable temporal length but it is required for determining the memory space to store the video frames for processing. Borko and Ken proposed an adaptive 3D-DCT compression algorithm [12], where cube construction is uneven for a single block (8 frames are grouped to form a block) that made further processing through the coder more complex. The proposed algorithms adaptively determine the motion level of the cube by using 3D-DCT energy based approach and encode it with optimal cube size. This will reduce the processing complexity and improve the video quality when compared to standard compression algorithms like MPEG-2 (Motion Picture Experts Group) and the existing fixed cube size 3D-DCT based compression algorithms. Experimental results using different motion level video sequences are encouraging.

2. Motion level prediction and classification

The block diagram of proposed 3D-DCT encoding algorithm is shown in Fig. 1. First the level of motion in a video sequence has to be determined. It is done by pre-processing the video sequence using motion level detection algorithm. The video frames are grouped into blocks (1 block=8 frames). Each block is then partitioned into many cubes of size [16 × 16 × 8] in the raster scan order [$N_r \times N_c \times N_d$], (i.e.) for a video block having the dimension 176 × 144 × 8, it will have 99 cubes. To determine the motion content of the cube the existing algorithm [12] uses normalized pixel difference (NPD) value that is taken between the 1st and 8th frame. It is stated in expression (1). The drawback of this algorithm is for NPD calculations only the 1st and 8th frames are taken, if any scene change occurs in between 2 to 7th frames it is left unnoticed by the existing algorithm. We propose a new energy based algorithm for determining the motion level of the cube. We observed that only few coefficients are required to reconstruct the 99% of cube energy based on that cubes are classified as low, medium and high motion cubes.

$$\frac{1}{N_r \, x \, N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \left| f(r,c)_1 - f(r,c)_8 \right| \tag{1}$$

3. 3D-DCT algorithm

Once the cube size is selected then each cube was processed through all the blocks starting from taking 3D-DCT followed by quantization, zigzag scanning and variable length coding to get the compressed video. The same process is reversed to get the original sequence. 3D-DCT can be obtained by taking one dimensional DCT along the three dimensions [$N_r \times N_c \times N_d$].

The forward three dimensional discrete cosine transform is defined by

$$F(R,C,D) = \sqrt{\frac{8}{N_r.N_c.N_d}} f_r f_c f_d \sum_{r=0}^{N_r} \sum_{c=0}^{N_c} \sum_{d=0}^{N_d} f(r,c,d). \tag{2}$$

$$\frac{Cos(2r+1)R\pi}{2N_r} \cdot \frac{Cos(2c+1)C\pi}{2N_c} \cdot \frac{Cos(2d+1)D\pi}{2N_d}$$

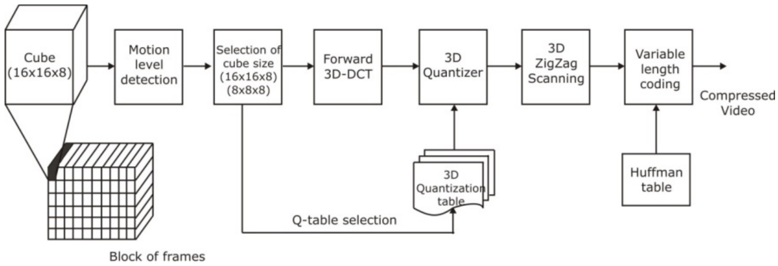


Fig.1. Block diagram of adaptive 3D-DCT encoder

$$Where\ f_r,\ f_c,\ f_d\ = \begin{cases} \frac{1}{\sqrt{2}}\ for\ r,\ c,\ d\ =\ 0 \\ 1\ \ otherwise \end{cases}$$

where F(R,C,D) and f(r,c,d) represent the frequency domain and time domain intensity values. The inverse three dimensional discrete cosine transform can be represented by

(3)
$$f(r,c,d) = \sqrt{\frac{8}{N_r.N_c.N_d}}\, f_r f_c f_d \sum_{R=0}^{N_r}\sum_{C=0}^{N_c}\sum_{D=0}^{N_d} F(R,C,D).$$
$$\frac{Cos(2r+1)R\pi}{2N_r}\cdot\frac{Cos(2c+1)C\pi}{2N_c}\cdot\frac{Cos(2d+1)D\pi}{2N_d}$$

Several fast algorithms are available to implement 3D-DCT efficiently [13].

4. 3D Quantization table and 3D Zigzag ordering

Quantization plays a vital role in the compression technique. In order to transmit the video sequence with minimum distortion and bit rate, proper selection of values for the quantization table is required. We cannot use the 2D quantization table that is stated for standard video compression technique because we are taking 3D-DCT. So we need a 3D Quantization table. The entries in the table are chosen by analyzing the dynamic range of DC and AC coefficients [14].

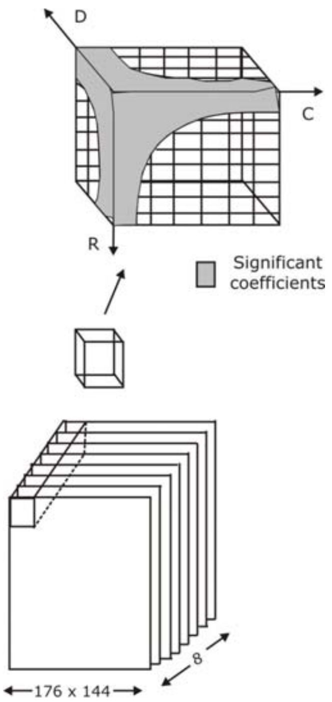


Fig.2. Distribution of significant coefficients

It is observed that 80% of total energy of the cube is spread along the major axis (R, C, D) as shown in Fig. 2. These coefficients are said to be significant coefficients [14,15]. Significant coefficients will be given due importance during quantization and it is stated by the equation (4).

(4)

(4) $$F(R,C,D) = \begin{cases} \text{Significant:} R + C + D \le k \\ \text{Insignificant:} R + C + D > k \end{cases}$$

where R, C, D is integer whose value ranges from 1 to 8 or 16 and k = 3, 4… (R + C + D). The maximum length of a quantized value as is been fixed as 8 bits. Detailed analyses are made and the value of significant coefficients ranges between 8 and 16 and for insignificant coefficients the value ranges between 45 and 250. To achieve high compression ratio the encoder requires effective ordering of sequences [16,17,18]. We used the equation (4) for reordering 3D quantized data into 1D array in the increasing frequency order direction. Further rate reduction can be achieved by encoding the sequence using variable length coding technique [19].

5. Experimental results

5.1. Experimental setup

Few test sequences [20] have been chosen for simulation; each having different category of motion levels. All the sequences have 296 frames with a frame rate 30 frames/sec and with dimension 176 × 144. Peak Signal to Noise Ratio (PSNR) is used as a measure to determine the video quality of the compressed video. We have taken only the luminance component [Y] for analysis. So in the entire figure, video quality is represented as Y – PSNR. The equation for finding the PSNR is given in (5) and (6).

(5) $$PSNR\ =\ 10\log_{10}\left[\frac{255^2}{Mean\ Squared\ error}\right] db.$$

(6) $$Mean\ Squared\ error = \frac{1}{N_r N_c N_d}\sum_{r=0}^{N_r}\sum_{c=0}^{N_c}\sum_{d=0}^{N_d}.$$
$$[f(r,c,d) - \bar{f}(r,c,d)]^2$$

where $f\ (r,\ c,\ d)$ represent the original frame and $\bar{f}(r,\ c,\ d)$ represent the reconstructed frame and $N_r \times N_c$ represent the frame size, N_d represent the number of frames in the sequence.

5.2. Motion level and cube size determination

The existing algorithm uses NPD values for determining the motion level of the cubes. However the continuous motion change between 2nd to 7th frames is left unnoticed. We propose a new energy based algorithm for determining the motion level of the cube. The cube for which the motion level has to be determined is subjected to 3D-DCT. We know DCT is having excellent energy compaction property. Once it is taken along all the three dimensions the coefficients are concentrated along the major axis of the cube (R, C, D) as shown in Fig. 2.

Table 1. Distribution of motion level for standard test video sequence

Name of the test sequence	Number of cubes in each category for the proposed algorithm			Number of cubes in each category for NPD based algorithm		
	Low motion category	Medium motion category	High motion category	Low motion category	Medium motion category	High motion category
akiyo	3454	163	46	3491	172	0
coast guard	2457	556	650	265	2881	517
hall	3052	567	44	3032	631	0
Container	3270	380	13	3098	565	0
News	893	2593	177	2205	1413	45
Mother	2761	902	0	2664	999	0
Foreman	3089	485	89	110	2676	877
Salesman	1598	1908	157	2960	703	0
Clairenews(three)	2367	1230	66	3264	399	0
Soccer (Out off 1782 cubes)	842	560	380	56	911	815

Detailed analysis was made on different video sequence of length 296 frames of dimension 176 × 144. Based on the number of coefficients required to reconstruct the 99% of cube energy the cubes are classified as low, medium, high motion cubes. For classification we adopted K-means clustering algorithm, and the threshold was set for different motion cube. It is found that for a cube of dimension 16 × 16 × 8, to reconstruct 99% of cube energy if the coefficient count is less than 31 it is classified as low motion cube. If the coefficient count is in between 32 and 101 it is classified as medium motion cube else it is high motion cube. Table 1 lists the distribution of different categories of motion for the standard set of test sequences for the proposed algorithm and the existing algorithm.

Table 2. Cube size for different motion category

Category of motion	Proposed algorithm / NPD based algorithm	Fixed cube algorithm [8,9,10]	Variable temporal length algorithm[11]
Low motion	[16 × 16 × 8]	[8 × 8 × 8]	[16 × 16 × 1]
Medium motion	[8 × 8 × 8]	[8 × 8 × 8]	[16 × 16 × 8]
High motion	[8 × 8 × 8]	[8 × 8 × 8]	[8 × 8 × 8]

It clearly shows that there is a significant change in predicting the motion of the cube between proposed energy based and the existing NPD based algorithms. The effectiveness of the proposed algorithm is verified by rate Vs distortion analysis. It is done by adopting suitable cube size for different motion category.

Cubes fall under different motion category are allotted different cube dimension in order to achieve high compression. To determine the cube dimension few test sequences are chosen. From the Table 1 we can see that for "akiyo" sequence majority of cube fall under the low motion category. For low motion cubes the neighboring pixel values are highly correlated, while taking DCT it is the property of DCT that whatever may be dimension of the cube more than 80% of energy is concentrated only in few coefficients. So only few coefficients are needed if we replace the fixed size [8 × 8 × 8] cube by [16 × 16 × 8] cube. Also the process complexity is reduced because of the chosen cube dimension [16 × 16 × 8] instead of [8 × 8 × 8]. So if a cube is identified as low motion cube then the optimal cube dimension will be [16 × 16 × 8]. Similarly we tried different cube dimension like [4 × 4 × 8] and [8 × 8 × 8] for the remaining sequence and arrived the conclusion shown in Table 2. Here the dimension of the cube are varied spatially, the possible variation are [16 × 16 × 8], [4 × 4 × 8] and [8 × 8 × 8]. If the cube dimension is varied temporally say [8 × 8 × 4], requires additional processing, also effective energy compaction in the temporal domain is not achieved that will tend to increase number of coefficients to be encoded that increases data rate. So in our analysis we have considered only the spatial variation.

Table 3 and Table 4 illustrate the Rate Vs distortion analysis between 3D-DCT and NPD based algorithm. Here, for comparison we have taken few video sequences that have a drastic shift in the count of low, medium, and high motion cubes. By adopting 3D-DCT based cube determination algorithm there is a significant improvement in the PSNR value when compared against NPD based compression algorithms.

Table 3. Comparison of Rate Vs Distortion analysis for the proposed and NPD based algorithm

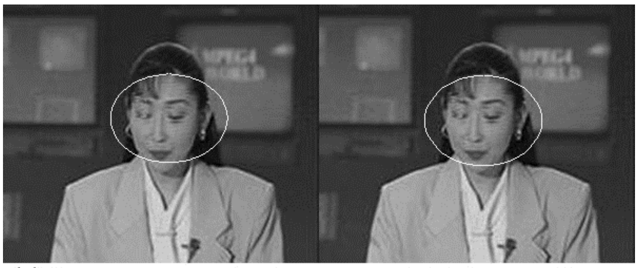
Coastguard				News			
Proposed 3D-DCT based algorithm		NPD based algorithm[12]		Proposed 3D-DCT based algorithm		NPD based algorithm[12]	
Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)
123.1781	30.8182	188.4439	31.8362	327.4332	35.7481	281.7401	31.8077
129.1713	31.5039	200.2887	32.2758	344.6109	36.6582	297.4364	32.3039
142.1990	32.8909	220.2661	33.1956	358.1989	37.3740	320.5655	33.2188
149.1142	33.6328	237.5555	34.2977	368.4308	37.8441	336.4402	34.1730
156.1852	34.4119	251.6614	35.3934	376.6879	38.2445	347.9668	35.1088
163.3204	35.2000	262.7136	36.4217	390.6186	38.9133	356.7888	36.0109
177.4161	36.6478	271.2299	37.3765	396.6785	39.1827	363.6152	36.8313
184.1932	37.2363	279.6320	38.4410	415.4608	39.9661	368.5153	37.4895
196.7470	38.2015	282.8763	38.8237	418.7775	40.0677	395.7824	37.9333
212.0676	39.2513	284.4058	39.0104	429.4186	40.3057	421.5681	38.1855

Table 4. Comparison of Rate Vs Distortion analysis for the proposed and NPD based algorithm

Foreman				Soccer			
Proposed 3D-DCT based algorithm		NPD based algorithm[12]		Proposed 3D-DCT based algorithm		NPD based algorithm[12]	
Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)	Data rate Kbps	PSNR (db)
252.5067	28.3483	402.5818	28.5657	492.5162	29.2334	615.9766	29.4204
305.4471	29.3156	435.1937	29.6731	534.9997	30.0687	682.6321	30.3355
359.0143	30.5428	453.9456	30.3515	601.6391	31.6749	744.5770	31.2751
385.0946	31.2714	474.3280	31.1476	625.5945	32.3551	801.4603	32.1789
432.1371	32.8669	495.9334	32.0757	644.6420	32.9158	852.8134	33.0188
451.8466	33.6432	891.2402	33.1492	659.4919	33.3809	899.3237	33.7540
468.4738	34.3864	965.5917	34.3365	681.4133	34.1346	941.6382	34.4086
482.2659	35.0070	1035	35.6325	696.9281	34.7154	980.4620	35.0068
501.3565	35.8380	1099	37.0856	710.9641	35.1912	1049.1	36.1255
515.6555	36.2524	1157	38.5975	717.4210	35.3453	1165.7	37.8699

5.3. Video compression

The effectiveness of the proposed algorithm has been verified by comparing the performance with the existing 3D-DCT algorithm and the standard compression algorithm with the help of Rate Vs Distortion analysis. Without any motion compensation the proposed algorithm outperforms the standard MPEG-2 algorithm. However if we design the 3D-DCT coder with the motion compensation capability the proposed algorithm could outperforms the standard H.264 algorithm. This paper reports only the adaptive algorithm without motion compensation technique, so we considered existing 3D-DCT and MPEG-2 algorithm for comparison. Fig. 3 shows the comparison of proposed algorithm with variable temporal length compression algorithms for ‘akiyo” sequence. The variable temporal algorithm uses different cube size with variable temporal length. It is observed that this algorithm suffer from blocking artifact effect.



(a) Sequence reconstructed using variable temporal length algorithm with PSNR 35.58 db

(b) Sequence reconstructed using proposed algorithm with PSNR 35.66 db

Fig.3. Shows the reconstructed frame 69,70 of “akiyo” sequence by variable temporal length and proposed algorithms

It is indicated by the circle shown in Fig. 3. The reason is it uses two different temporal length cubes [16 × 16 × 1] and [8 × 8 × 8]. According to the encoding process explained in the variable temporal length algorithm, once the cube of dimension [16 × 16 × 8] is identified to have minimum motion, and then only the first frame is considered for encoding the entire cube of data. While decoding, the first frame is copied to the remaining 7 frames. This causes blocking artifact effect. Because all natural video frames have slight variation in the intensity level with respect to successive frames. So it is not an effective method if we consider only the first frame as a replacement of entire cube. However minimum motion is experienced in all the remaining frames with reference to the first frame. This blocking artifact effect is strongly pronounced at all levels of PSNR value. Also for the desired video quality it requires higher data rate compared to the proposed algorithm. Because of the above mentioned reasons it is not considered for evaluating the performance of the proposed algorithm.

.

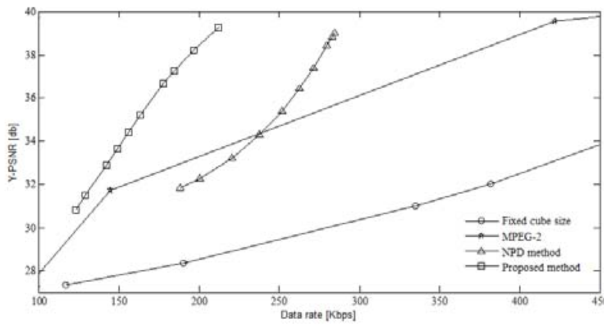


Fig.4. Rate Vs Distortion plot for “coastguard” sequence

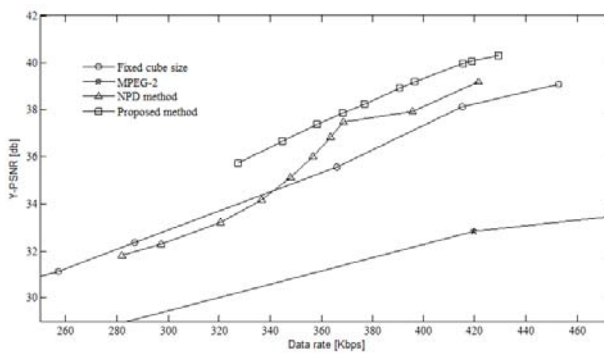


Fig.5. Rate Vs Distortion plot for “news” sequence

Though the cubes adopted by proposed and NPD based algorithms for different motion category is same [refer Table 2], to prove the efficiency of the proposed algorithm we considered video sequences having greater variations in low and high motion category like “coastguard” and “News” sequence. Rate distortion values of the video sequence are plotted against existing and standard compression video compression algorithms like MPEG-2 and it is observed that the proposed algorithm outperforms the other compression algorithms. Rate distortion analysis plot of “coastguard” sequence is illustrated in Fig. 4

For the given PSNR value of 39 db the proposed algorithm requires 50% lower data rate compared to standard MPEG-2 algorithm and 24% low data rate than NPD based algorithm. Similarly for “News” sequence as illustrated in Fig. 5 for the given PSNR value of 38.8 db the propose algorithm requires 58% lower data when compared to MPEG-2 algorithm and 12% lower data rate than NPD based compression algorithm. Also the compression ratio (CR) of proposed algorithm ranges from 15: 1 to 160: 1 that mainly depend on the desired video quality.

5.4. Process complexity

One more method to measure the performance of the proposed algorithm is the process complexity. Here we considered the process complexity as the time consumed by algorithm to encode and decode all the 296 frames that is fed at the input. Because of the blocking effect the variable temporal length algorithm is not considered for comparison. Also in MPEG-2 it has complex motion detection and estimation algorithm so it not reasonable to compare it with 3D-DCT based compression algorithm. It is observed that for fixed cube size algorithm we require 420.83 seconds to encode and decode the video sequences. However for the proposed algorithm as pointed in Table 2 it adopts the cube dimension [16 × 16 × 8] instead of [8 × 8 × 8] for low motion cubes. This made the proposed algorithm execute faster compared to the other algorithm. Also it clearly states that if any video sequence has more number of low motion cubes then execution speed is increased. (i,e.,) there is a proportional variation in execution speed with respect to the number of low motion cubes. But in case of fixed cube algorithm the execution speed is around 420 seconds and it is constant. The processing time of the proposed algorithm varies between 212.65 seconds to 302.75seconds. For all category of video sequence by adopting cube size dynamically there is a significant improvement in the data rate for the given PSNR and process complexity when compared to the existing 3D-DCT based compression algorithm.

6. Conclusion

We present an adaptive cube selection algorithm for 3D-DCT based video compression technique that will dynamically select the cube size relative to the motion level of video sequence. Motion level is determined by energy based adaptive 3D-DCT based method, based on that suitable cube size is chosen. Various test sequences were taken for evaluating the proposed algorithm by performing Rate Vs Distortion comparison with the fixed cube size algorithm. Experimental result reveals that the proposed algorithm outperforms MPEG-2 and other existing 3D-DCT based compression algorithms. Better performance is achieved by adopting variable cube size for low [16 × 16 × 8], medium [8 × 8 × 8] and high motion [8 × 8 × 8] cubes which in turn reduces the process complexity of the algorithm and significant rise in the PSNR value. Because of its high compression ratio and reduced process complexity the proposed algorithm is well suited for video surveillance application.

REFERENCES

[1] Jiang J., Xia J., Xiao G., MPEG-2 Based Lossless Video Compression, *IEE P-Vis. Image Sign.*, 153 (2006), No. 2, 244-252

[2] AlQaralleh E. A., Chang T., Lee K., An Efficient Binary Motion Estimation Algorithm and its Architecture for MPEG-4 Shape Encoding, *IEEE T. Circ. Syst. Vid.,*16 (2006), No. 7, 859-868

[3] Telceken S., Gerek O.N., A Video Encoder Design Combining Edge-Adaptive Lifting and Scalable Block-Matching with Motion Compensation, *Turk J. Elec. Eng. & Comp. Sci.,* 20 (2012), No. 1, 111-123

[4] Ahmed N., Natarajan T., Rao K. R., Discrete Cosine Transform, *IEEE T. Comput.,* C-23 (1974), No. 1, 90-93

[5] Hakki Ilgin , Hakki Ilk , Miroslav Voznak , Luis Chaparro, Fast DCT Image Resizing and Video Compositing for Multi-Point Video Conferencing, Przeglad Elektrotechniczny, (2013), nr 02b/2013, 193-197

[6] Rao K. R., Yip P., *Discrete Cosine Transform,* Academic Press, 2007

[7] Westwater R., Furht B., Three-dimensional DCT Video Compression Technique Based on Adaptive Quantizers, i*n Proceedings of the second IEEE International Conference on Engineering of Complex Computer Systems,* Montreal, Canada (1996), 189-198.

[8] Servais M., De Jager G., Video Compression using the Three Dimensional Discrete Cosine Transform (3D-DCT), i*n Proceedings of the 1997 South African Symposium on Communications and Signal Processing,* Grahams Town, South Africa (1997), 27-32

[9] Li L., Hou Z., Multiview Video Compression with 3D-DCT, in Proceedings of 5th IEEE International Conference on Information and Communications Technology, Cairo, Egypt (2007), 59-61

[10] Lai T. H., Guan L., Video Coding Algorithm using 3-D DCT and Vector Quantization, i*n Proceedings of the International Conference on Image Processing,* Rochester, New York, USA (2002), I-741-I-744

[11] Chan Y., Siu W., Variable Temporal Length 3-D Discrete Cosine Transform Coding, *IEEE T. Image Process.,* 6 (1997), No. 5, 758-763

[12] Furht B., Gustafson K., Huang H., Marques O., An Adaptive Three-Dimensional DCT Compression Based on Motion Analysis, i*n Proceedings of the ACM Symposium on Applied Computing*, Melbourne, Florida, USA (2003), 765-768

[13] Alshibami H. O., Boussakta S., Fast Algorithm for 3-D DCT-II, *IEEE T. Signal Proces.,*52 (2004), No. 4, 992-1001

[14] Chan R. K. W., Lee M. C., 3D-DCT Quantization as a Compression Technique for Video Sequences, i*n Proceedings of the IEEE International Conference on Virtual Systems and Multimedia,* Geneva, Switzerland (1997), 188-196

[15] Yeo B., Liu B., Volume Rendering of DCT-Based Compressed 3D Scalar Data, *IEEE T. Vis. Comput. Gr.,*1 (1995), No. 1, 29-43

[16] Bhaskaranand M., Gibson J. D., Distribution of 3D-DCT Coefficients for Video, i*n Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing,* Taipei, Taiwan (2009), 793-796

[17] Lee M. C., Chan R. K. W., Adjeroh D. A., Quantization of 3D-DCT Coefficients and Scan Order for Video Compression, *J. Vis. Commun. Image R.,*8 (1997), No. 4, 405-422

[18] Bozinovic N., Konrad J., Motion Analysis in 3D DCT Domain and its Application to Video Coding, *Signal Process-Image,* 20 (2005), No. 6, 510-528

[19] Fryza T., Scan Order and Huffman Coding of 3D DCT Coefficients, *7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering*, Sofia (2005), 235-238

[20] http://trace.eas.asu.edu/yuv/

Authors: *Prof. Augustin Jacob JACOB, Department of Electronics and Communication Engineering, Kamaraj College of Engineering and Technology, TamilNadu, India, E-mail: mailtoaugus@yahoo.co.in, Dr. Senthilkumar NATARAJAN, Department of Electrical and Electronics Engineering, MEPCO Schlenk Engineering College, TamilNadu, India, E-mail: nsk_vnr@yahoo.com.*