A Novel Approach To Medical Image Compression Using Sequential 3D DCT

V Naga Prudhvi Raj
Lecturer, Department of ECE
VR Siddhartha Engineering College
Vijayawada
nagaprudhvi@yahoo.com

Dr T Venkateswarlu.
Professor, Department of ECE
Sri Venkateswara College of Engg,
Sri Venkateswara University
Tirupati

Abstract

The output of recently developed imaging devices is more often in a digital format and both storage space and transmission times of these images profit from data compression.

In this paper a novel approach to medical image compression based on three-dimensional Discrete Cosine Transformation (DCT) is proposed. The basic idea is to de-correlate similar pixel blocks through three-dimensional DCT transformation. A number of adjacent pixel blocks are grouped together to form a three-dimensional data cube. Each data cube is 3D DCT transformed, quantized, and Entropy coded (using Variable length coding scheme-Huffman coding scheme). An experimental result says that the new approach is effective specifically for medical images.

1. Introduction.

The Discrete cosine transform is one of the most popular transforms in the field of digital signal processing and communications. It has been widely used in many applications such as speech and image compression. Its use has been expanded to cover three-dimensional applications such as 3-D image and video compression and coding. In these applications, the 3-D image is divided into 8x8x8 or 16x16x16 cubes which are then transformed using 3-D DCT. There have been a number of compression algorithms that employ 3D DCT. These algorithms can be classified into three categories based on the DCT transformation: fixed-size 3D DCT, variable-size 3D DCT, and hybrid 2D/3D DCT. The method adopted in this paper is taken from XYZ video compression algorithm [1] [2]. In the XYZ video compression algorithm [1] [2], a video clip is divided into groups of 8 frames. Each group is further divided into data cubes of 8x8x8 pixels. Each data cube is then independently 3D-DCT transformed, quantized, and entropy encoded.

2. Sequential 3D DCT Image Compression

The block diagram of the sequential 3D DCT image codec is shown in Figure 1.

2.1 3D Data Cube Formation

Starting with the top-left corner of the image, every eight adjacent two-dimensional pixel blocks are taken to construct a three-dimensional data cube. As illustrated in Figure 2, each 3D data cube is formed using a 2x4 adjacent blocks of 8x8 pixels in the following manner. An 8x8 block is considered as a unit. The first data cube is formed by considering the first four blocks in the first row of blocks and numbering them 1, 2, 3 and 4 respectively. Then, similarly considering the first four blocks in the second row and numbering them 5, 6, 7, and 8 respectively.

0-7695-3050-8/07 $25.00 © 2007 IEEE
DOI 10.1109/ICCIMA.2007.115

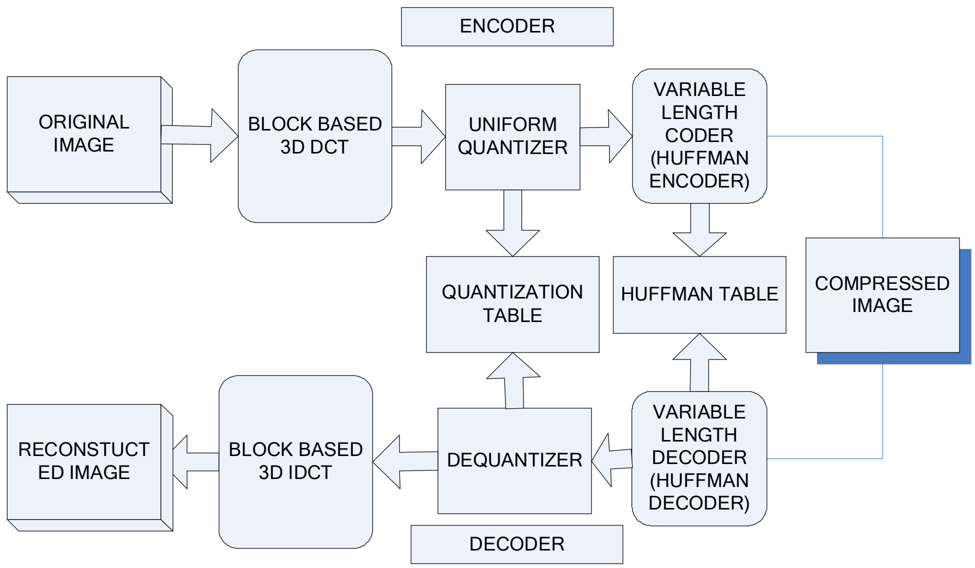

Figure 1: 3D DCT based Image CODEC

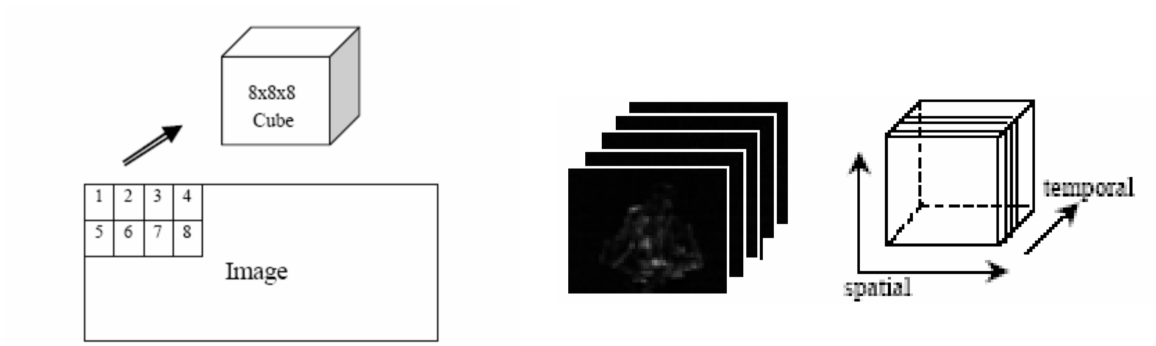

Figure2: 3D Cube Formation.

These eight blocks are then assembled into a three dimensional data cube of 8x8x8 pixels. The order in which a block is placed in the 3D data cube is based on its block number. The second data cube is constructed by taking the next four pixel blocks in the first row of blocks and then the next four pixel blocks in the second row of blocks. Blocks are numbered and ordered in the same way as in the formation of the first data cube. The rest of the image is processed in a similar way.

2.2 3D FDCT Transformation

The intensity values of pixels in each block are normalized to [-128, +127]. Then the three-dimensional forward discrete cosine transformation is performed on each three-dimensional data cube. The purpose is to reduce the redundancy of similar pixel blocks. The 3D DCT is implemented by running the one dimensional transform once in all three axes: namely the X-Y spatial axes and the temporal axis T [2]. The DCT is first run in the X and Y directions for every 8x8 block in the image. Then , the temporal DCT is run on each row in the time axis, for 8 successive blocks which are collected to form a 3D cube. After the 3D DCT transformation, only a small number of low-frequency coefficients are significant. Most high-frequency coefficients are near zeros.

The 3D Forward Discrete Cosine Transformation & 3D IDCT used in the proposed 3D DCT image coder is based on the following formulae:

$$F(u,v,w)=\frac{c(u)c(v)c(w)}{8}\sum_{x=0}^{7}\sum_{y=0}^{7}\sum_{z=0}^{7}f(x,y,z)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\cos\frac{(2z+1)w\pi}{16}$$

$$f(x,y,z)=\frac{1}{8}\left[\sum_{u=0}^{7}\sum_{v=0}^{7}\sum_{w=0}^{7}c(u)c(v)c(w)F(u,v,w)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\cos\frac{(2z+1)w\pi}{16}\right]$$

$$c(i)=\frac{1}{\sqrt{2}}\,for-i=0;c(i)=1\,for-$$

In the above equations *x,y,z* are pixel indices in the time domain, *u,v,w* are coefficient indices in the frequency domain, *f(x,y,z)* denotes a normalized pixel intensity value, and *F(u,v,w)* is a DCT coefficient value.

2.3 Quantization

All DCT coefficients in the same data cube are quantized using the following formula:

$$F_q(u,v,w) = round\left(\frac{F(u,v,w)}{Q(u,v,w)}\right)$$

where *u, v,* and *w* are the spatial indices; *F*(u,v,w) refers to the coefficient value before the quantization; *Q*(u,v,w) denotes the element in the quantization table; F_q (u,v,w) represents the quantized coefficient.

2.4 Zigzag and Entropy Coding

Each 2-D block in a 3D cube is zigzagged into a vector. The difference between the DC values in the adjacent 3D cubes is computed and Huffman encoded. The run-length of zero AC coefficients and the non-zero AC coefficients in each 2-D block in the same data cube are also Huffman encoded.

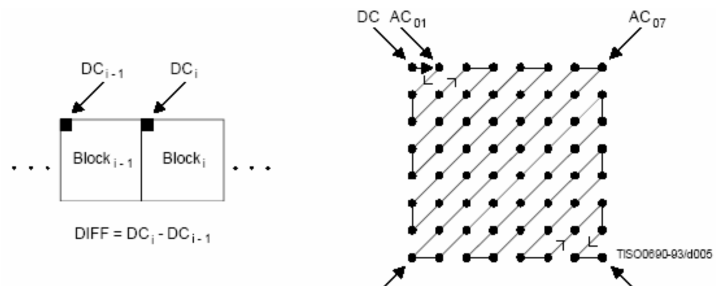

Figure:3 (a) DC encoding (b) Zig-Zag Order

The decoding procedure is the reverse of the encoding procedure. The compressed image is first Huffman decoded. All 2-D blocks of quantized coefficients in the same data cube are identified and dequantized. The 3D inverse DCT is then applied to each data cube. At the last stage, the intensity values are shifted back to [0,255].

3. Experimental Results

The proposed sequential 3D DCT image compression algorithm is benchmarked against the baseline JPEG compression algorithm. Medical images are used for performance evaluation purpose. The compression efficiency is measured using the compression ratio. The quality of an uncompressed image is measured using the Peak Signal to Noise Ratio (PSNR). The PSNR is computed based on the Mean Square Error (MSE).

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right)[dB]; MSE = \frac{1}{MN}\sum_{x=0}^{M}\sum_{y=0}^{N}\left(I(x,y)-\hat{I}(x,y)\right)^2$$

M and N refer to the number of pixels in a row and a column respectively, *I(x,y)* signifies the original intensity value of a pixel at spatial location (x,y), and $\hat{I}(x,y)$ denotes the intensity value of the pixel at the same spatial location in the reconstructed image.

The experimental results for a medical image chest.bmp are shown in Figures 4 and 5. Figure 4(a) shows the original image, while Figure 4(b) is the reconstructed image using the proposed algorithm when the PSNR is 41.65dB and the BPP is 0.67. Figure 4(c) is the reconstructed image using JPEG when the PSNR is 41.47dB and the BPP is 0.69. The graphs showing the PSNR as a function of bits/pixel are shown in Figure 4.

Figure4: (a) Original Image (b) Proposed algorithm (c) JPEG baseline

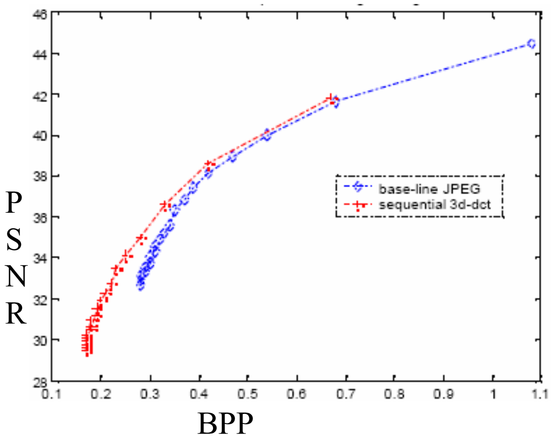

Figure 5: The PSNR as a function of bits/pixels.

Table: 1 Comparison between JPEG & Proposed method.

Compression Method	PSNR (in dB)	BPP
Proposed method	41.65	0.67
JPEG	41.47	0.69

The experimental result says that the proposed approach is slightly better than the JPEG for the specific medical images.

4. Conclusions and Future Work

In this paper, an image compression algorithm that utilizes three-dimensional discrete cosine transformation is proposed. The algorithm first divides an image into 8x8 blocks. Then eight adjacent blocks are taken sequentially and repeatedly to form three dimensional data cubes, which are required by the 3D DCT transformation. Next, a three-dimensional discrete cosine transformation is performed on each data cube. The DCT coefficients in the same 3D data cube are then quantized and Huffman encoded. The experimental results have shown that the new algorithm is better than JPEG. The sequential 3D DCT image coder uses fixed-size data cubes. It does not group all pixel blocks that are similar to each other. If we use the variable-size data cubes to group pixel blocks that are similar to each other, the performance of the coder may be improved better than the sequential 3D DCT image coder.

5. References

[1] B. Furht, “Video Presentation and Compression,” in Handbook of Multimedia Computing, CRC Press, Boca Raton, FL, 1999.

[2] R. Westwater and B. Furht “The XYZ Algorithm for Real-Time Compression of Full-Motion Video,” Journal of Real-Time Imaging, Vol. 2, No. 1, February 1996, pp.19-34.

[3] 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, May 1997, pp. 758-763.

[4] B. Furht, K. Gustafson, H. Hesong, and O. Marques, “An Adaptive Three-Dimensional DCT Compression Based on Motion Analysis,” Proceedings of the ACM Symposium on Applied Computing, 2003.

[5] Xiuqi Li and Borko Furht.: An Approach to Image Compression Using Three-Dimensional DCT. Proceedings of the Visual 2003 Conference. Miami, Florida. (2003).