ISSN: 0976 – 9102 (ONLINE)
DOI: 10.21917/ijivp.2010.0013

A MODIFIED EMBEDDED ZERO-TREE WAVELET METHOD FOR MEDICAL IMAGE COMPRESSION

T. Celine Therese Jenny[1] and G. MuthuLakshmi[2]
Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tamil Nadu, India
E-mail: tct.jenny@yahoo.com[1] and lakshmi_05@yahoo.com[2]

Abstract
The Embedded Zero-tree Wavelet (EZW) is a lossy compression method that allows for progressive transmission of a compressed image. By exploiting the natural zero-trees found in a wavelet decomposed image, the EZW algorithm is able to encode large portions of insignificant regions of an still image with a minimal number of bits. The upshot of this encoding is an algorithm that is able to achieve relatively high peak signal to noise ratios (PSNR) for high compression levels. The EZW algorithm is to encode large portions of insignificant regions of an image with a minimal number of bits. Vector Quantization (VQ) method can be performed as a post processing step to reduce the coded file size. Vector Quantization (VQ) method can be reduces redundancy of the image data in order to be able to store or transmit data in an efficient form. It is demonstrated by experimental results that the proposed method outperforms several well-known lossless image compression techniques for still images that contain 256 colors or less.

Keywords:
Image Compression, Embedded Zero-Tree Wavelet, PNG, BMP, Vector Quantization, Entropy Coder, Image Compression, Self Organizing Feature Map

1. INTRODUCTION

Image compression is the application of data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to store or transmit data in an efficient form. The purpose of image compression is to represent images with less data in order to save storage costs or transmission time. Without compression, file size is significantly larger, usually several megabytes, but with compression it is possible to reduce file size to 10 percent from the original without noticeable loss in quality. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as still imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

VQ is a powerful method for lossy compression of data such as sounds or images, because their vector representations often occupy only small fractions of their vector spaces. Illustrate this distribution in the case of a simple representation of a grayscale image in a 2D vector space. The vectors will be composed by taking in pairs the values of adjacent pixels. If the input image has 256 shades of gray, visualize the vector space as the [0,0]-[255,255] square in the plane. Then take the two components of the vectors as XY coordinates and plot a dot for each vector found in the input image.

2. REVIEW OF RELATED WORK

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. The pixels that compose an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

2.1 BMP FORMAT

The BMP file format, sometimes called bitmap or DIB file format (for device-independent bitmap), is an image file format used to store bitmap digital images, especially on Microsoft Windows and OS/2 operating systems.

The bits representing the bitmap pixels may be packed or unpacked (spaced out to byte or word boundaries), depending on the format or device requirements. Depending on the color depth, a pixel in the picture will occupy at least n/8 bytes, where n is the bit depth.

2.2 PNG FORMAT

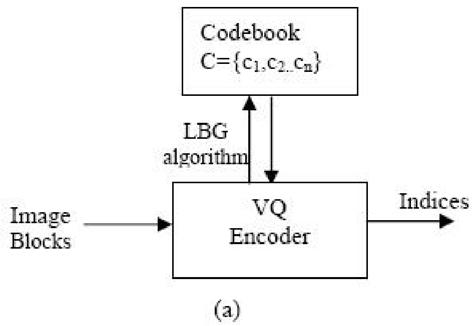
[2] Portable Network Graphics (PNG) is a bitmapped image format that employs lossless data compression. PNG was created to improve upon and replace GIF (Graphics Interchange Format) as an image-file format not requiring a patent license. It is pronounced or spelled out as P-N-G. The PNG acronym is optionally recursive, unofficially standing for “PNG's Not GIF”.

3. PROPOSED STILL IMAGE CODING ALGORITHM

3.1 OUTLINE OF THE PROPOSED WORK

This paper proposes the evaluation of performance of various decomposition schemes for real time application using performance measurement parameters such as Compression Ratio and PSNR rate.

Fig. 1(a) and 1(b), [[1]] shows the encoder and decoder process of the Vector Quantization Technique. The original image is decomposed into the proposed Embedded Zero-Tree Wavelet Decomposition approach.



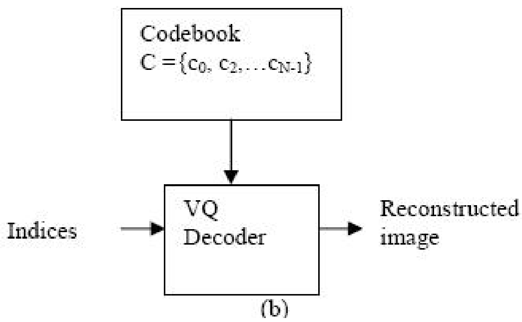


Fig.1(a). VQ Encoder. 1(b). VQ Decoder

3.2 EMBEDDED ZERO-TREE WAVELET DECOMPOSITION [8]

Most regions of a natural image have their energy distributed in the low frequency portion of their spectrums, we expect that if a node of a 4-ary tree is insignificant than its descending leaves are also likely insignificant. This intuition is confirmed the decomposition of a natural image. These resulting insignificant trees are called "zero-trees" since their values relative to a chosen threshold are zero. With the notion of zero-trees, it is possible to efficiently encode a large portion of an image with a very small number of bits. This can be done by encoding the node of a zero tree in a manner that will inform the decoder that the entire tree is insignificant.

The algorithm itself consists of an outer iterative loop that has two inner passes inside it: the significance pass and the refinement pass. The significance pass uses the concept of zero-trees to efficiently encode the spatial location of significant portions of the image, while the refinement pass uses embedded encoding to create an approximation of the significant values. Because the encoding process is progressive, the EZW encoder may be terminated at any point in the algorithm and an exact output bit budget may be met exactly.

3.3 EMBEDDED ENCODING

Embedded encoding is a method of progressively transmitting a binary stream of bits that approximate a desired number. The process begins by transmitting a most significant bit that represents a very rough estimate of a value. Each successive bit provides a more accurate estimate. Eventually the bit stream converges to a complete representation of the number. This process very much mirrors the method of converting a decimal number to its binary equivalent; however, the embedded encoding process does not require that each bit represent a power of two value. The power of the embedded encoding process comes from the encoder's ability to terminate encoding at any point without the already constructed bit stream losing its relative meaning. Although terminating the encoding process before the stream completely characterizes the desired number will result in quantization error, the transmitted bits will still provide the decoder with a rough estimate.

3.4 THE EZW ALGORITHM

Now having established the concepts of zero-trees and embedded encoding, we will move on to describe the EZW algorithm. The EZW algorithm consists of an outer iterative loop that encompasses two passes: the significance pass and the refinement pass (see Figure II). With each pass the EZW algorithm refines an approximation of the image until a pre-established output bit budget is met. To do this, an initial threshold is chosen by the encoder. This threshold determines which values in the image are significant. These values are added to a significance table and are increasingly refined by the refinement pass. With each iteration the EZW decreases the threshold by a factor of two. Thus, as the algorithm progresses, more values in the decomposed image become significant.

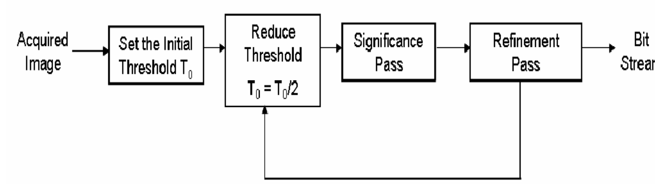


Fig.2. Block diagram of EZW algorithm

3.5 SIGNIFICANCE PASS

The significance pass is the portion of the EZW encoder that determines which portions of the decomposed image are significant. The significance pass uses the inherent zero-trees in the decomposition to efficiently code large portions of the image with only a couple of bits. Since the decoder knows where all the leaves of a corresponding zero-tree node are located, these positions can be automatically marked as soon as a zero-tree node is found and their indices will require no additional transmitted data. Thus, the significance pass is able to efficiently communicate the spatial location of the data. The significance pass consists of a scanning process that searches through each sub-band of the decomposed image from lower to higher frequency bands. When a node is found to be insignificant, its corresponding tree is searched to check if it is a zero-tree. If so, the location is transmitted as a node of a zero-tree and the decoder knows to mark all leaves of the zero-tree as insignificant in the reconstruction process. As a result the encoder will not have to take any further action with regard to these locations and their indices will be skipped when reached in the lower frequency bands.

3.6 STEP-WISE IMPLEMENTATION

3.6.1 EZW Decomposition Analysis Stage

The Embedded Zero-tree Wavelet (EZW) is a lossy compression method that allows for progressive transmission of a compressed image. By exploiting the natural zero-trees found in a wavelet decomposed image, the EZW algorithm is able to encode large portions of insignificant regions of an image with a minimal number of bits. The upshot of this encoding is an algorithm that is able to achieve relatively high peak signal to noise ratios (PSNR) for high compression levels.

Input:

I : input image

level : wavelet decomposition level

Lo_D : low-pass decomposition filter

Hi_D : high-pass decomposition filter

Output:

I_W : decomposed image vector

S : corresponding bookkeeping matrix

[C,S]=AVEDEC2(X,N,'wname') returns the wavelet decomposition of the matrix X at level N, using the wavelet named in string 'wname'. Outputs are the decomposition vector C and the corresponding bookkeeping matrix S. N must be a strictly positive integer.

Instead of giving the wavelet name, the filters are,

[C,S]=AVEDEC2(X,N,Lo_D,Hi_D), Lo_D is the decomposition low-pass filter and Hi_D is the decomposition high-pass filter.

The output wavelet 2-D decomposition structure [C,S] contains the wavelet decomposition vector C and the corresponding bookkeeping matrix S. Vector C is organized as:

C = [A(N) | H(N) | V(N) | D(N) | ...

H(N-1) | V(N-1) | D(N-1) | ... | H(1) | V(1) | D(1)]. _________ 1

where A, H, V, D, are row vectors such that:

A = approximation coefficients,

H = hori. detail coefficients,

V = vert. detail coefficients,

D = diag. detail coefficients,

Each vector is the vector column-wise storage of a matrix.

Matrix S is such that:

S(1,:) = size of app. coef.(N)

S(i,:) = size of det. coef.(N-i+2) for i = 2,...,N+1 and S(N+2,:) = size(X).

3.6.2 EZW Decomposition Encoding

In this method first setting the threshold value. It determines where to stop encoding. A lower threshold value gives better image reconstruction quality. A string matrix containing significance data for different passes ('p','n','z','t'), where each row contains data for a different scanning pass. This is called significance map. Refinement defines a string matrix containing refinement data for different passes ('0' or '1'), each row contains data for a different scanning pass.

When calculating Morton scan order, the matrix should be n by n. For position, start counting from 0. The number of bits needed to represent position. Then convert position into binary. In the binary values, odd bits represent row number and even bits represent column number. Then calculating initial threshold value. The current subordinate list containing coefficients that are already detected as significant.

First row is the original coefficient

Second row is current reconstruction value of this coefficient

This list should be in the correct scan order to reduce complexity of decoder (Morton).

In the encoding level the matrix containing 0's and 1's for refinement of the subordinate list containing reconstruction values is updated to the include refinement -> new reconstruction values.

In the Huffman encoding, A significance map defines, a string array containing significance map data ('p','n','z' and 't') and a refinement defines a string array containing refinement data ('0' and '1').

Steps for Huffman encoder,

- Insert significance map using Huffman
- Insert separator
- Insert length of refinement (20 bits)
- Append end of stream
- Append zeroes, when string length should be multiple of 8

3.6.3 EZW Decomposition Decoding

A string matrix containing significance data for different passes ('p','n','z','t'), where each row contains data for a different scanning pass. This is called significance map. Refinement defines a string matrix containing refinement data for different passes ('0' or '1'). The input bit streams are ‘0’ and ‘1’. Then setting the initial threshold value used while encoding.

When calculating Morton scan order, the matrix should be n by n. For position, start counting from 0. The number of bits needed to represent position. Then convert position into binary. In the binary values, odd bits represent row number and even bits represent column number. Then calculating initial threshold value. The current subordinate list containing coefficients that are already detected as significant.

First row is the original coefficient

Second row is current reconstruction value of this coefficient

This list should be in the correct scan order to reduce complexity of decoder (Morton).

The following steps are the EZW decoder function.

Get matrix index for element

Check whether the element should be processed

Determine type of element

‘p’ – Element is significant positive. So the threshold value is divided by 2 and then added to current value.

‘n’ – Element is significant negative. So the threshold value is divided by 2 and then subtracted from current value.

‘z’ – Element is isolated zero.

‘t’ – Element is zero-tree root.

Using the mask function. Mask is the returned matrix to select the relevant coefficients from the wavelet coefficient matrix.

Get the size of refinement data (next 20 bits).

Update the significant map and the refinement data.

If refinement bit is 1, divided the threshold value by 4 and added to current value.

If refinement bit is 0, divided the threshold value by 4 and subtracted from current value.

Finally, we get the bit rate of the encoded image.

Vector Quantization Based Compression

Vector quantization is the procedure of approximating continuous with discrete values; the input values to the quantization procedure are often also discrete, but with a much finer resolution than that of the output values. The goal of quantization usually is to produce a more compact representation of the data while maintaining its usefulness for a certain purpose.

A vector quantizer maps k-dimensional vectors in the vector space Rk into a finite set of vectors Y = {yi: i = 1, 2, ..., N}. Each vector yi is called a code vector or a codeword. and the set of all the codewords is called a codebook. Associated with each codeword, yi, is a nearest neighbor region called Voronoi region, and it is defined by:

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, for\ all\ j \neq i\} \quad (1)$$

The set of Voronoi regions partition the entire space Rk such that:

$$\bigcup_{i=1}^{N} V_i = R^k \quad (2)$$

$$\bigcap_{i=1}^{N} V_i = \phi \quad \text{for all } i \neq j \quad (3)$$

The representative codeword is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2} \quad (4)$$

where xj is the jth component of the input vector, and yij is the jth is component of the codeword yi.

3.6.4 VQ Compression

A vector quantizer is composed of two operations. The first is the encoder, and the second is the decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel (the channel could be computer storage, communications channel, and so on). Fig.3 [[6]] shows a block diagram of the operation of the encoder and decoder.

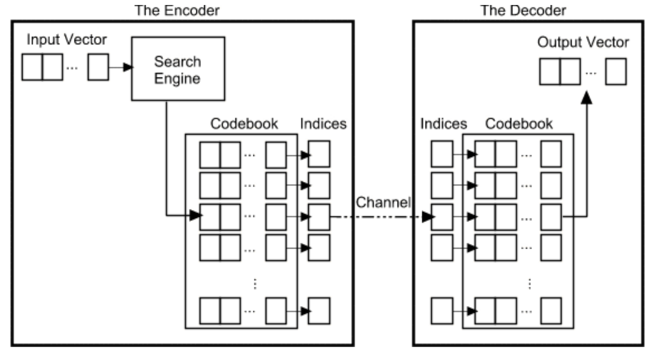


Fig.3. The Encoder and decoder in a vector quantizer

Given an input vector, the closest codeword is found and the index of the codeword is sent through the channel. The decoder receives the index of the codeword, and outputs the codeword.

$$y_i = \frac{1}{m} \sum_{j=1}^{m} x_{ij} \quad (5)$$

where i is the component of each vector (x, y, z, ... directions), m is the number of vectors in the cluster.

4. EXPERIMENTAL RESULT

In the experimental result, we take two different data types of images. The first data type is PNG and second one is BMP. Decomposed these two images using quad-tree decomposition and get the output. Then decomposed output will give the input of vector quantized image compression technique.

4.1 PERFORMANCE ANALYSIS OF COMPRESSION METHODS

There are various tools are available to evaluate the performance of the compression techniques. They are

- Compression Time Taken
- Decompression Time Taken
- Peak Signal to Noise Ratio (PSNR)

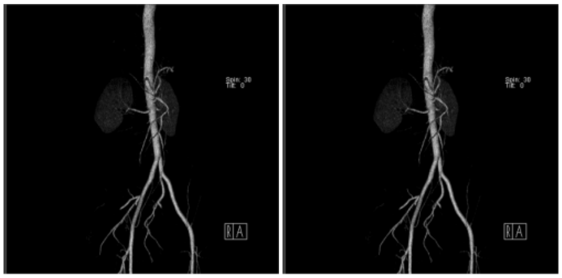
The formula for PSNR is given below,

PSNR=10*log10(MAX(pure_image))^2 / mean square error.

4.2 IMAGES FOR EXPERIMENTAL

The performances of the algorithm are evaluated on .bmp and .png data format images. All the images are of 512x512 pixels. These pictures are the most widely used standard test images used for compression algorithms. The image contains a nice mixture of detail, flat regions, shading, and texture that do a good job of testing various image processing algorithms.

These are still in the industry standard for tests. It is a good test image. These are used to cropping, scanning, resizing, compression or conversion from color to gray-level. These images are used for many image processing researches. In this project, evaluate these images by the compression ratio and PSNR value.



CT_Vascular.png CT_Vascular.bmp

Fig.4. Decoded Image

Table.1. Compression ratio for .png and .bmp image data types

Image	Compression Method	Compression Ratio
CT_Vascular.png	Vector Quantization	1.5345
CT_Vascular.bmp	Vector Quantization	1.5515

4.2.1 Observation

From the above table to achieve the compression ratio for CT_Vascular.png and CT_Vascular.bmp images. The CT_Vascular.png image has the compression ratio is 1.5345 and CT_Vascular.bmp image has the compression ratio is 1.5515.

Table.2. PSNR value for .png and .bmp images

Image	Compression Method	PSNR value
CT_Vascular.png	Vector Quantization	28.9398
CT_Vascular.bmp	Vector Quantization	27.8826

4.2.2 Observation

From the above table achieve the PSNR value for CT_Vascular.png and CT_Vascular.bmp images. The CT_Vascular.png image has the PSNR value is 26.12 and CT_Vascular.bmp image has the PSNR value is 54.14

5. CONCLUSION

The performance of this decomposition has been evaluated in the context of compression ratio and PSNR value. When compared with PNG we achieve better compression ratio and PSNR value.

REFERENCES

[1] Haykin. S, 1994, "Neural Networks: A Comprehensive Foundation", Macmillan Co., New York.

[2] ISO/IEC 15948: 2003 (E), Portable Network Graphics (PNG) Specification (Second Edition) Information technology - Computer graphics and image processing - Portable Network Graphics (PNG): Functional specification.

[3] Netravali, B. Haskel, 1995, "Digital Pictures: Representations, Compression, and Standards (Applications of Communications Theory)" Plenum Pub. Co.

[4] Paeth, A.W., 1991, "Image File Compression Made Easy", in Graphics Gems II, James Arvo, editor. Academic Press, San Diego.

[5] Puzicha. J, Held. M, Ketterer. M, J.M. Buchmann and D.W. Fellner, 2000, "On Spatial Quantization of Color Images", IEEE Transactions on Image Processing, Vol. 9, No. 4, pp.666-682.

[6] Rafael C. Gonzalez, University of Tennessee, Richard E. Woods, MedData Interactive, Steven L. Eddins, 2009, "Digital Image Processing Using Matlab", The MathWorks Gatesmark Publishing.

[7] Taubman. D, 2000, "High performance scalable image compression with EBCOT," IEEE Trans. on Image Processing, Vol. 9, No. 7, pp.1158-1170.

[8] Walker, James S., Nguyen, Truong Q, 2000, "Wavelet-Based Image Compression," Handbook of Transforms and Data Compression, Chap. 7, pp. 267 - 312, CRC Press, Boca Raton.