International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 6, ISSN No. 2455-2143, Pages 72-75
Published Online in IJEAST (http://www.ijeast.com)

RETRIEVE QUALITY AND INTELLIGENCE OF THE ORIGINAL IMAGE USING SET PARTITION IN HIERARCHICAL TREE ALGORITHM

Govinda Rao T
Department of ECE
GMRIT, Rajam, Andhra Pradesh, India

***Abstract*— the main abstract of this paper is to make sure that there is no loss in quality and intelligence of the original image, so that we can retrieve almost exactly the original image from the compressed image. Here Set partition in Hierarchical Tree (SPIHT) Algorithm is used to compress and encode the Image. In this paper Input image is decomposed and encoded using SPIHT algorithm. After this process the output stream is decoded and reconstructed using bi-orthogonal wavelet. The key purpose of image compression is to reduce the size of the images. The advantage of reducing the size of the image is efficient memory usage and reduction in transit time of images through network. We got the results for Lossy image compression using SPIHT algorithm. The results are compared in terms of PSNR and MSE parameters. Finally in this paper Observed that SPIHT algorithm gives high PSNR value than DCT method PSNR value. So resultant image is good in quality by using SPHIT algorithm. Here MATLAB version 2013 used for coding purpose.**

Keywords— **Image Compression, SPIHT Algorithm, DCT, PSNR**

I. INTRODUCTION

Image compression
Image compression is important for many applications that involve huge data storage, transmission and retrieval such as for multimedia, documents, videoconferencing, and medical imaging. Uncompressed images require considerable storage capacity and transmission bandwidth. The objective of image compression technique is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. The task of compression consists of two components, an encoding algorithm that takes a message and generates a "compressed" representation (hopefully with fewer bits), and a decoding algorithm that reconstructs the original message or some approximation of it from the compressed representation. These two components are typically intricately tied together since they both have to understand the shared compressed representation. This results in the reduction of file size and allows more images to be stored in a given amount of disk or memory space. Image compression can be Lossy or lossless.

II. PROPOSED ALGORITHM

A. Process of image compression based on wavelet transforms

The process of image compression based on wavelet transform has three steps: Firstly, original image can receive the transform coefficient matrix through biorthogonal wavelet transform. Because wavelet transform adopts filter of limited length, memory is saved [1]. Then we calculate the wavelet coefficient. Secondly, the transform coefficient matrix is chosen and set quantitative parameters, and adopts symmetrical biorthogonal wavelet with symmetrical periodic continuation. Under the precondition of approving not adding data, avoiding well high frequency noise with boundary process, it has become high efficient and practical continuation method. Thirdly, the quantitative symbol stream can receive efficient and compact image compression bits through SPIHT coding [3]. The process of decoding is similar to the process of coding, but the operation process is opposite. The operations of the suggested image compression scheme are started by applying a suitable color transform on the input colored image, as a second step the wavelet transform will be applied on the luminance and the two chrominance sub-bands. Then, some re sampling processes are performed on the chrominance components. After re sampling stage the transform coefficients are quantized. Some quantized coefficients are selected in adaptive way, this could be done by using improved bit-plane slicing method, and finally an entropy coding method is applied on the output.

B. Set partitioning in Hierarchical Tree

Embedded Zero tree Wavelet (EZW) algorithm has two major strengths. First, the bit stream is embedded and the coefficients are ordered in significance and precision, so that it can be truncated according to the bit-rate requirements of the

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 6, ISSN No. 2455-2143, Pages 72-75
Published Online in IJEAST (http://www.ijeast.com)

channel. Second, it efficiently utilizes the self-similarity between the sub bands of similar orientation and achieves significant data reduction. However, we have pointed out that the EZW algorithm is not exactly optimal and some adjustments of parameters (like, initial threshold) may be necessary to make it optimal with respect to a target bit rate [4]. A single embedded file is unable to give the best performance at all target bit rates. Secondly, EZW has some limitations in efficiently encoding the insignificant coefficients. Although it exploits the self-similarity of coefficients across sub bands of same orientation, it does not make any grouping of insignificant coefficients to improve the coding efficiency. SPIHT algorithm is a modified form of embedded coding of wavelet coefficients that carries the major strengths of EZW, namely ordered coefficient transmission and self-similarity across sub bands of similar orientation. In addition, it partitions the set of coefficients into subsets of insignificant coefficients and identifies each significant coefficient. The approach, proposed by Said and Pearlman is known as Set Partitioning in Hierarchical Trees (SPIHT) [5]. At identical target bit rates, experimentations have shown that SPIHT algorithm has improved performance over EZW, because of its ability to exploit the grouping of insignificant coefficients.

C. Proposed algorithm Encoding and Decoding

The SPIHT algorithm applies the set partitioning rules, as defined above on the sub band coefficients. The algorithm is identical for both encoder and decoder and no explicit transmission of ordering information, as needed in other progressive transmission algorithms for embedded coding, are necessary [6]. This makes the algorithm more coding efficient as compared to its predecessors. Both the encoder and decoder maintain and continuously update the following three lists, viz.

- List of Insignificant Pixels (LIP)
- List of Significant Pixels (LSP)
- List of Insignificant Sets (LIS)

LIP (List of Insignificant Pixels) – It contains individual coefficients that have magnitudes smaller than the thresholds
LIS (List of Insignificant Sets) – It contains set of wavelet coefficients that are defined by tree structures and are found to have magnitudes smaller than the threshold.
LSP (List of Significant Pixels) – It is a list of pixels found to have magnitudes larger than the threshold (significant).
In all lists, each entry is identified by a coordinate. In LIP and LSP, the entry represents individual pixels, whereas in LIS, the entry represents either the set $D(n_1,n_2)$ or the set $L((n_1,n_2)$
As an initialization step, the number (*n*) of magnitude refinement passes that will be necessary is determined from the maximum magnitude of the coefficients. Initially, all pixels are treated as insignificant. The initialization is followed by three major passes – the sorting pass, the magnitude refinement pass and the quantization step update pass which are iteratively repeated in this order till the least significant refinement bits are transmitted. During the sorting pass, the pixels in the LIP, which were insignificant till the previous pass, are tested and those that become significant are moved to the LSP. Similarly, the sets in LIS are examined in order for significance and those which are found to be significant are removed from the list and partitioned. The new subsets with more than one element are added to the LIS and the single pixels are added to LIP or the LSP, depending upon their significance. During the magnitude refinement pass, the pixels in the LSP are encoded for most n^{th} significant bit [7].

D. Proposed Algorithm

Step-1: Initialization

$$n=[\log_2(max_{(n1,n2)}\{|c_{n1,n2}|\})]$$

set the LSP=$\{\emptyset\}$

Set the LIP=$\{(n_1,n_2) \in H\}$ and LIS=$\{D(n_1,n_2), (n_1,n_2) \in H\}$

Step-2: Sorting pass

Step-2.1: For each entry in the LIP, output the significance ("1" if significant, "0" if not significant). If found significant, remove it from the LIP and add to the LSP.

Step-2.2: For each entry in the LIS, output the significance. If found significant, output its sign. Perform the set partitioning using the rule-2 or rule-3, depending upon whether it is the $D(n_1,n_2)$ set or the set $L((n_1,n_2)$. According to the significance, update the LIS, LIP and LSP.

Step-3: Refinement pass

For each entry in the LSP, except those which are added during the sorting pass with the same *n*, output the n^{th} most significant bit.

Step-4: Quantization-step update pass

In this pass, *n* is decremented by 1 and the steps-2, 3 and 4 are repeated until $n = 0$. The decoder steps are exactly identical. Only the output from the encoder will be replaced by the input to the decoder. As with any other coding method, the efficiency of the algorithm can be improved by entropy coding its output, but at the expense of very high coding / decoding time.

III. Experiment and Result

A. Proposed Output for Lossy (.jpg) image using DCT Method

Input image is in .jpg format which is Lossy image. Now this image is converted into grayscale image in matlab. This image is converted into blocks of M*N pixels.

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 6, ISSN No. 2455-2143, Pages 72-75
Published Online in IJEAST (http://www.ijeast.com)


Fig. 1. Input Lossy (.jpg) Image

First size of columns of given image is calculated. Then for each row of coefficients DCT is applied. MATLAB stores data as integers, but DCT and IDCT etc. functions involve floating pt. numbers and thus the integer data values need to be converted into floating point..After IDCT is applied on each block .IDCT function append zeros or truncates the DCT vector length to size of columns before transforming the columns. Now size of rows is found, and now for each column

DCT is applied. MATLAB stores data as integers, but DCT, inverse DCT etc functions involve floating pt. numbers and thus the integer data values need to be converted into floating point..After IDCT is applied on each block. Inverse DCT function append zeros or truncates the DCT vector length to size of columns before transforming the rows. Now original image and reconstructed image are displayed by using matlab commands. The array of compressed blocks that constitutes the image is stored in a drastically reduced amount of space.


Fig. 2. Reconstructed image using DCT

After IDCT is applied on each block .IDCT function append zeros or truncates the DCT vector length to size of columns before transforming the columns. Now size of rows is found, and now for each column DCT is applied. MATLAB stores data as integers, but DCT, inverse DCT etc functions involve floating pt. numbers and thus the integer data values need to be converted into floating point..After IDCT is applied on each block. Inverse DCT function append zeros or truncates the DCT vector length to size of columns before transforming the rows.

Table-1 Output Parameter for DCT

SPIHT	
MSE	**6.55**
PSNR	**39.97dB**

B. Proposed output for lossy (.jpg) image using SPIHT algorithm

Input image is first converted into grayscale image .Then grayscale image is decomposed using bior4.4 filter. Wavelet decomposition level calculated by size of image. It can also be defined by users manually. After we get wavelet decomposed vector and corresponding book keeping matrix .Decomposition is done along horizontal, vertical and diagonally. Then SPIHT algorithm take inputs as image in wavelet domain ,maximum bits required to represent it and wavelet decomposition level, block size .In sorting pass LIP,LIS,LSP are updated based on threshold condition. In Refinement pass sign of significant coefficient sign also taken into account. If sign is positive then it encoded as 1 else encoded as 0. In quantization update pass the value of n is decremented. Then again sorting pass, refinement pass are applied. This process is repeated until n value as 0. The output of SPIHT encoding algorithm is a bit stream. Now for this bit stream SPIHT decoding algorithm is applied. SPIHT decoding algorithm is same as encoding algorithm.


Fig. 3. Reconstructed image using SPIHT

Then SPIHT algorithm take inputs as image in wavelet domain ,maximum bits required to represent it and wavelet decomposition level, block size .The output of SPIHT encoding algorithm is a bit stream. Now for this bit stream SPIHT decoding algorithm is applied. Reconstructed image having good quality.

DCT	
MSE	**145.77**
PSNR	**26.49dB**

Table -2 Output parameters for SPIHT

Compressing an image is significantly different than compressing raw binary data. General purpose compression programs can be used to compress images, but the result is less than optimal. Reconstructed images of camera man having PSNR value of 26.49dB, MSE value as 145.77. The performance of the SPIHT image compression for various wavelet filters is measured and we can conclude that Biorthogonal wavelet i.e, Bior4.4 has a PSNR value as

39.97dB for camera man and we can say it is the best suitable wavelet filter for SPIHT Image compression.

a) The SPIHT algorithm adopts a hierarchical quad-tree.
b) The energy of a wavelet-transformed image is concentrated on the low frequency coefficients.
c) The Performance of this algorithm is verified with respect to the existing algorithms like Discrete Cosine Transform.
d) For this performance analysis we have considered two parameters (Mean Square Error, Peak Signal to Noise Ratio)

IV.CONCLUSION

The proposed Lossy image compression algorithm is simple and effective method for gray-scale image compression and is combined with SPIHT encoding for further compression that saves a lot of bits in the image data transmission. Here mat lab is used for our coding purpose. For input lose less (.bmp) image DCT is applied. Its Mean square error and PSNR values are measured. Again for that image we have applied bi-orthogonal wavelet. Then for wavelet decomposed image SPIHT encoding algorithm is applied. SPIHT encoder output is bit stream. SPIHT decoding algorithm is applied on bit stream and then bi orthogonal reconstruction wavelet is applied. The output is reconstructed image. Now MSE, PSNR values are calculated. In comparison to DCT method for SPIHT algorithm we have obtained less MSE and high PSNR. Similarly for Lossy image (.jpg) extension image first we converted the image into grayscale image then have applied DCT, and SPIHT algorithms. MSE and PSNR values calculated from both these methods are compared. By comparing MSE values we observed that SPIHT algorithm gives 47 percentage less values than those are calculated using DCT method.

But in case of DCT method compressed file occupies less space than that of SPIHT. When image is compressed by using bi orthogonal wavelet and encoding using SPIHT algorithm low MSE and high PSNR is achieved compared to those values of DCT method .So when we consider MSE and PSNR SPIHT algorithm is suitable.

V. REFERENCE

[1] Reuben and ,A. Farrugia, " Reversible De-Identification for Lossless Image Compression using Reversible Watermarking ," MIPRO pp. 1258-1263, May 2014.

[2] Jan E. Odegard C. Sidney Burrus, “Smooth Biorthogonal Wavelets For Applications In Image Compression”, Department of Electrical and Computer Engineering Rice University, Houston, Texas 77005-1892, USA

[3] En-Hui Yang, Xiang Yu, Jin Meng, and Chang Sun, Transparent Composite Model for DCTCoefficients: Design and Analysis, IEEE Transactions on Image Processing, Vol. 23, No. 3, March 2014.

[4] Liu,C.P.; Poularikas,A.D.” A New Subband Coding Technique Using (JPEG) DCT for ImageCompression,” IEEE Trans. on Image processing, pp.317-321,1996.

[5] Said,A. Pearlman,W.A. “ A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, ” IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, pp. 243-250,1996.

[6] Amir Averbuch, Danny Lazar, and Moshe Israeli,"Image Compression Using Wavelet Transform and Multiresolution Decomposition"IEEE Trans. on Image Processing, Vol. 5, No. 1, JANUARY 1996.

[7] Antonini,M.; Barlaud,M.; Mathieu, P.; Daubechies,I.“ Image Coding Using Wavelet Transform, ” IEEE Trans. on Image Processing, Vol. 1, No. 2, pp.205-220.1992.

[8] Ronald A.DeVore; Bjorn Jawerth; Bradley J. Lucier " Image Compression Through Wavelet Transform Coding, " IEEE Trans. on Information Theory, Vol.38.NO.2,pp.719-746, MARCH 1992..

[9] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. on Computers, vol. C-23, pp. 90-93,1974.