ICSP2008 Proceedings

MEDICAL IMAGE COMPRESSION USING ADVANCED CODING TECHNIQUE

K.V.Sridhar*, Prof. K.S.R.Krishna Prasad*
*National Institute of Technology, Warangal (AP)-India

Abstract - *Modern medical imaging requires storage of large quantities of digitized clinical data. Due to the constrained bandwidth and storage capacity, however, a medical image must be compressed before transmission and storage. Among the existing compression schemes, Integer based Discrete Cosine transform coding is one of the most effective strategies. Image data in spatial domain is transformed into spectral domain after transformation to attain higher compression gains. Based on the quantization strategy, coefficients of low amplitude in the transformed domain are discarded using a threshold technique: Set Partitioning In Hierarchical Trees (SPIHT) where in only significant coefficients are retained to increase the compression ratio without inducing salient distortion. In this paper, we used two advanced coding engines Context Adaptive Variable Length Coding (CAVLC) and Embedded Block Coding with Optimal Truncation (EBCOT) to code the significant coefficients. Recording or transmitting the significant coefficients instead of the whole coefficients achieves the goal of compression.. Simulations are carried out on different medical images, which include CT skull, angiogram and MR images. Consequent images demonstrate the performance of two coding engines in terms of PSNR & bpp without perceptible alterations. Simulation results showed that the Integer DCT with SPIHT and CAVLC coding has shown better results compared to JPEG & JPEG2000 schemes. Therefore, our proposed method is found to preserve information fidelity while reducing the amount of data*

1. INTRODUCTION

Advanced **M**edical imaging applications require storage of large quantities of digitized clinical data. Due to the constrained requirements of medical data archiving, compression is adapted in most of the storage and transmission applications. There are two categories of compression : Lossy and lossless methods. The choice between the two depends on the system requirements. Lossless compression ensures complete data fidelity after the reconstruction, and yet the compression ratio is limited in general from 2:1 to 3:1. The application of lossy techniques results in information loss to some degree, but it can provide more than 10:1 compression ratio with little perceptible difference between reconstructed and original images. Lossy compression techniques have been widely utilized for image compression applications. Among the existing compression schemes, transform coding is one of the most effective strategy. After the transformation Image data in spatial domain will be transformed into spectral domain to attain higher compression gain. Based on the quantization strategy, coefficients of low amplitude in the transformed domain are discarded and significant coefficients are preserved to increase the CR without inducing salient distortion. Further employing coding technique yields lesser number of bits per pixel.

978-1-4244-2179-4/08/$25.00 ©2008 IEEE

2. DCT

Discrete cosine transform *(DCT)* is one of the most popular transforms [1], [2], [3] for lossy coding of still image or video signals since the international standard JPEG, MPEG and H.26x have adopted the DCT due to its good property of energy distribution in the frequency domain. However, the DCT has not been used for lossless coding since it outputs real numbers in general. The next international standard JPEG-2000 replaced the DCT by the integer wavelet transform (WT) which contains the rounding operations in the lifting structure. As a result, the lossless coding with the transform became possible. The integer DCT, similar to the integer WT, is also attractive since it can be used for not only lossless coding but also lossy coding compatible to the world widely prevailed conventional DCT.

3. INTEGER DCT

The integer DCT [4-9] method of transform has the following features compared to DCT. They are:

1. It is an integer transform (all operations can be carried out using integer arithmetic, without loss of decoding accuracy).
2. It is possible to ensure zero mismatch between encoder and decoder inverse transforms (using integer arithmetic).
3. The core part of the transform can be implemented using only additions and shifts.
4. A scaling multiplication (part of the transform) is integrated into the quantiser, reducing the total number of multiplications.

Thus keeping in view of the advantages integer DCT provides higher CR with preserved quality can be obtained.

The 4x4 integer DCT forward transform [11]
is given by:

$$Y = A \times A^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} [X] \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

Where the coefficients a, b, c are:

$$a = \frac{1}{2}, b = \sqrt{\frac{1}{2}} \cos\frac{\pi}{8}, c = \sqrt{\frac{1}{2}} \cos\frac{3\pi}{8}$$

This matrix multiplication can be factorized further. The basic forward quantizer operation is:

$$Z_{ij} = \text{round}\,(Y_{ij} / Q_{step})$$

where Y_{ij} is a coefficient of the transform, Qstep is a quantizer step size and Z_{ij} is a quantized coefficient. A total of 52 values of Qstep are supported by the standard. The rounding operation here need not round to the nearest integer; for example, biasing the 'round' operation towards smaller integers can give perceptual quality improvements. PF is a^2, ab/2 or $b^2/4$ depending on the position (i, j). In order to simplify the arithmetic, the factor (PF/ Q_{step}) is implemented in the reference model as a multiplication by a factor *MF* and a right-shift, avoiding any division operations:

$$Z_{ij} = round\left(W_{ij}\frac{MF}{2^{qbits}}\right)$$

where $\frac{MF}{2^{qbits}} = \frac{PF}{Q_{step}}$

and qbits=15 + floor (QP / 6)

In integer arithmetic, it can be implemented as follows:

$$|Z_{ij}| = \left(|W_{ij}|.MF + f\right) \quad qbits$$

$$Sign(Z_{ij}) = Sign(W_{ij})$$

where >> indicates a binary shift right

Inverse operation is:

$$Y^1_{ij} = Z_{ij}\, Q_{step}$$

The pre-scaling factor for the inverse transform is incorporated in this operation, together with a constant scaling factor of 64 to avoid rounding errors: $W^1_{ij} = Z_{ij}\, Q_{step} \cdot PF \cdot 64$

W^1_{ij} is a scaled coefficient, which is transformed by the inverse transform: $C_i^T\, WC_i$. The values at the output of the inverse transform are divided by 64 to remove the scaling factor (this can be implemented using only an addition and a right-shift). The parameter V = (Q_{step} · PF · 64) is defined for 0 ≤ QP ≤ 5 and for each coefficient position so that the scaling operation becomes:

$$W^{I}_{ij} = Z_{ij}V_{ij}.2^{floor(QP/6)}$$

4. SPIHT

The SPIHT algorithm [12], [13], [14] adopts a hierarchical quadtree data structure on wavelet-based image. The wavelet coefficient are encoded and transmitted in multiple passes in the SPIHT algorithm. The same holds good even for transmitting the DCT coefficients.

Step 1:

In each pass, only the DCT coefficients that exceed threshold are encoded. The threshold is computed according to the expression

$$T(u) = 2^{p-u}$$

Where u= 0, 1, 2, 3… p denotes the pass number and $p = \lfloor \log \max(c(i,j)) \rfloor$

Where *c (i, j)* is the coefficient at position *(i, j)* in the image.

Step 2:

In this pass the DCT coefficients in the sub-block of the image are compared against the threshold in that corresponding block. The coefficients **out[i][j]** that are greater than the value of threshold (**t**) are retained and the rest are discarded as shown below

if(abs(out[r+i][c+j])< Xt) out[r+i][c+j]=0,

where **r,c** represents the values for the corresponding block and X represents the value of threshold needed. In this way further compression is achieved. The process is repeated for different levels of threshold and the best deal is considered.

5. ENCODING

5.1 ENCODING ALGORITHM WITH COMPLETE DECOMPOSITION

In [15], David Taubman proposed EBCOT (Embedded Block Coding with Optimized Truncation) algorithm, which became the coding scheme kernel of JPEG 2000, the new generation image compression standard. The algorithm supports many wavelet decomposition forms, partitions the subbands into a collection of relatively small code-blocks after wavelet decomposition, and encodes each code-block nearly independently. One of its advantages is that it is independent to decomposition forms. We apply the coefficient rearrangement method in section 5.2 to encode the integer DCT coefficients by EBCOT. The simulation results are shown in section 6.

5.2 REARRANGEMENT OF TRANSFORM COEFFICIENTS FOR EBCOT:

Having transformed an image, we can encode the transform coefficients with wavelet-based coding scheme, such as EBCOT [14]. In order to apply these coding scheme we rearrange the coefficients to take full advantage of these scheme .We adopt a method in which we regard each sample in an 4x4 block as one frequency coefficient, and then collect the coefficients of the same frequency from all the blocks according to the block order. For transform coefficients matrix Tran[Mr][Nc], the matrix of coefficients rearranged became *W* [Mr] [Nc]. There is a relation between them: W[(i%4)M/4 + i/4] [(j%4)N/4 + j/4] = Tran[i][j], where *i*=1,2,...,*M*r, *j*=1,2,...,*Nc*, % denotes modulus operator. This rearranging method corresponds to the complete decomposition of wavelet coefficients

5.3 ENCODING ALGORITHM CAVLC

Context Adaptive Variable Length Coding (CAVLC) [16], the entropy encoder algorithm is used to encode transformed and quantized 4x4 residual blocks to increase the compression efficiency. CAVLC algorithm uses multiple VLC tables for a syntax element. In this scheme, VLC tables for various syntax elements are switched depending on already transmitted syntax elements. Since the VLC tables are designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using a single VLC table.

In the CAVLC entropy coding method, the number of nonzero quantized coefficients (N) and the actual size and position of the coefficients are coded separately. After zig-zag scanning of transform coefficients, their statistical distribution typically shows large values for the low frequency part decreasing to small values later in the scan for the high-frequency part. CAVLC algorithm maps each 4x4 block of transform coefficients in accordance with the zig-zag scan order. Typically, after quantization a block contains only a few significant, i.e., nonzero coefficients, where, in addition, a predominant occurrence of coefficient levels with magnitude equal to 1, so-called trailing 1's (T1), is observed at the end of the scan. It encodes each block in the following five steps.

1. Since the number of non-zero coefficients in neighboring blocks is correlated, CAVLC algorithm generates coeff_token the number of non-zero coefficients (Totalcoeff) and the number of trailing ±1 values (Trailing ones) for a block context adaptively. It uses one of the four different VLC tables for generating the coeff_ token for a block based on the number of non-zero coefficients in the neighboring blocks as follows depending on the parameter nC based on the number of non-zero coefficients in the left-hand and upper previously coded blocks, nA and nB respectively. If upper and left blocks nB and nA are both available (i.e. in the same coded slice), nC = round ((nA + nB) /2). If only the upper is available, nC = nB; if only the left block is avail-able, nC = nA; if neither is available, nC = 0. It, then, selects the VLC table that will be used for generating the coeff_token based on the value of nC as shown in Table 1.

nC	**VLC Table for coeff_token**
0,1	Table 1
2,3	Table 2
4,5,6,7	Table 3
8 or above	Table 4

2. It encodes the sign of each Trailing one with a single bit 0 for +1 and 1 for 1 for -1 in reverse order starting with the highest frequency trailing one.

3. It encodes the level (sign and magnitude) of each remaining non-zero coefficient in the block in reverse order starting with the highest frequency coefficient and working back towards the DC coefficient. The codeword for a level consists of a prefix and a suffix. CAVLC algorithm adapts the suffix length for the level parameter depending on recently coded level magnitudes. It sets the suffix length for the first level, except in some special cases, to 0. It then increments the current suffix length, if the magnitude of the current level is larger than a predefined threshold for this suffix length.

4. It encodes the total number of zeros before the last non-zero coefficient (Total_Zeros) using a VLC table.

5. It encodes the number of zeros preceding each non-zero coefficient (Run_Before). Since after transformation and quantization, blocks typically contain mostly zeros, CAVLC uses run-level coding to represent strings of zeros compactly.

6. EXPERIMENTAL RESULTS

A C-code has been developed and several experiments have been carried with i) different levels of quantization parameter (**Qp**) used in integer DCT and ii) various levels of threshold **(t)** set by SPIHT. The process is carried to CAVLC, which takes care of the final compression (Fig: 1 & 2). Another Advanced coding method EBCOT has also been applied to the integer DCT coefficients and a comparison is brought out with the technique proposed i.e., with Integer DCT based SPIHT with CAVLC coding technique. The comparison has been made between the two coding techniques (Fig: 3) along with the standards JPEG and JPEG 2000.

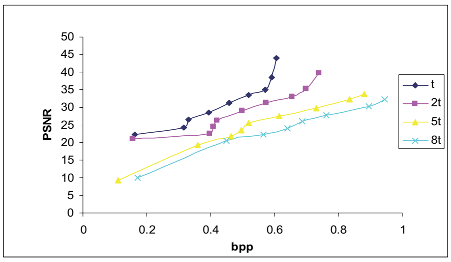


Fig: 1

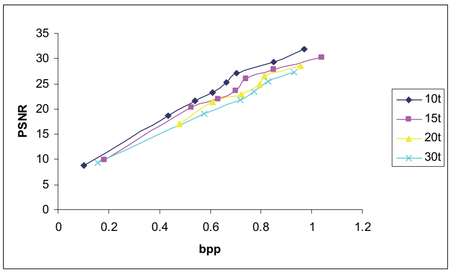


Fig: 2

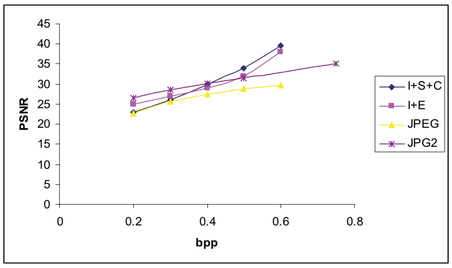


Fig: 3

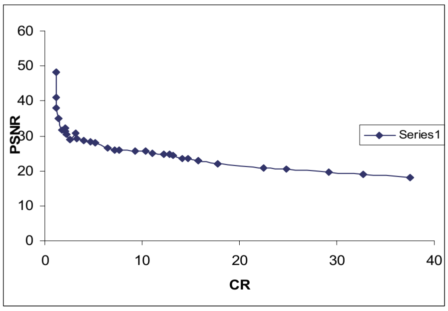
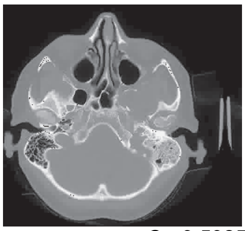


Fig:4

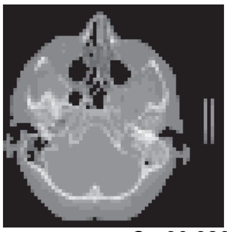
Table :2

Qp	PSNR	CR
0	48.1769	1.165209
10	40.9674	1.165209
20	34.86292	1.461291
25	31.71004	1.74968
30	30.69281	3.150163
35	26.68083	6.450394
40	24.90528	11.033
46	22.93007	15.73871
48	20.77082	22.5055
50	19.61143	29.15303
52	18.01514	37.49199

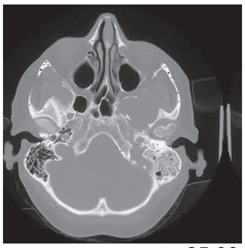
Fig:5



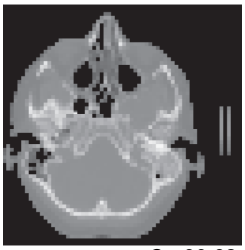
T = 2t, Qp = 20 Cr=9.5925 , PSNR=28.906dB



T = 2t, Qp = 40 Cr=30.826 , PSNR=20.992 dB



T = 10t, Qp = 20 Cr=25.93, PSNR=23.22 dB



T = 40t, Qp = 15, Cr=30.62 PSNR=21.24dB

7. CONCLUSIONS

In this paper we presented an advanced coding technique and performed parameter comparison (Figs:2-5) and in Table:2 with JPEG, wavelet based SPIHT method and integer DCT based EBCOT method. It has been observed that the proposed method gives better results in terms of PSNR and bpp compared to the rest. The coding schemes employed (I+S+C, I+E) out performs JPEG at all bit rates and out performs JPEG2K at moderate bit rates for medical images.

8. REFERENCES

[1] Yung-gi wu, “Medical image compression by sampling DCT coefficients” IEEE Transactions on information technology in biomedicine, vol. 6, no. 1, march 2002

[2] Yusong Yan, Qingyun Shi, “Reversible Integer DCT and lossless image coding”, Software Journal, Vol.11, No.5, pp.620-627, May, 2000

[3] SChokchaitam, MJwahashi, SJitapanakul, “A New Lossless- DCT for Unified Losslesa / Lossy Image Coding”, IEEE MWSCAS 2004, July 2004.

[4] Y. G. Wu and S. C. Tai, “Medical Image Compression by Discrete Cosine Transform Spectral Similarity Strategy”, IEEE Trans. Information Technology in Biomedicine, vol. 5, no. 3, Sep. 2001, pp. 236-243

[5] S.Chokchaitam, MJwahashi, P.Zavarsky, N.Kambayashi, “A Bit-Rate Adaptive Coding System Based on Lossless DCT’, IEICE Trans. on Fundamentals, E85-A no.2, pp.403413, Feb. 2002.

[6] Y.J.Chen, SOraintara, T.Nguyen, INTEGER DISCRETE COSINE TRANSFORM (Int DCI)”, invited paper, the 2nd International Conference on Information, Communications and Signal Processing, Singapore, Dec. 1999.

[7] Pengwei Hao, Qingyun Shi, “The integer implementation of reversible linear transform”. China Science (Serie E), April 2000, No. 2, pp. 132-141

[8] KKomatsu, KSezaki. “2D Lossless Discrete Cosine Transform”, IEEE KIP 2001, pp.466-469.2001.

[9] Wei, Pengwei Hao, Qingyun Shi, “Integer DCT-based Image Coding”, Changjiang National Lab on Machine Perception, Peking University, Beijing, CHINA

[10] Khalid Sayood “Introduction to Data compression” Second Edition, Morgan kaufmann Publishers

[11] Iain E. G.Richardson “H.264 and MPEG-4 Video Compression. Video Coding for Next-generation Multimedia”. The Robert Gordon University, Aberdeen

[12] A. Said and W.A. Pearlman, “A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees”. IEEE Trans. Circuits and Systems for Video Tech. Vol. 6, no. 3, pp. 243-250, June 1996.

[13] Yen-Yu Chen and Shen-Chuan “Embedded Medical Image Compression Using DCT Based Subband Decomposition and Modified SPIHT Data Organization”, National Cheng Kung University and ChengChou Institute of Technology, China

[14] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard, and Ya-Qin Zhang, “A Comparative Study of DCT- and Wavelet-Based Image Coding”. IEEE transactions on circuits and systems for video technology, vol. 9, no. 5, august 1999

[15] David Taubman, “High Performance scalable Image Compression with EBCOT”. IEEE Trans. Image Proc. Vol.9, no 7, July 2000

[16]Yi-Ming Lin and Pei-Yin Chen, “An Efficient Implementation of CAVLC for H.264/AVC”, Proc.of the First Intl Conference On Innovative Computing,Information and Control (ICICIC'06).