Wavelet Based Dual Encoding Lossless Medical Image Compression

*V. Manohar, Asst. Professor, SR Engineering College, Warangal, TG, India, manoharvu@gmail.com
**Dr. G. Laxminarayana, Professor, Holy Mary Institute of Technology, Hyderabad, TG, India, gln9855@gmail.com

ABSTRACT

Medical image compression using dual encoding scheme which helps in lossless image compression and wavelet helps in increasing the sparsity of the image. Dual encoding scheme comprised with Run length Coding and 8X8 Band encoding schemes, these will increase the affinity of the image and reduces the redundancy of pixels in image. To authorize this novel encoding scheme Huffman table created for calculating average length and entropy of the image by Ternary Huffman code or non-binary Huffman code. Finally PSNR, MSE, Block compression, Total compression and Entropy were calculated and tabulated.

KEY WORDS: Medical Image, Compression, run length encoding, Huffman coding, 8X8 block coding.

1. INTRODUCTION

Picture squeezing coding is will store those picture under bit-stream as conservative Likewise conceivable and with show the decoded picture in the screen as accurate as time permits. At the encoder receives the first picture file, the picture document will a chance to be changed over under an arrangement of double data, which will be known as those bit-stream. Those decoder after that receives the encoded bit-stream and decodes it to structure the decoded picture. On the downright information amount of the bit-stream will be short of what those aggregate information amount of the first image. At that point this will be called picture squeezing.

Those layering proportion will be characterized as takes after:

\[ C_r = \frac{n_1}{n_2}, \]

where n1 is those information rate of first picture and n2 is that of the encoded bit-stream.

Entire encoding building design of picture squeezing framework may be indicated will be figure 1. The essential principle and idea of every utilitarian square will a chance to be presented in the Emulating segments.

1.1 Reduce the Correlation between Pixels

The reason is that the correlation between one pixel and its neighbor pixels is very high, or we can say that the values of one pixel and its adjacent pixels are very similar. Once the correlation between the pixels is reduced, we can take advantage of the statistical characteristics and the variable length coding theory to reduce the storage quantity. This is the most important part of the image compression algorithm; there are a lot of relevant processing methods being proposed. The best-known methods are as follows:

- **Predictive Coding**: Predictive Coding such as DPCM (Differential Pulse Code Modulation) is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element.
- **Orthogonal Transform**: Karhunen-Loeve Transform (KLT) and Discrete Cosine Transform (DCT) are the two most well-known orthogonal transforms. The DCT-based image compression standard such as JPEG is a lossy coding method that will result in some loss of details and unrecoverable distortion.
- **Subband Coding**: Subband Coding such as Discrete Wavelet Transform (DWT) is also a lossy coding method. The objective of subband coding is to divide the spectrum of one image into the lowpass and the highpass components. JPEG 2000 is a 2-dimension DWT based image compression standard.

1.2 Quantization

The objective of quantization is to reduce the precision and to achieve higher compression ratio. For instance, the original image uses 8 bits to store one element for every pixel; if we use less bits such as 6 bits to save the information of the image, then the storage quantity will be reduced, and the image can be compressed. The shortcoming of quantization is that it is a lossy operation, which will result into loss of precision and unrecoverable distortion.

1.3 Entropy Coding

The main objective of entropy coding is to achieve less average length of the image. Entropy coding assigns codewords to the corresponding symbols according to the

![Figure 1. The general encoding flow of image compression](image-url)
probability of the symbols. In general, the entropy encoders are used to compress the data by replacing symbols represented by equal-length codes with the codewords whose length is inverse proportional to corresponding probability.

2. PROPOSING SCHEME

2.1 BLOCK DIAGRAM

2.2 WORK FLOW

Step 1: Input a 2-D image ‘A’.
Step 2: Obtaining Pre-processing Stage on input image ‘A’.
Step 3: Apply wavelet decomposition of the image [Here the decomposition is depends on the number of levels for an input this may vary from 1 to 7 Discrete Wavelet Transformations and output is C].
Step 4: Zero DWT is apply by using a threshold output levels.
Step 5: For Row is equal to one down to Length of last level decomposed image.
Step 6: If $C_i < \text{Threshold value}$, then
Step 7: $C_i = 0 \& \text{The Number of zeros will be increased.}$
Step 8: end
Step 9: end
Step 10: Quantization matrix will be calculated and applied on to the image.

$$\text{Quantization(CQ)} = \frac{(-1 + 2^Q)^n(C - DW T_{\min})}{(DW T_{\max} - DW T_{\min})}$$

Where: $Q$ is the Block size

$DW T_{\min}$ is the minimum value of $C$

$DW T_{\max}$ is the maximum value of $C$

Step 11: Apply Run length encoding scheme.
Step 12: Apply Huffman encoding scheme.
Step 13: Run length coding output Length of data is $L$

1. If $L \leq 4$ then
2. If $L = 0$ then
3. $L_c = L_c + 1$
4. End
5. End

Step 14: Finally the image values were stored with *HDWT scheme.

2.3 WAVELET 2D TRANSFORM

In subband coding, the spectrum of the input is decomposed into a set of band limited components, which is called sub bands. Ideally, the sub bands can be assembled back to reconstruct the original spectrum without any error. Figure 3 shows the block diagram of two-band filter bank and the decomposed spectrum. At first, the input signal will be filtered into low pass and high pass components through analysis filters. After filtering, the data amount of the low pass and high pass components will become twice that of the original signal; therefore, the low pass and high pass components must be down sampled to reduce the data quantity. At the receiver, the received data must be upsampled to approximate the original signal. Finally, the upsampled signal passes the synthesis filters and is added to form the reconstructed approximation signal.

After subband coding, the amount of data does not reduce in reality. However, the human perception system has different sensitivity to different frequency band. For example, the human eyes are less sensitive to high frequency-band color components, while the human ears is less sensitive to the low-frequency band less than 0.01 Hz and high-frequency band larger than 20 KHz. We can take advantage of such characteristics to reduce the amount of data. Once the less sensitive components are reduced, we can achieve the objective of data compression.
Now back to the discussion on the DWT. In two dimensional wavelet transform, a two-dimensional scaling function, \( \phi(x,y) \), and three two-dimensional wavelet function \( \psi^H(x,y) \), \( \psi^V(x,y) \), and \( \psi^D(x,y) \), are required. Each is the product of a one-dimensional scaling function \( \phi(x) \) and corresponding wavelet function \( \psi(x) \).

\[
\phi(x,y) = \phi(x)\phi(y) \quad \psi^H(x,y) = \psi(x)\phi(y) \\
\psi^V(x,y) = \phi(y)\psi(x) \quad \psi^D(x,y) = \psi(x)\psi(y)
\]

where \( \psi^H \) measures variations along columns (like horizontal edges), \( \psi^V \) responds to variations along rows (like vertical edges), and \( \psi^D \) corresponds to variations along diagonals.

Similar to the one-dimensional discrete wavelet transform, the two-dimensional DWT can be implemented using digital filters and samplers. With separable two-dimensional scaling and wavelet functions, we simply take the one-dimensional DWT of the rows of \( f(x,y) \), followed by the one-dimensional DWT of the resulting columns. Figure 4 shows the block diagram of two-dimensional DWT.

**2.3.1 2D-WAVELET MALICIOUS TREE DECOMPOSITION**

The following sub-band decomposition of an image can be given as

\[
X(m,n) = X_0(m) \otimes X_1(n)
\]

(3)

Where \( X(m,n) \) is two-dimensional image and is sub-band decomposition is observed in the equation, subscripts \( j, k \) are stands for temporal direction. From the above equation we can find 8 sub-bands.
Figure 5. 2D Wavelet Decomposition

2.4 QUANTIZATION

To reduce the number of bits needed to represent the transform coefficients, the coefficient $a_{b(u,v)}$ of subband $b$ is quantized to value $q_{b(u,v)}$ using (4)

$$q_{b(u,v)} = (-1 + 2^R) * \left( \frac{|a_{b(u,v)}|}{\Delta_{b(u,v)}} \right)$$

(4)

where the quantization step size is

$$\Delta_{b(u,v)} = 2^{R_b} \left( 1 + \frac{\mu_b}{2^{11}} \right)$$

(5)

$R_b$ is the nominal dynamic range of subband $b$, and $e_b$ and $\mu_b$ are the number of bits allotted to the exponent and mantissa of the subband’s coefficients, respectively.

The nominal dynamic range of subband $b$ is the sum of the number of bits used to represent the original image and the analysis gain bits for subband $b$.

Quantization operation is defined by the step size $\Delta_b$, the selection of the step size is quite flexible, but there are a few restrictions imposed by the JPEG 2000 standard.

1. **Reversible wavelets**: when reversible wavelets are utilized in JPEG 2000, uniform dead zone scalar quantization with a step size of $\Delta_b = 1$ must be used.

2. **Irreversible wavelets**: when irreversible wavelets are utilized in JPEG 2000, the step size selection is restricted only by the signalling syntax itself. The step size is specified in terms of an exponent $e_b$, $0 \leq e_b < 2^5$, and a mantissa $\mu_b$, $0 \leq \mu_b < 2^{11}$.

2.5 RUN LENGTH ENCODING

Run length encoding (RLE) is a simple technique to compress digital data by representing successive runs of the same value in the data as the value followed by the count, rather than the original run of values. The goal is to reduce the amount of data needed to be stored or transmitted. The need to use run length encoding often arises in various applications in DSP, especially image processing and compression.

Example of RLE:

Original:

| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

RLE Representation:

| 1 | 3 | 0 | 2 | 1 | 3 |

As we can see from the above simple example, RLE works the best when applied to data where there are successive runs of the same values. Although one might think that such situations are trivial and not the norm, they actually appear over and over in many fields of DSP.

As an example, when applying a low-frequency digital filter against a random input of varying frequencies, all the frequencies above the cut-off frequency will be represented as 0. Thus in the corresponding output, there would be runs of 0’s which could be better represented as the value (0) followed by the count of consecutive 0’s.

Another real world application of RLE is in image processing. During image compression, higher spatial “frequencies” are filtered out.

In the best case, RLE can reduce data to merely two numbers if all the values in the original data are exactly the same, regardless of the size of the input. However, in the worst case, that is if there are no repeating values in the data, RLE could actually double the amount of numbers compared with the original data. Thus RLE should only be used in cases where runs of the same value are expected.

Another advantage of RLE is that it is a lossless (or reversible) compression technique. That is unlike some other compression techniques, such as JPEG, one can obtain exactly the original data. This is done through an RLE decoder which we have also implemented in both Mat lab and assembly code.

![Figure 6: Current coding bits and their neighbours for RLC operation](image)

2.6 HUFFMANN CODING

The idea behind Huffman coding is to find a way to compress the storage of data using variable length codes. Our standard model of storing data uses fixed length codes. For example, each character in a text file is stored using 8 bits. There are certain advantages to this system. When reading a file, we know to ALWAYS read 8 bits at a time to read a single character.

As we can see from the above simple example, RLE works the best when applied to data where there are successive runs of the same values. Although one might think that such situations are trivial and not the norm, they actually appear over and over in many fields of DSP.

As an example, when applying a low-frequency digital filter against a random input of varying frequencies, all the frequencies above the cut-off frequency will be represented as 0. Thus in the corresponding output, there would be runs of 0’s which could be better represented as the value (0) followed by the count of consecutive 0’s.

Another real world application of RLE is in image processing. During image compression, higher spatial “frequencies” are filtered out.

In the best case, RLE can reduce data to merely two numbers if all the values in the original data are exactly the same, regardless of the size of the input. However, in the worst case, that is if there are no repeating values in the data, RLE could actually double the amount of numbers compared with the original data. Thus RLE should only be used in cases where runs of the same value are expected.

Another advantage of RLE is that it is a lossless (or reversible) compression technique. That is unlike some other compression techniques, such as JPEG, one can obtain exactly the original data. This is done through an RLE decoder which we have also implemented in both Mat lab and assembly code.

![Figure 7: Huffman Encoding table](image)
In order to evaluate the performance of the image compression coding, it is necessary to define a measurement that can estimate the difference between the original image and the decoded image. Two common used measurements are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR), which are defined in (6) respectively. \( f(x,y) \) is the pixel value of the original image, and \( f'(x,y) \) is the pixel value of the decoded image. Most image compression systems are designed to minimize the MSE and maximize the PSNR.

\[
\text{PSNR} = 20 \cdot \log_{10}(\frac{\text{MSE}}{\max(X)})
\]

3. RESULTS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comparison Of Different Wavelets</th>
<th>Haar</th>
<th>Daubechies</th>
<th>Symlets</th>
<th>Coieflets</th>
<th>Biorthogonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Compression</td>
<td></td>
<td>2.340</td>
<td>2.939</td>
<td>2.036</td>
<td>2.830</td>
<td>3.906</td>
</tr>
<tr>
<td>Total Compression</td>
<td></td>
<td>57.282</td>
<td>65.984</td>
<td>50.902</td>
<td>64.671</td>
<td>74.402</td>
</tr>
<tr>
<td>PSNR</td>
<td></td>
<td>27.58</td>
<td>27.0651</td>
<td>27.897</td>
<td>29.2848</td>
<td>32.6796</td>
</tr>
</tbody>
</table>

Figure 8: a) Original Image, Compressed and Reconstructed Images b) Haar, c) Daubechies, d) Symlets, e) Coieflets, f) Biorthogonal.

The above mentioned images in the figure were original image and compressed images of various wavelets these are Haar, Daubechies, Symlets, Coieflets and Biorthogonal wavelets and extracted five metrics from compressed data and
reconstructed compressed image are the parameters of Descriptor Coefficients, Block Compression, Total Compression, Entropy and PSNR.

By using above mentioned five parameters and the results of the simulated images represents Biorthogonal wavelets are much more efficient then compared to all other wavelets.

4. CONCLUSION
A new image compression scheme based on discrete wavelet transform is proposed in this research which provides sufficient high compression ratios with no appreciable degradation of image quality. The effectiveness and robustness of this approach has been justified using a set of real images. From the experimental results it is evident that, the proposed compression technique gives better performance compared to other traditional techniques. Wavelets are better suited to time-limited data and wavelet based compression technique maintains better image quality by reducing errors. The future direction of this research is to implement a compression technique on 3D images.

REFERENCES