EFFICIENT BIT-PLANES BASED METHOD FOR COMPRESSION OF 3D-DCT COEFFICIENTS

.*Andriy V. Bazhyna*[1], *Karen O. Egiazarian,*[1] *Nikolay N. Ponomarenko*[2]
[1] Institute of Signal Processing, Tampere University of Technology, P.O. Box 553 FIN-33101, Tampere, Finland
[2] Department 504, National Aerospace University 17 Chkalova St., 61070, Kharkov, Ukraine

ABSTRACT

In this paper, a new efficient method for compression of 3D-DCT coefficients is presented. The method views at values of coefficients as a set of bit-planes. The bits of each bit-plane are classified into a number of probability models depending on the context of the bit. The use of contexts allows us to efficiently encode cubes with different localization of significant coefficients. This adaptation is performed automatically during the coding process and does not require any additional computations as well as preliminary assumptions on a nature of input data. The experimental results have shown superiority of the proposed method for a 3D-DCT based video compression in comparison to the some previously proposed approaches.

Index Terms: - 3D-DCT, compression, video, multidimensional

1. INTRODUCTION

Three-dimensional Discrete Cosine Transform (3D-DCT) is found to be a useful tool for many applications that require compression of multidimensional data, such as images and video compression [1-11]. For video compression, 3D-DCT could be used as a low-complexity alternative to the traditional hybrid 2D-DCT and motion-estimation/compensation approach which is utilized in current video coding standards [1-3,8,11].

Note, that except video compression, 3D-DCT was also successfully used for compression of still images [4,5], hyperspectral images [6,7] and full parallax colour 3D TV image data [9].

The compression efficiency of 3D-DCT based methods highly depends on a nature of the input data. However, for the same input data efficiency could be significantly improved by using an appropriate quantization and coefficient scanning method.

Traditionally, 3D-DCT based compression method uses a simple plane-by-plane 2D-zigzag scan [1], which came from the JPEG standard. An improved 3D version of the zigzag scan [10] allows in some applications slightly better compression comparing to the 2D-zigzag. More complex parabolic scan [4] and hyperboloid scan [7] were reported. They based on an assumption that significant coefficients tend to get grouped along x, y and z axes of the cube. This assumption, however, is hold only in case of small dissimilarities between frames. The amplitude based scan was proposed in [3].

All the methods mentioned above are non-adaptive. In contrast authors in [11] studied properties of 3D DCT spectrum in case of video sequences with motion. It was shown that in this case significant coefficients tend to be grouped along a plane whose orientation is defined by a dominant motion in the video sequence. Based on this study, two adaptive scanning methods were proposed. Despite the fact that adaptive methods provide better performance than non-adaptive ones, they require analysing of some features of input data (such as motion) or analysis of distribution of coefficients within the cube. All these preliminary calculations require some additional computations.

We can summarize that the 3D-DCT coefficients are tends to concentrate in some areas of the cube. Depending on nature of input data these areas could be different and additionally varying from a cube to cube. Thus, it is challenging task to find a universal coding method that would be able to efficiently compress coefficients having different localizations within a cube.

In this paper, we present a new efficient method for compression of 3D-DCT coefficients. The proposed method is based on bit-plane coding of 3D-DCT coefficients and is an extension of the bit-plane coding for 2D-DCT introduced in [12,13]. The DCT coefficients are viewed as a set of bit-planes. The bits of each bit-plane are coded using context adaptive binary coder. The context of each pixel is determined by corresponding values of neighbouring bits as well as bits from upper bit-planes. The use of contexts makes method adaptive and allows us to efficiently encode cubes where significant coefficients are located in different areas. This adaptation is done automatically during the coding process. Further computations or preliminary assumptions about a nature of the input data are not required.

The rest of the paper is organized as follows. In next section 2, basics of 3D-DCT compression are reviewed. In Section 3, proposed bit-planes based method is presented in details. In Section 4, the results of experimental evaluation of described coding method are reported. Finally, conclusions are given.

2. BASICS OF 3D-DCT COMPRESSION

The block diagram of 3D-DCT coder for multidimensional data is shown on Figure 1. First, 2D data is divided into groups of sequential frames (most commonly, 8 frames per group). Then each group of frames is partitioned into 3D cubes typically with a size 8x8x8. Later on, three-dimensional cosine transform is applied to every cube. Due to a decorrelation property of the DCT, most of the energy is concentrated in a small number of coefficients. The quantization is used to remove some of less relevant, for observer, information and to facilitate farther compression. Finally, the entropy encoding is performed. For the reconstruction of the data, the above described steps inverted and executed in the reverse order.

In order to be efficient, the entropy encoding is done separately for the different components of DCT spectrum. The DC coefficient could be efficiently predicted from the neighbouring blocks. Thus, only difference between predicted and the real value is encoded. The signs of DCT coefficients are assumed to be practically random variable with approximately equal probability to be positive or negative. Therefore, each sign is coded using one bit.

The most significant part of a bitstream is utilized for AC coefficients position and absolute value information. The AC coefficients are encoded via scanning and subsequent zero-run-length coding using variable-length codes. The purpose of scanning is to order quantized coefficients into vector suitable for entropy coding.

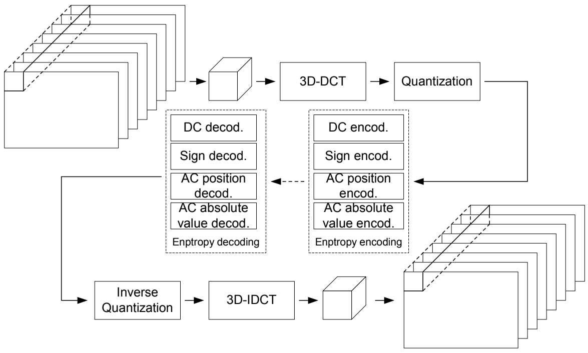

Figure 1 – Block diagram of 3D-DCT coder for multidimensional data.

The good scanning method should be simple and preferably predefined, from one side. On the other hand, it should efficiently group significant coefficients at the beginning of the sequence allowing to zero-run-length coding be applied in a most efficient way. However, in practice, localization of significant coefficients within the cube could be very complex even for the case of a small dissimilarity between frames, as illustrated in Figure 2. Thus, often, scanning method could not efficiently group significant coefficients. This leads to non-optimal coefficient ordering and shorter zero-runs, which significantly reduces efficiency of the compression.

In next section, we propose an alternative approach to coefficient position and absolute value coding which potentially could resolve problems mentioned above.

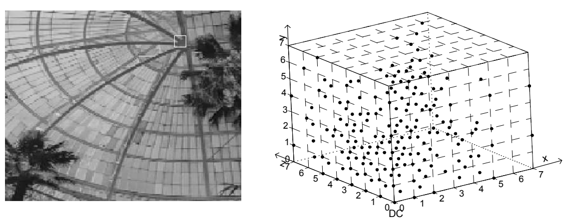

Figure 2 – Illustrative example: part of Glasgow test sequence with slow camera rotation in horizontal direction. Frame 30 of test sequence with selected block (left). Distribution of significant 3D-DCT coefficients within cube that corresponds to selected block (right).

3. PROPOSED METHOD

After the calculation of 3D-DCT in 8x8x8 blocks and quantization of obtained coefficients, we have cubes contained integer valued DCT coefficients. Let us divide the cube of absolute values of DCT coefficients into N bit-planes, where N is the number of highest bit-plane that contains non-zero values. (It would be more appropriate to use term “bit-cubes”, however “bit-planes” is used to follow terminology introduced used in [12,13]).

This paper is not focusing on encoding of DC coefficients. Thus, they are transferred to output stream directly using appropriate number of bits. The signs of non-zero coefficients are stored using 1 bit per sign.

In [12] it was proposed to code bit-planes starting from the bit-plane N (most significant) and ending by the bit-plane 1 (least significant). This approach, however, has one limitation. It is relatively slow, because it checks all bits at every bit-level even if DCT coefficient is zero. In order to overcome above limitation in our method significance map coding is introduced. That is, at the beginning, a single bit-plane is coded, where each bit indicates whether DCT coefficient is significant on this position or not. Later, bit-plane coding is applied only at positions indicated by a significance map. The achieved speed up factor is proportional to K_{tot}/K_{nz}, where K_{nz} and K_{tot} are number of non-zero coefficients and the total number of DCT coefficients, respectively.

Later in this section, a significance map and absolute value encoding is described.

3.1 Significance map coding

Let $S_{l,m}(x,y,z)$ defines a bit value of a significance map with the index $x,y,z=0..7$ within the cube in a set of 8 frames with the index $l=0..L-1$, $m=0..M-1$ (L,M are number of cubes in horizontal and vertical direction).

The bits at positions with indexes x=y=z=0 corresponds to the DC coefficient of the cube and assumed to be always significant and, thus, they are not coded. The bits at the rest of positions are classified into a number of probability models using the flowchart presented in Figure 3. The conditions C_1-C_{16} are following.

$C_1(l,m,x,y,z)$=true, *if* z=0. $C_2(l,m,x,y,z)$=true, *if* x=0. $C_3(l,m,x,y,z)$=true, *if* y=0. $C_4(l,m,x,y,z)$=true, *if* y=1. $C_5(l,m,x,y,z)$=true *if* x=1. $C_6(l,m,x,y,z)$=true *if* z=1. These conditions separate bits that belonging to plane of the cube defined by corresponding equation. The reason is that statistics for these bits is significantly differs from statistics in others parts of the cube.

$C_7(l,m,x,y,z)$=true, *if* $(x>1$ & $y>1)$. $C_8(l,m,x,y,z)$=true, *if* $(y>1$ & $z>1)$. $C_9(l,m,x,y,z)$=true, *if* $(x>1$ & $z>1)$. These conditions separate bits that belonging to some volumes inside the cube defined by corresponding equation. The reason is same as for conditions C_1-C_6.

$C_{10}(l,m,x,y,z)=1$, *if* $1\in\{S_{l-1,m}(x,y,z), S_{l,m-1}(x,y,z), S_{l-1,m-1}(x,y,z)\}$. This condition is true if there were significant bits at neighbouring and already coded blocks at the same position as current bit. This condition allows for take to account correlation between neighbouring cubes.

$C_{11}(l,m,x,y,z)= S_{l,m}(x-1,y,z)+ S_{l,m}(x,y-1,z)+ S_{l,m}(x,y,z-1)+ S_{l,m}(x-1,y-1,z)+ S_{l,m}(x,y-1,z-1)+ S_{l,m}(x-1,y,z-1) + S_{l,m}(x-1,y-1,z-1)$. This condition is equal to number of non-zero bits among neighbouring and already coded bits. If some of the positions are not available for current coded bit they are just not checked.

$C_{12}(l,m,x,y,z)=C11(l,m,x,y,z)+ S_{l,m}(x-2,y,z)+ S_{l,m}(x,y-2,z)+ S_{l,m}(x,y,z-2)+ S_{l,m}(x-2,y-2,z)+ S_{l,m}(x-2,y,z-2)+ S_{l,m}(x,y-2,z-2)+ S_{l,m}(x-1,y-2,z)+ S_{l,m}(x-2,y-1,z)+ S_{l,m}(x,y-2,z-1)+ S_{l,m}(x,y-1,z-2)+ S_{l,m}(x-1,y,z-2)+ S_{l,m}(x-2,y,z-1)$. This condition additionally takes to the account number of non-zero bits at already coded positions and displaced from the coded bit by 2 rows and 2 columns. If some of the positions are not available for current coded bit they are just not checked.

$C_{13}(l,m,x,y,z)=1$, *if* $1\in\{S_{l,m}(x-1,y,z-1), S_{l,m}(x,y-1,z-1), S_{l-1,m}(x-1,y-1,z-1)\}$. $C_{14}(l,m,x,y,z)=1$, *if* $1\in\{S_{l,m}(x-1,y,z-1), S_{l,m}(x-1,y-1,z), S_{l-1,m}(x-1,y-1,z-1)\}$. $C_{15}(l,m,x,y,z)=1$, *if* $1\in\{S_{l,m}(x,y-1,z-1), S_{l,m}(x-1,y-1,z), S_{l-1,m}(x-1,y-1,z-1)\}$. $C_{16}(l,m,x,y,z)=1$, *if* $1\in\{S_{l,m}(x-2,y,z), S_{l,m}(x,y-2,z), S_{l-1,m}(x,y,z-2)\}$. These conditions allow classify bits more accurately.

The use of combinations of conditions allows to identify some region inside the cube more precisely. For example, the C_1 separate bits belonging to the front plane of the cube. A combination of C_1 and C_2, or C_1 and C_3 separate bits belonging to the first row and the first column of the front plane and so on. This is needed because for the bits located near the boarders of the cube less number of neighbouring bits is available. As a result, distribution of bits between models differs significantly.

In the case if DCT coefficient coded as a significant, the 1 is subtracted from its absolute value at the encoder. This 1 is added back at the decoder after decoding of absolute value. The bits belonging to every probability model are coded using dynamic version of the binary arithmetic encoder [14].

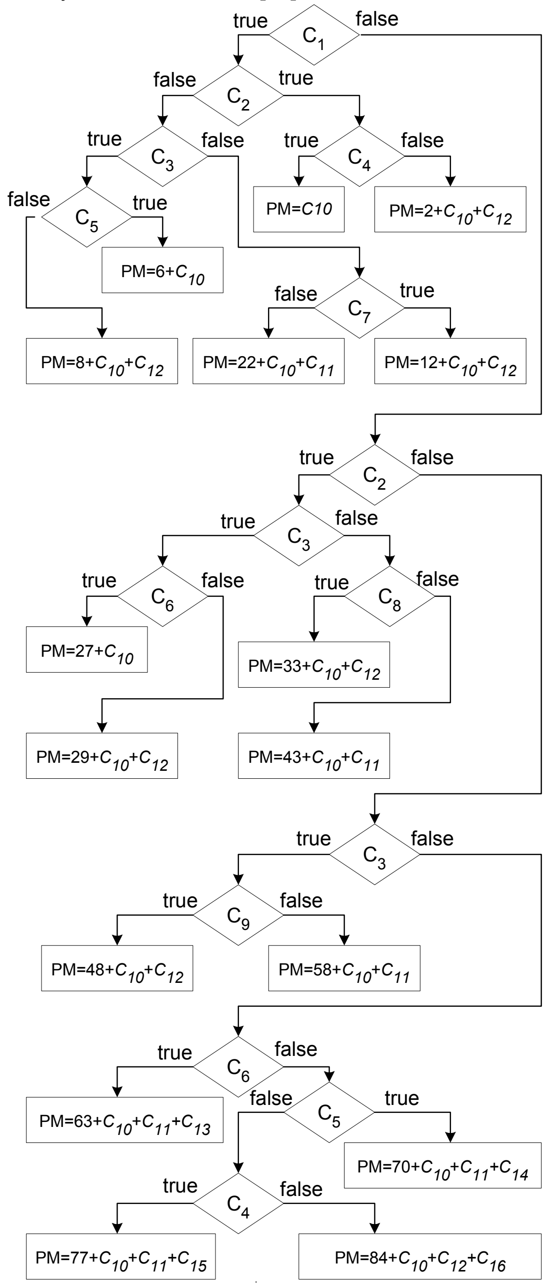

Figure 3 – Flowchart of classifying significance map bits, where PM denotes a probability model number and C_1-C_{16} are conditions (see text).

3.2 Coding of absolute values

After compression of significance map, the coding of absolute values is started. The coding is done only for positions indicated by the significance map. The coding starts from the bit-plane N and ends by bit-plane 1. Let $B^k_{l,m}(x,y,z)$ defines a bit value of a DCT coefficient at bit-level $k=1..N$, where N is highest bit level. The indices x,y,z,l,m are defined as earlier for the significance map. The bits of every bit-plane are classified into a number of probability models using flowchart presented on Figure 4. The conditions C_{17}-C_{23} are following.

$C_{17}(k,l,m,x,y,z)$=true, *if* $1\in \{B^{k+1}_{l,m}(x,y,z),\ldots, B^N_{l,m}(x,y,z)\}$. This condition separates bits those higher bit-planes contains non-zeros bits. This means that higher bits of this coefficient have been already coded and it is practically impossible to predict lower bits.

$C_{18}(k,l,m,x,y,z)$=true, *if* $1\in \{ C_{17}(k,l,m,x-a,y-b,z-c) \mid a,b,c \in \{-1,1\}\}$. This condition is true if there neighbouring coefficients, which were coded at higher bit-levels.

$C_{19}(k,l,m,x,y,z)$=true, *if* $1\in \{ C_{17}(k,l-1,m,x,y,z), C_{17}(k,l,m-1,x,y,z), C_{17}(k,l-1,m-1,x,y,z)\}$. This condition is true if there were significant bits at upper bit-planes in neighbouring and already coded blocks at the same position as current bit.

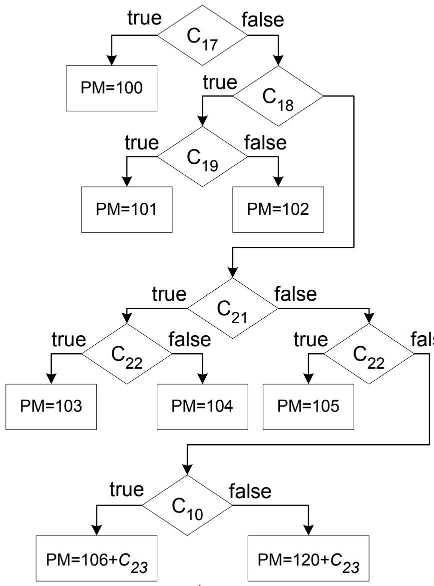

Figure 4 – Flowchart of classifying absolute value bits, where PM denotes a probability model number and C_{10}, C_{17}-C_{23} are conditions (see text).

$C_{20}(k,l,m,x,y,z)$=true, *if* $B^{k+1}_{l,m}(x,y,z)=1$.

$C_{21}(k,l,m,x,y,z)$=true, *if* $1\in \{ C_{20}(k,l,m,x-a,y-b,z-c) \mid a,b,c \in \{-1,0\}\}$. This condition is true if there are non-zero neighbouring bits at the current bit-plane.

$C_{22}(k,l,m,x,y,z)$=true, *if* $1\in \{ C_{20}(k,l-1,m,x,y,z), C_{20}(k,l,m-1,x,y,z), C_{20}(k,l-1,m-1,x,y,z)\}$. This condition is true if there are non-zero bits in the neighbouring and already coded blocks in the current bit-plane.

$C_{23}(k,l,m,x,y,z)$=k. Used to additionally classify bits that do not satisfy some of previous conditions. Utilizes the fact that for higher bit-planes probability of non-zero bit is lower.

Similarly to the process of coding of significance map, the bits belonging to every probability model are coded using dynamic binary arithmetic encoder. The conditions C_{10}, C_{19} and C_{22} take into account correlation between neighbouring cubes. If such a dependency is not desirable for some reason, one could just skip checking of these conditions. This leads to the increase of the compressed file size by about 3-5%

4. EXPERIMENTAL RESULTS

The coding efficiency of the proposed method was analysed for compression of test video sequences in QCIF format. The video sequences were Glasgow, Akiyo, Foreman and Carphone. The proposed method is compared to other scanning/coding methods: plane-by-plane 2D-zigzag scan [1], 3D-zigzag scan [10], hyperboloid based scan [7] and absolute value based scan [3].

The compression was done only for the luminance components. The sequences first were split into groups of 8 frames. Then 3D-DCT with size 8x8x8 was applied to every group of frames. After that, all coefficients were uniformly quantized by quantization factor (Q). The quantized DC coefficients and signs of AC coefficients were transferred to output bit-stream without any compression. The quantized AC coefficients were compressed by each of the three methods. The obtained compression ratios for all methods, quantization factors and sequences are presented in Table 1.

Table 1. – Compression results. 1- 2D zigzag, 2- 3D zigzag, 3- absolute value based scan, 4- hyperboloid based scan, 5- proposed.

Q	Video sequence	1	2	3	4	5
8	Glasgow	4.50	4.53	4.49	4.29	6.18
	Akiyo	23.2	23.0	21.7	20.7	29.6
	Foreman	5.79	5.94	5.73	5.12	8.00
	Carphone	7.72	7.95	7.75	7.37	10.9
16	Glasgow	8.24	8.33	8.32	7.90	11.6
	Akiyo	37.9	37.0	36.1	35.4	48.5
	Foreman	11.0	11.4	11.2	10.5	15.7
	Carphone	15.0	15.7	15.1	14.3	21.4
32	Glasgow	16.7	17.5	17.2	17.0	24.9
	Akiyo	65.0	64.7	62.8	62.6	83.5
	Foreman	23.0	23.8	24.5	22.7	33.9
	Carphone	31.9	32.8	32.0	31.1	46.3

As one can see, the proposed method is able to provide a increasing of compression ratio by 1.27-1.41 times. The amount of the increasing mainly depends from the type of video sequence rather than from the compression ratio. For the sequences with a fast stochastic motion, the DCT coefficients are widely distributed within the cube. The proposed method localizes better such areas, and, as a result, codes them more efficiently. For low-motion videos, like Akiyo, most of 3D-DCT coefficients localized in a few front planes and could be quite well captured by a simple 2D-zigzag scan. For this reason, reduction in bitrate is smaller, but, nevertheless, still significant.

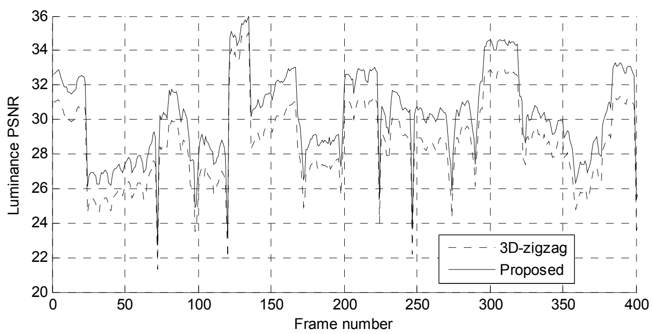

Figure 5. – Frame-by-frame PSNR of compressed Glasgow video sequence

The frame-by-frame PSNR of the part of Glasgow sequence compressed by the proposed method with Q=32 is presented in Figure 5. For comparison, Figure 5 also depicts PSNR of this part of a sequence compressed by the 3D-zigzag to have the same file size. As it could be seen, the proposed coding method provides PSNR gain by 0.72-2.05 dB (1.56 dB in average).

5. CONCLUSIONS

In this contribution a new efficient method for compression of 3D-DCT coefficients was presented. The proposed method is based on coding of significance map for indication of non-zero coefficients. This is followed by a bit-plane based coding of absolute values of 3D DCT coefficients at positions specified by the significance map. The bits of significance map and absolute values are divided into a number of probability models using a set of conditions. These conditions are taking into account context in which the current bit is occurred. The use of contexts allows to efficiently encoding of 3D-DCT cubes with different distributions of significant coefficients within the cube. The experimental results have demonstrated that usage of the proposed method for 3D-DCT based video compression provides reduction of bitrate by the factor 1.27-1.41 comparing to the traditional plane-by-plane 2D-zigzag and 3D-zigzag approaches. This is equivalent to up to 2 dB improvement in PSNR for the same bitrate. The proposed method could be used for compression of other types of multidimensional data with 3D-DCT. Also it could be integrated to any existing 3D DCT based codec.

REFERENCES

[1] M. Servais, and G. De Jager, "Video compression using the three dimensional discrete cosine transform (3D-DCT)", in *Proc. of COMSIG '97*, South Africa, 1997, pp.27-32.

[2] M. Lee, R. Chan, and D. Adjeroh, "Quantization of 3D-DCT coefficients and scan order for video compression," *Journal of Vis. Commun. Image Represent*, No. 8, pp. 405-422, Dec. 1997.

[3] D. Furman and M. Porat, "Video Compression in the Cosine Time-Spatial Domain using Human Vision Characteristics", *IAPR/IEEE Conf. on Image and Vision Comp.*, New Zealand IVCNZ'01, pp. 333-337 (2001).

[4] S. Tai, Y. Wu, and C. Lin, "An adaptive 3-D discrete cosine transform coder for medical image compression", *IEEE Trans. Inf. Tech. in Biomedicine*, vol. 4, pp. 259-263, Sept 2000.

[5] L. Xiuqi, F. Borko, "An Approach to Image Compression Using Three-Dimensional DCT," in *Proc. of VIS2003*, Miami, Florida, USA September 24-26, 2003.

[6] G. Abousleman, M. Marcellin, and B. Hunt, "Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCMIDCT," *IEEE Transactions on Geoscience and Remote Sensing*, Vol 33, No. 1, pp. 26-34.

[7] D. Markman, D. Malah "Hyperspectral image coding using 3D transforms," in *Proc. of the ICIP2001*, Thessaloniki, Greece, October 7-10, 2001. Vol. 1 pp. 114-117.

[8] D. Rusanovskyy, K. Egiazarian, "Postprocessing for three-dimensional discrete cosine transform based video coding", *Proc. of 2005 Int. TICSP Workshop on Spectral Methods and Multirate Signal Processing*, SMMSP 2005, Riga, Latvia, 20-22 June 2005, pp 145-150.

[9] R. Zaharia, A. Aggoun, and M. McCormick, "Adaptive 3D-DCT compression algorithm for continuous parallax 3D integral imaging," *Journal of Signal Processing: Image Communication* Vol. 17, pp. 231-242, 2002.

[10] B. Yeo and B. Liu, "Volume rendering of DCT-based compressed 3D scalar data," *IEEE Trans. Vis. And Comp. Graph*, Vol. 1, No. 1, pp. 29-43, 1995.

[11] N. Bozinovic, J. Konrad, "Scan order and quantization for 3D-DCT coding" in *Proc. of SPIE Vis. Comm. and Im. Proc*. Vol. 5150, pp. 1204-1215, 2003.

[12] N. Ponomarenko, V. Lukin, K. Egiazarian, and J. Astola, "DCT Based High Quality Image Compression", in *Proc. of SCIA2005*, Joensuu, Finland, pp. 1177-1185, June, 2005.

[13] N. Ponomarenko, K. Egiazarian, V. Lukin, and J. Astola, "Additional lossless compression of JPEG images", in *Proc. of 4th Symposium on Image and Signal Processing and Analysis*, Zagreb, Croatia, September 2005, pp. 117-120.

[14] G. Langdon, and G. Rissanen, 'A simple general binary source code' *IEEE Trans. Inf. Theory*, 1982, pp.800-803