International Journal of Pure and Applied Mathematics
Volume 119 No. 14 2018, 29-34
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Effect of Block Size on the Performance of a new Algorithm to Compress an Image using 3D DCT

Surekha R Gondkar[1], Dr.Girish V Attimarad[2], Dr.B Chandrasekhar[3]

[1]Dept. of TE, BMS Institute of Technology and Management,Bengaluru-560064, India
[2]Dept. of ECE, KS School of Engineering and Management, Bengaluru-560109, India
[3]Dept. of CSE, RR Institute of Technology, Bengaluru, 560 090, India
**Corresponding Authors and E-mails:*
*[*1]gattimarad@gmail.com*
*[*2]surekha.r.gondkar@bmsit.in*
*[*3]drchandrasekhar.b@gmail.com*

Abstract

Image compression is a technique to reduce the image file size without affecting the visual quality considerably. In this paper a new technique to compress an image based on the Three-Dimensional Discrete Cosine Transform(3D DCT) is presented. In this method the input image is first partitioned into N sub images and are grouped together to form an N×N×N data cube. Each data cube is transformed using 3D DCT, and quantized. The resulting coefficients are zigzag scanned and entropy encoded to get the compressed image. In decompression the subimages are first reconstructed by applying the inverse operations in the reverse order. The full size image is then obtained by concatenating the reconstructed subimages. The new algorithm is tested on different images for various block sizes. The quality parameters are evaluated in each case. The results for the same are presented and analyzed in this paper.

Keywords— *Compression, 2D-DCT, 3D-DCT, MAD, .MSE, PSNR, CR.*

1. Introduction

In today's world enormous amount of images are being transferred over internet, mobiles, computers, hd tv's, tablets etc,. Generally large number of bits is required to represent the images. It is impractical to store or transfer the images, without reducing their size as it increases the storage requirements, communication bandwidth and hence cost. This has increased the need to develop techniques to store and transmit the data efficiently.

The basic objective of image compression is to reduce the redundancies in image representation with acceptable reconstructed image quality. This not only minimizes the memory requirements, but also reduces the cost of communication. Also reduction in the memory requirements results in storage of more data in less space and hence increases the communication bandwidth [1].

Image compression basically aims at reduction of redundant and irrelevant information in the image. The redundancy reduction is achieved by removing duplication from the input image. The irrelevancy reduction removes those details of the input image for which the Human Visual System (HVS) is insensitive, that is the distortion introduced does not affect the visual perception [2]. As a result, some difference in the regenerated pixel values may be allowed as the HVS will not find any noticeable difference between original and the reconstructed images [3].

The performance of Discrete Cosine Transform (DCT) is very close to that of statistically optimal Karhunen-Loeve transform[4], and hence it has been used as a kernel in many standards like JPEG (e.g., [5], [6]) for compression of still images, MPEG-1 [7], MPEG-2 [8] and H.263 [9] for video coding, CCITT H.261 (also known as Px64), for video compression (video telephony and teleconferencing) [10]. Because of this a number of applications use it for image compression, speech processing, feature extraction etc[11].

The main steps in image compression are transformation, quantization and encoding. Quantization and encoding processes play a significant role in reducing the size of an image.

Even though several compression algorithms are developed based on 2D DCT or a combination of 2D DCT with other transformations (Hybrid techniques), similar attention has not been paid for compressing an image using 3D DCT. The extension of 2D DCT to 3D DCT has some implicit advantages like high compression rates for homogeneous video sequences and symmetric codec structure [12]. In 1976, J. A. Roeseet al. used the three dimensional DCT for the first time, into inter frame transform image coding. The complexity of computation restricted 3D DCT for theoretical analysis only [11]. It can be found from the recent literature that many new algorithms have been developed for the fast calculation of 3D DCT to compress motion picture (video) [15-19]. This has given rise to many 3D sequential applications based on the 3D-DCT.

In this paper a new technique to compress a still image using 3D DCT has been presented. The performance of the new algorithm has been compared to the JPEG compression scheme based on 2D DCT in our previous work [20].In the present work the experimental results for a sample image to demonstrate the effects of various block sizes on the quality of the reconstructed image are presented and analyzed.

2. New Algorithm To Compress An Image Using 3d Dct

The proposed image compression algorithm uses 3D DCT for compressing a 2D image. The block diagram is shown in Fig.1. An input image of size M×N is first partitioned into N sub images of equal size. Each of these sub images should have the

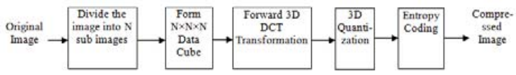
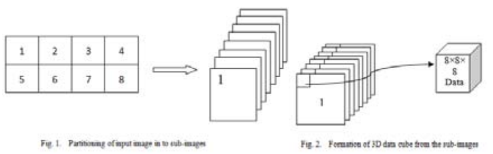


Fig.1 Compression process based on 3D-DCT



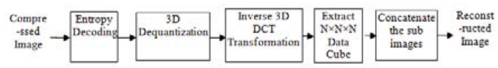


Fig. 3 Decompression process based on 3D-DCT

number of pixels that are multiples of N along the rows and columns. To achieve this condition the image is initially appended with zeros at the ends. Then the 3D data is formed by arranging the N sub images in a sequential order similar to a video sequence as shown in Fig.2. Then N blocks of size N×N are picked from each sub image to form a 3D data cube of size N× N ×N as shown in Fig.3.

These 3D data cubes are transformed using 3D-DCT. The transformed data cube is quantized, scanned in a 2-D zig-zag pattern and entropy coded to get the compressed image. In decompression, all the processes are inverted and applied in reverse order to get the reconstructed image as shown in Fig.4.

A 3D Discrete Cosine Transform

The forward 3D DCT of an N×N×N image block f(x,y,z) is evaluated from (1) given below-

$$C(u,v,w)=\alpha_1(u)\alpha_2(v)\,\alpha_3(w)\sum_{x=0}^{N1-1}\sum_{y=0}^{N2-1}\sum_{z=0}^{N3-1} f(x,y,z)\cos\left[\frac{\pi(2x+1)u}{2N_1}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]\cdot\cos\left[\frac{\pi(2z+1)w}{2N}\right],$$

0≤u≤N-1, 0≤v≤N-1, 0≤w≤N-1, $\alpha(0)=\sqrt{\frac{1}{N}}$ and $\alpha(i)=\sqrt{\frac{2}{N}}$, $i = 1,2,..N-1$ (1)

The (u,v,w) entries of the transformed image are calculated from the elements of the input image matrix.

The Inverse Transform 3D-IDCT is calculated from (2) –

$$f(x,y,z)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{w=0}^{N-1}\alpha(u)\alpha(v)\alpha(w)C(u,v,w)\cdot\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cdot\cos\left[\frac{\pi(2y+1)v}{2N}\right].\cos\left[\frac{\pi(2z+1)w}{2N}\right]$$

0≤x≤N-1, 0≤y≤N-1, 0≤z≤N-1 and $\alpha(0)=\sqrt{\frac{1}{N}}$,

$\alpha(i)=\sqrt{\frac{2}{N}}$ $i = 1,2,..N-1$. (2)

The literature on the distribution of coefficients of 3D-DCT and their dynamic range for the compression of sequential images from video based on 3D DCT show that the DC coefficients range from several hundred to several thousands, whereas the magnitude of AC coefficients is much smaller and most of them are close to zero. Only in the case of busy scenes larger AC coefficients result. Thus the energy of AC coefficients is very less for the entire 3D-DCT cube compared to the DC coefficient [12-14].

B Quantization

This is a major process in achieving the compression because it involves the division of the transform coefficients by a constant and this reduces their magnitude. The smaller coefficients require lesser bits for their representation. Since the coefficients with very small magnitudes are rounded off to zeros after quantization, these cannot be restored by any technique; this step introduces some loss of information in the reconstructed image. The selection of quantization matrix plays a major role in deciding the quality and the compression rate to be attained. Thus it is very important to select a proper quantization matrix for any lossy compression scheme using transforms.

For a 2D DCT based image compression, a standard quantization table has been developed in JPEG compression and other related standards, but for 3D DCT based compression schemes, such standard table is not available. The 3D DCT when applied on a data cube gives a volume of coefficients. Thus the standard tables developed for 2D DCT based compression schemes may not be appropriate to use in the compression schemes using 3D DCT. Hence it is necessary to use a general quantization volume for 3D DCT compression. The quantization matrix design has to be based on the parameters like- The distribution of AC and DC coefficients resulting from subjective and qualitative experiments for some standard test data cube and the matrix elements must have small values at low frequencies and large values at high frequencies. Raymond K.W,Chan and M.C.Lee[12] have introduced a general formula to obtain optimum 3D quantization matrix for the 3D DCT given by (3). As in our work a still image is to be compressed using the 3D DCT, the same function for quantization has been selected.

$$Q(u,v,w)=\begin{cases}A_i\left(1-\frac{e^{-\beta_i(uvw)}}{e^{-\beta_i}}\right) \text{ for } f(u,v,w)\le C\\ A_0\left(1-e^{-\beta_0(uvw)}\right) \text{ for } f(u,v,w)>C\end{cases}\quad(3)$$

Each transformed coefficient is divided by the corresponding element of the quantization matrix. The result of this division is rounded off to complete the quantization step as given by (4).

$$D(u,v,w)=\text{round}\left[\frac{C(u,v,w)}{Q(u,v,w)}\right]\ (4)$$

Here D(u,v,w) denote the quantized coefficients, Q(u,v,w) represent the elements of quantization matrix and C(u,v,w) represent the 3D DCT coefficients obtained after applying 3D DCT.

C Entropy Coding

The DC and AC coefficients are separated and then entropy coded. Entropy coding technique is a lossless method of compressing the data. That is the recovered data is identical to the original. In this technique the symbols with greater probability of occurrence are assigned with shorter code words and the symbols that occur with less probability are assigned longer code words. The example for entropy coding is the Huffman's encoding procedure.The scanning order should be in accordance with the order of quantization value from small to large and then Run Length Encoded. Hence a "zig-zag" scanning order is incorporated. This is also called diagonal scanning.

D Decompression

Image cube is reconstructed by first decoding the Run Length coded sequence and restoring it in a matrix form. This matrix is then dequantized by multiplying each value of D by the corresponding value from the quantization matrix Q as given by (5).

$$R(u, v, w) = Q(u, v, w) \text{ X } D(u, v, w)$$

The Inverse Discrete Cosine Transform (3D IDCT) is then applied to matrix R. The 3D IDCT transform convert the data from frequency domain to spatial domain. These steps are repeated on each data cube to complete the image compression and decompression.

3. Experimental Results And Discussions

The new algorithm developed has been applied on different images of varied dimensions and content for different data cube sizes. The results for a sample input image of size 160 x 120 are shown in this section. Fig.5. shows the input image considered. Fig.6 shows the partitioning of the input image into sub images for N=4.

Fig. 5 Input Image

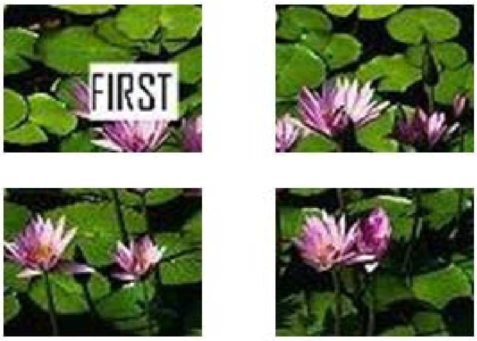


Fig. 6 Partitioning of input image into sub images

Fig.7. shows reconstructed sub sub-images after compression

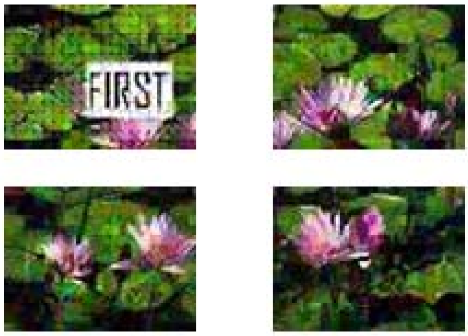


Fig. 7 Reconstructed Sub images

Fig.8 shows the reconstructed image obtained by concatenating the sub images.

Fig. 8 Reconstructed Image

Various quality measures are calculated such as Mean Absolute Deviation (MAD), Mean Square Error (MSE), PSNR, Compression Ratio(CR) from equations (6) to (9)respectively and execution time.

$$MAD = \frac{1}{MNP}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=0}^{p-1}[f'(i, j, k) - f(i, j, k)] \quad (6)$$

$$\mathrm{MSE} = \frac{1}{\mathrm{MNP}}\sum_{\mathrm{i}=0}^{\mathrm{M}-1}\sum_{\mathrm{j}=0}^{\mathrm{N}-1}\sum_{k=0}^{P-1}[\mathrm{f}'(\mathrm{i},\mathrm{j},\mathrm{k}) - \mathrm{f}(\mathrm{i},\mathrm{j},\mathrm{k})]^2 \quad (7)$$

In equations "(6)" and "(7)", $f'(i,j,k)$ is the reconstructed image, $f(i,j,k)$ is the original input image; M and N are the number of pixels along rows and columns of the image respectively. P is number of frames (Here P=3 as there are 3 planes R, G and B) respectively.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255\times255}{\mathrm{MSE}}\right)\ \mathrm{dB} \quad (8)$$

$$\text{Compression Ratio (CR)} = \frac{\textit{Size of the original image}}{\textit{Size of the compressed image}} \quad (9)$$

Table.1.lists various quality measures calculated for the different block sizes.

Table.1 Quality measures for different block sizes

Lily	MAD	MSE	PSNR	CR	Execution time in seconds
N=4	10.6012	71.8864	29.5643	1.87128	2.937857
N=8	10.6111	74.0051	29.4382	2.54349	3.147991
N=16	11.6878	77.1145	29.2594	4.49648	13.020991

Fig. 9 Shows the graph of MAD for different block sizes.

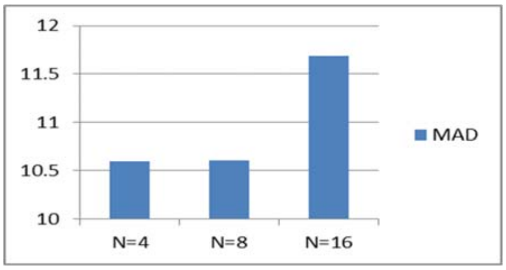


Fig. 9 Graph of MAD versus block size

From the graph it can be observed that Mean Absolute Deviation which is a measure of distortion in the reconstructed image increases with increase in the block size.

Fig. 10 shows the graph of MSE for different block sizes.

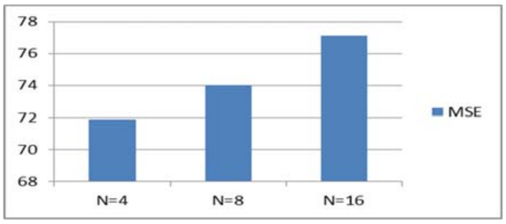


Fig. 10 Graph of MSE versus block size

From the graph, it is observed that if the block size is increased then the MSE also increases. That is quality of the reconstructed image reduces as the block size increases.

Fig.11 shows the graph of PSNR for different block sizes.

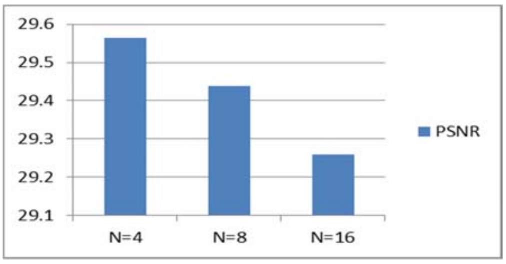


Fig. 11 Graph of PSNR versus block size

From the graph it is clear that the PSNR value reduces with increase in the block size. The result is that the quality of the reconstructed image has reduced but not significantly.

Fig.12 shows the graph of CR for different block sizes.

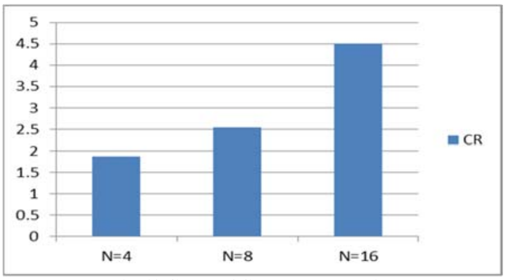


Fig. 12 Graph of CR versus block size

It is clear from the graph that the compression ratio for N=16 is more than double as compared to N=4.

Fig.13 shows the graph of elapsed time versus block size.

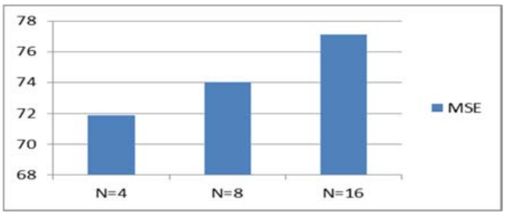


Fig. 13 Graph of Elapsed Time versus block size

The time of execution increases with the increase in block size drastically. This is due to increase in the number of computations for large N values.

4. Conclusions

In general, 2D DCT is used for compressing a still image and 3D DCT to compress a sequence of images. In this research work, an image is compressed using 3D DCT. In this technique, the image is first divided into N sub images. The data cubes of size N×N×N is formed by selecting N× N blocks from the N sub images. Each data cube is then 3D DCT transformed, quantized and encoded. In the decompression process after 3D IDCT, the reconstructed sub images are concatenated to get the full size original input image. This algorithm is tested for different block sizes like N=4,8 and 16 and in each case the quality metrics like MAD, MSE, PSNR,CR and elapsed time are computed. It is found that as the data block size increases the MAD and MSE values increase which results in reduced PSNR value. Also the elapsed time increases as the block size increases. This is because, as the data block size increases, the number of computations also increases drastically.

References

[1] Abhishek Kaushik, Maneesha Gupta, "Analysis of Image Copression Algorithms", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, pp.773-779, Mar-Apr 2012.

[2] Tsai, M. & Hung, H. "DCT and DWT based image watermarking using sub sampling", in the proceeding of 2005 IEEE Fourth International Conference on Machine Learning and Cybernetics, China (pp. 5308–5313).

[3] Al-Haj, A. "Combined DWT-DCT digital image watermarking", Journal of Computer Science, 3(9), 740–746, 2007.

[4] K. R. Rao and P. Yip, "Discrete Cosine Transform; Algorithms, Advantages and Applications", Academic Press Inc., 1990.

[5] G. K.Wallace, "The JPEG still-picture compression standard," Commun.ACM, vol. 34, pp. 30–44, Apr. 1991.

[6] W. B. Pennebaker and J. L. Mitchell, "JPEG Still Image Data Compression Standard", New York: Van Nostrand Reinhold, 1992.

[7] Puri, A."Video coding using the MPEG-1 compression standard". Society for Information Display Digest of Technical Papers 23: 123-126, 1992.

[8] "MPEG-2 video," ITU-T Recommendation H.262-ISO/IEC 13818-2,Jan. 1995.

[9] "Video coding for low bitrate communication," ITU-T Recommendation H.263, Dec. 1995

[10] Andrew B. Watson," Image Compression Using the Discrete Cosine Transform" Mathematica Journal, 4(1), pp.81-88, 1994,

[11] Zhishun Wang, Zhenya He, Fellow, IEEE, Cairong Zou, and J. D. Z. Chen, Senior Member, IEEE "A Generalized Fast Algorithm for n-D Discrete Cosine Transform and Its Application to Motion Picture Coding" IEEE Trans, Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 46, No. 5, pp. 617-627, MAY 1999.

[12] Raymond K.W. Chan and M.C. Lee, "3D-DCT Quantization as a Compression Technique for Video Sequences", Proceedings. International Conference on Virtual Systems and MultiMedia VSMM 97 (Cat. No.97TB100182), 0-8186-8150-0/97 IEEE, pp 188 – 196, 1997.

[13] Tang Haiyan, Sun Wenbang, Gao Bingzhe, Zhang Fengjing," Research on Quantization and Scanning Order for 3-D DCT Video Coding", International Conference on Computer Science and Electronics Engineering, 978-0-7695-4647-6/12, IEEE, pp 200-204, 2012.

[14] Malavika Bhaskaranand and Jerry D. Gibson," DISTRIBUTIONS OF 3D DCT COEFFICIENTS FOR VIDEO", IEEE International Conference on Acoustics, Speech and Signal Processing- ICASSP 2009, 978-1-4244-2354-5/09, IEEE, pp 793-796, 2009.

[15] Said Boussakta and Hamoud O Alshibami,"Fast Algorithm for the 3-D DCT II", IEEE Transactions on Signal Processing, Vol 52,No 4, pp 992-1001, April 2004.

[16] JieLiangandTrac D. Tran," Fast Multiplierless Approximations of the DCT With the Lifting Scheme", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 49, NO. 12, DECEMBER 2001,pp 3032-3044.

[17] Tomáš FRÝZA, Stanislav HANUS, "Algorithms for Fast Computing of the 3D-DCT Transform", ,RADIOENGINEERING, VOL. 12, NO. 1, APRIL 2003, pp 23-26.

[18] Qionghai Dai, Xinjian Chen, and Chuang Lin," Fast Algorithms for Multidimensional DCT-to-DCT Computation Between a Block and Its Associated Sub blocks", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 8, AUGUST 2005,pp 3219-3225.

[19] Geetha K. S. and M. Uttarakumari," Multidimensional Fast Multiplierless DCT Algorithm Using Ramanujan Ordered Numbers", International Journal of Computer and Electrical Engineering, Vol. 4, No. 4, August 2012,pp 511-514.

[20] Gondkar S.R., Attimarad G.V., Chandrasekhar B. (2016) Compression of a 2D Image Using 3D-DCT. In: Shetty N., Prasad N., Nalini N. (eds) Emerging Research in Computing, Information, Communication and Applications. Springer, Singapore.