Lossy To Lossless Medical Image Coding Using Joint Bit Scanning Method

Tirupathiraju kanumuri (Corresponding author)
Electrical Engineering Department, IIT Roorkee
Roorkee 247667, India
Tel: 91-8057713039 E-mail: ktrajuiitr@gmail.com

M.L. Dewal
Electrical Engineering Department, IIT Roorkee
Roorkee 247667, India
E-mail: mohanfee@iitr.ernet.in

R.S. Anand
Electrical Engineering Department, IIT Roorkee
Roorkee 247667, India
E-mail: anandfee@iitr.ernet.in

Abstract

A new algorithm for progressive medical image coding is presented. On the 8-bit gray scale image, lifting based integer wavelet transform (IWT) are applied to get the three level multi-resolution Integer wavelet transformed image. Then, it is encoded using block based partitioning scheme to exploit the energy clustering in frequency and in space. Whenever a pixel is found significant, pixel value is completely transmitted using vertical bit scanning and then proceeds again with block based coding. Experiments are carried on MRI images to prove the effectiveness of the proposed algorithm. The results shows a significant improvement in terms of distortion measured as peak signal to noise ratio (PSNR) and Correlation Coefficient (CoC) for a given bit rate compared to the existing state of the art embedded image coding methods.

Keywords: progressive image coding, Medical Image Compression, joint bit scanning, telemedicine

1. Introduction

With the advent development of digital imaging and image processing technology, all the hospitals are moving towards digitization of medical images for processing, storage and transmission purposes. This requires higher band width for image transmission. In telemedicine, the medical images are transmitted over Internet over long distances. In remote places such as villages, ships and air planes, the communication channel is very narrow; embedded image coding method that can provide progressive reconstruction is preferred so that the user can stop decoding based on the individual requirements at the decoding end.

Many wavelet transform based image coding methods available in the literature. An embedded image coding method, called embedded zerotree wavelet coding (EZW) is introduced by (Jerome, M, Shapiro., 1993). Due to obvious advantages of embedded coding, many methods are proposed due to its lossy to lossless progressive coding property. A compression technique with the reversible wavelets (CREW) is proposed by (A, Zandi. et al. 1995). A new method known as set partitioning in hierarchical trees (SPIHT) which uses the parent child relationships exploiting the magnitude correlation across bands of decomposition to encode the image is introduced by (Amir, Said., 1996). A new coding method called set partitioning embedded block coder (SPECK) is proposed by (William, A, Pearlman et al. 2004) which exploit the parent child relationships as well as clustering of energy in the frequency domain. It scans the transformed image horizontally bit plane by bit plane. A block based coding using rate distortion optimization called Embedded Block Coding with Optimized Truncation of the embedded bit-streams (EBCOT) is proposed by (David, Taubman., 2000). It provides resolution scalability and is included in JPEG 2000 image compression standard.

In this paper a new method is proposed which combines the advantages of vertical bit scanning employed for coding DCT coefficients in baseline JPEG and embedded coding methods that employs horizontal bit scanning to exploit energy clustering in frequency and in space. The remaining part of paper is organized as follows. Proposed method is discussed in Section 2. Performance evaluation parameters are given in Section 3. The results and discussions are given in Section 4 and finally the conclusions are given in Section 5.

2. Proposed Method

In base line JPEG, a discrete cosine transformed coefficient in each block is completely encoded before encoding the next coefficient in a zig zag scanning order as shown in Fig. 1. Initially the bits of coefficient A1 are encoded from MSB to LSB and then go to B1, A2, A3, B2 and so on as indicated by the arrow lines. The image is transmitted coefficient by coefficient in vertical bit scanning order. In vertical scanning a coefficient is completely refined before going to the next coefficient. In embedded coding adopted by EZW, SPIHT and SPECK, the MSBs of all the coefficients are grouped together to form one layer and encoded initially. The algorithm then moves to the layer of second significant bit plane and so on. Such a bitplane scanning approach can be regarded as horizontal bit scanning. In horizontal bit scanning, most important coefficients can be completely refined only after encoding all the bitplanes and hence not fully efficient. If vertical bit scanning is added to the horizontal bit scanning the significant coefficients can be refined at a faster rate and hence the PSNR will be more for any bit rate.

In the proposed method initially the transformed coefficients are encoded using horizontal bit scanning using block based coding to exploit the energy clustering in frequency and in space. Whenever a coefficient is found significant vertical bit scanning is employed. If a coefficient is found significant its MSB is always 1. So, 1 is encoded to represent that it is significance. The sign bit is encoded and then the binary equivalent of the coefficient except the MSB is encoded. With this the significant coefficients are encoded at a faster rate and we need not wait until all bit planes are encoded. Hence for the transmitted number of bits the reconstructed image will have more PSNR than the other embedded coding methods. For lossless encoding, the number of bits required is same as that of the SPECK algorithm. As this method employs both horizontal and vertical bit scanning it is termed as joint bit scanning method. The implantation procedure of the proposed method is explained below.

The input image is transformed for 3 levels using lifting based (2, 2) integer wavelet transforms (Adams, M.D., and Kossentini, F., 2000). Initially horizontal bit scanning is employed using block based partitioning.

We use two sets. Sets of type ***S*** and ***I***. Initially ***LL3*** in Fig. 2 is considered as ***S***. The entire block is checked for significance. If it is significant, 1 is transmitted and it is divided into 4 sets as shown in fig 3 and each block is processed separately. If any block is insignificant 0 is transmitted and it is added to ***LIS*** and processed in the next iteration with lower threshold. If it is reduced to single coefficient while processing and if it is significant, 1 is transmitted and then its sign is transmitted and its binary equivalent except the MSB is transmitted. With this the significant coefficient which contributes major energy are reconstructed at a faster rate compared to the SPECK algorithm.

Initially entire transformed image except ***LL3*** is considered as ***I*** as shown in Fig. 2. ***I*** is checked for significance. If it is significant, it is divided into three ***S*** and one ***I*** as shown in Fig. 3 and each ***S*** is processed first and then ***I*** is processed.

Decrement the threshold and go for next iteration. In each iteration ***LIS*** is checked for significance starting from smaller size to larger size and then ***I*** is processed. The algorithm of the proposed method is given below.

The significance of a set ***T*** is calculated using

$$\tau_n(T) = \begin{cases} 1, & if\ 2^n \leq \max_{(i,j)\in T}\{|c_{i,j}|\} < 2^{n+1} \\ 0, & else \end{cases} \quad (1)$$

Where

n is the threshold given by

$$n = \left\lfloor \log_2\left(\max_{(i,j)\in T}\{|c_{i,j}|\} \right) \right\rfloor \quad (2)$$

$c_{i,j}$ is the coefficient value at pixel location (i,j)

2.1. Algorithm:

1) Initialization

- Partition image transform ***x*** into two sets: ***S*** = root, and ***I*** = ***x*** - ***S***
- output $n_{\max} = \left\lfloor \log_2\left(\max_{(i,j)\in x}\{|c_{i,j}|\} \right) \right\rfloor$
- Add ***S*** to LIS

2) Sorting Pass

- In increasing order of size $|S|$ of sets

  – for each set ***S*** $\in$ LIS, *ProcessS(S)*
- if ***I*** $\neq \phi$, *ProcessI( )*

3) Quantization Step

- decrement ***n*** by 1, and go to step 2

Procedure ***ProcessS( )***

1) output $\tau_n(S)$

2) if $\tau_n(S)=1$

- if ***S*** is a pixel, then output sign of ***S*** and binary equivalent of ***S*** except MSB.
- else *CodeS(S)*
- if ***S*** $\in$ LIS, then remove ***S*** from LIS

3) else

- if ***S*** $\notin$ LIS, then add ***S*** to LIS

4) return

Procedure ***CodeS( )***

1) Partition ***S*** into four equal subsets *O(S)*

2) for each set $S_i \in 0(S)(i=1,2,3)$

- output $\tau_n(S_i)$
- if $\tau_n(S_i) = 1$

  – if ***S*** is a pixel, then output sign of ***S*** and binary equivalent of ***S*** except MSB.

  – else *CodeS(* S_i *)*

- else

  – add S_i to LIS

3) return

Procedure ***ProcessI( )***

1) output $\tau_n(I)$

2) if $\tau_n(I) = 1$

- *CodeI( )*

3) return

Procedure ***CodeI( )***

1) Partition ***I*** into four sets. Three S_i and one ***I***

2) for each of the three sets S_i $(i=1,2,3)$

- *ProcessS(* S_i *)*

3) *ProcessI( )*

4) return

3. Performance Evaluation Parameters

The metric for compression rate is the bit-rate and is given by

$$\text{Bit} - \text{rate} = \frac{\text{(bits per pixel of original image)}}{\text{compression ratio}} \quad (3)$$

Where,

$$\text{compression ratio} = \frac{\text{original image size in bits}}{\text{encoded image size in bits}} \quad (4)$$

The measurement of ability to reduce the noise is given by the peak signal to noise ratio (PSNR) and is the appropriate parameter to judge the quality of compression. Higher the values of PSNR better the compression quality and vice versa. The PSNR is defined as:

$$\text{PSNR}(\text{dB}) = 10\log\left(\frac{255^2}{\text{MSE}}\right) \text{ in dB} \tag{5}$$

Where,

$$MSE = \frac{1}{N.M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[f(x,y) - f^*(x,y)\right]^2 \tag{6}$$

$f(x,y)$ is the original image data

$f^*(x,y)$ is the reconstructed image data

M and *N* are the matrix dimensions.

The correlation coefficient (CoC) suggests how closely the reconstructed image is correlated with the original image, on a scale of 0-1. The closure the value of CoC to 1, higher the correlation of compressed image to the original image and vice versa. The CoC is defined as:

$$CoC = \frac{\sum_{x=1}^{m}(f(x,y) - \bar{f})\sum_{y=1}^{n}(\hat{f}(x,y) - \bar{\hat{f}})}{\sqrt{\left(\sum_{x=1}^{m}\sum_{y=1}^{n}(f(x,y) - \bar{f})^2\right)\left(\sum_{x=1}^{m}\sum_{y=1}^{n}(\hat{f}(x,y) - \bar{\hat{f}})^2\right)}} \tag{7}$$

Where,

$\bar{f}$ is the mean of original image.

$\bar{\hat{f}}$ is the mean of reconstructed image.

4. Results and Discussions

The performance of the proposed method is evaluated on 512×512 size Brain MRI images. The proposed method is compared with the state of the art embedded image coders SPIHT, SPECK and EBCOT. All methods are implemented using (2, 2) lifting based integer wavelet transforms. The comparative results of the PSNR for a given bit-rate are shown in Table's 1 and the corresponding plots are shown in Fig. 5. The comparative results of the CoC for a given bit-rate are shown in Table's 2 and the corresponding plots are shown in Fig's. 6. Figs. 7, 8, 9 and 10 shows the reconstructed images in each iteration using EBCOT, SPIHT, SPECK and Proposed methods respectively. In each figure it can be seen that the first image shows original image and second image onwards shows the output after each loop. It can be seen that the image clarity goes on increasing from loop to loop and doctor can stop at any instant.

From the plot of Fig. 5 it can be seen that for a bit-rate of 0.03 all methods are giving almost similar result. From bit-rates of 0.05 to 0.2 the proposed method is giving better construction quality than other methods. The proposed method and SPECK are giving loss less image at a bit-rate of 0.25. SPIHT is giving lossless coding at a bit-rate of 0.3 and EBCOT at 0.36. From the results it is proved that the proposed method is best suited for progressive coding of medical images. From the plot of Fig. 6 it can be seen that for a bit-rate of

0.03 all methods are giving almost similar CoC. For all other bit-rates the proposed method is having better correlation than the other methods.

5. Conclusions

New block based image coding method with joint bit scanning is proposed which exploits energy clustering in frequency and in space. The effectiveness of the proposed method is performed on Brain MRI images of size 512×512 . The results are compared with the state of the art embedded image coding methods and found that the proposed method is giving better results than the existing methods for all bit-rates on MRI images.

References

Adams, M.D., and Kossentini, F., (2000). Reversible integer-to-integer wavelet transform for image compression: performance evaluation and analysis. IEEE Trans. Image Processing, 8 (6), 1010-1024.

Amir, Said., (1996) A New fast and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits and Systems for Video Technology. 6 (3), 243 – 250.

A, Zandi., J.D, Allen., E.L, Schwartz., M, Boliek., (1995). CREW: compression with reversible embedded wavelets. IEEE Data Comp. Conf. 212-221.

David, Taubman., (2000) High Performance Scalable Image Compression with EBCOT. IEEE Trans. Image Processing, 9 (7), 1159-1170.

Jerome, M, Shapiro., (1993). Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Image Processing. 41 (12), 3445 – 3462.

William, A, Pearlman., Asad, Islam., Nithin, Nagaraj., and Amir, Said., (2004) Efficient, Low-Complexity Image Coding With a Set-Partitioning Embedded Block Coder. IEEE Trans. Circuits and Systems for Video Technology, 14 (11), 1219-1235.

Table 1. Comparative results for bit-rate vs. PSNR

Bit-rate	**PSNR**			
	Proposed	SPECK	SPIHT	EBCOT
0.03	27	27	27	26
0.05	32	30	30	29
0.08	35	34	31	30
0.1	36	35	35	31
0.15	42	42	40	34
0.2	50	46	45	40
0.25	Inf	Inf	55	46
0.3			Inf	58
0.36				Inf

Table 2. Comparative results for CoC vs. PSNR

Bit-rate	**CoC**			
	Proposed	SPECK	SPIHT	EBCOT
0.03	0.56	0.56	0.56	0.55
0.05	0.976	0.975	0.975	0.916
0.08	0.993	0.996	0.992	0.975
0.1	0.994	0.997	0.997	0.992
0.15	0.999	0.99	0.998	0.996
0.2	0.999	0.999	0.999	0.998
0.25	1	1	0.999	0.999
0.3			1	0.999
0.36				1

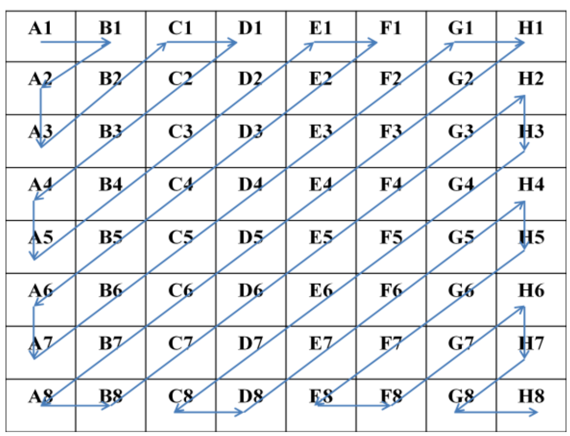

Figure 1. zig zag scanning order in baseline JPEG.

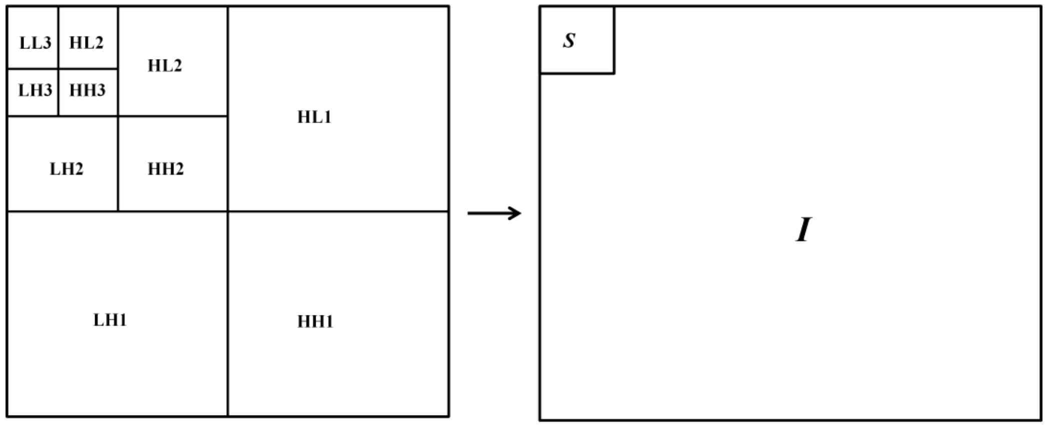

Figure 2. Partitioning of transformed image into sets of type *S* and *I*

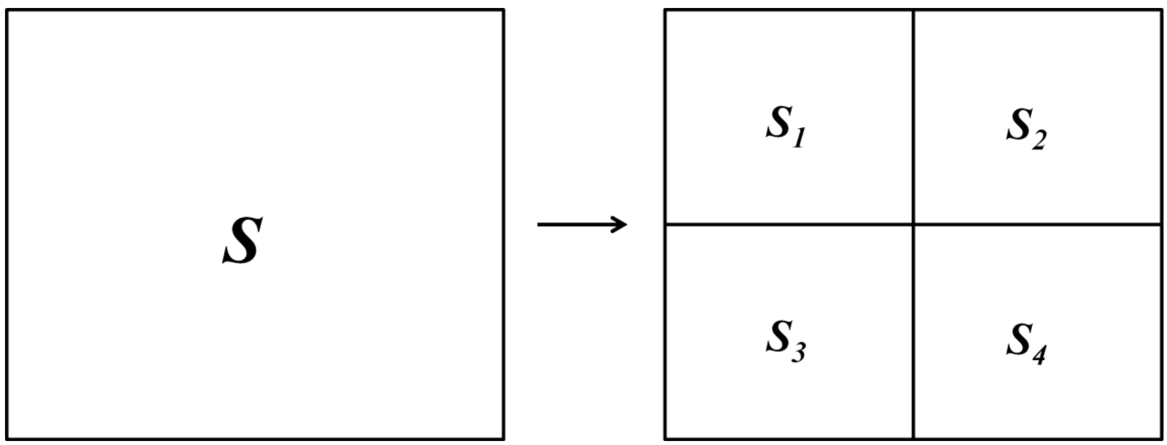

Figure 3. Partitioning of *S*

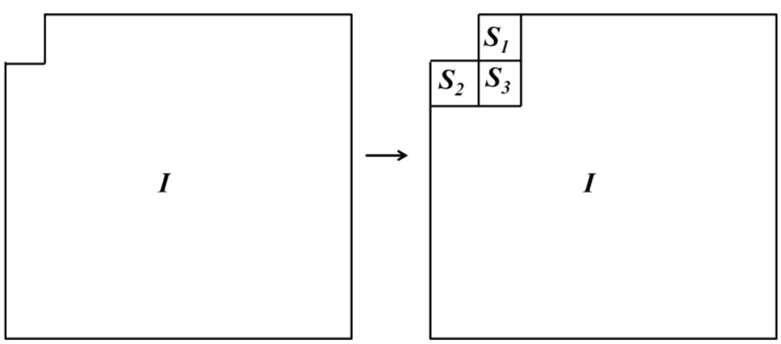

Figure 4. Partitioning of *S*

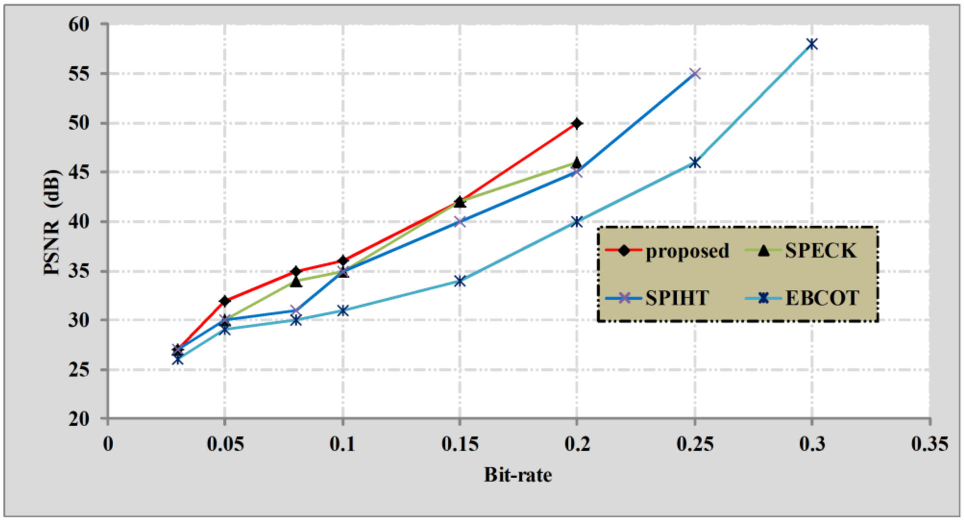

Figure 5. comparative result for bit-rate vs PSNR

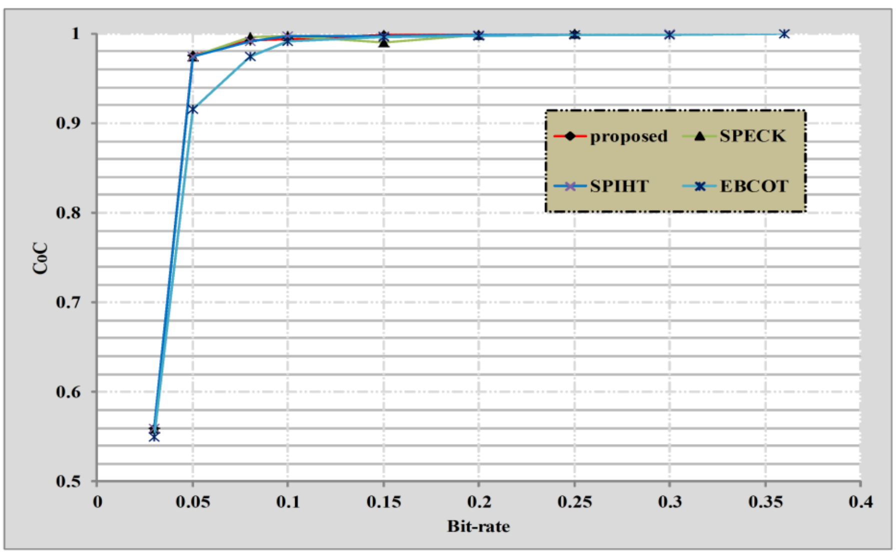

Figure 6. comparative result for CoC vs PSNR