INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS IN ENGINEERING, TECHNOLOGY AND SCIENCES (IJ-CA-ETS)

IMPLEMENTATION OF IMAGE COMPRESSION ALGORITHM ON MATLAB

[1]A. C. Suthar, [2]A. M. Kothari, [3]V. S. Vora, [4]Dr. G. R. Kulkarni

[1]Asst. Prof., Dept. of E. & C., C. U. Shah College of Engg. & Tech., Gujarat.
[2] Lecturer, Department of E. & C., A.I.T.S., Rajkot, Gujarat.
[3] Lecturer, Department of E. & C., A.I.T.S., Rajkot, Gujarat.
[3]Principal, C. U. Shah College of Engineering & Technology, Wadhwan. Gujarat.
[1]acsuthar@ccetvbt.org, [2]amkothair@aits.edu.in, [3] vsvora@aits.edu.in, [4]grkulkarni29264@yahoo.com

ABSTRACT:
The amount of data might be a problem from a storage point of view or when the data is sent over a network. To overcome these problems data compression techniques adapted to these applications are needed. An Image Compression is very hot area in today's communication era. Many different techniques are used to compress an image but we are going for SPIHT (Set Partition In Hierarchical Tree). A complete work involves the knowledge of Image processing, different issues related to image compression like MSE, PSNR, CR etc. and the detailed study of SPIHT algorithm.

1. INTRODUCTION

In a broad sense, image compression is based on the fact that the numerical values of neighboring pixels are very closely correlated. Those similar numerical values can be used for compression via the following strategies: **a)** Transform the original image, represented by a table or matrix, containing independent data for each pixel into a transform domain, which contain data about pixel groups. This will typically result in a transform representation with much smaller numerical values, which need fewer digits for representation. To allow perfect reconstruction of the original image, the transform needs to be perfectly reversible. **b)** Send the image, or the transformed data, with an efficient protocol. The protocol transmits efficient commands to the decoder, which then reconstructs the original bit stream. For perfect reconstruction, the entire bit stream of decoder commands must be recovered. **c)** Lossy compression is usually based on substituting the exact transformed data with data that allow reconstruction of an image similar to the original, but described with much fewer bits. The differences between the new and old images are the reconstruction errors.

A typical lossy image compression scheme is shown in Figure 1. The system consists of three main components, namely, the source encoder, the quantizer, and the entropy encoder. The input signal (image) has a lot of redundancies that needs to be removed to achieve compression [1]. These redundancies are not obvious in the time domain. Therefore, some kind of transform such as discrete cosine, fourier, or wavelet transform is applied to the input signal to bring the signal to the spectral domain. The spectral domain output from the transformer is quantized using some quantizing scheme [2],[3]. The signal then undergoes entropy encoding to generate the compressed signal.

2. OBJECTIVES

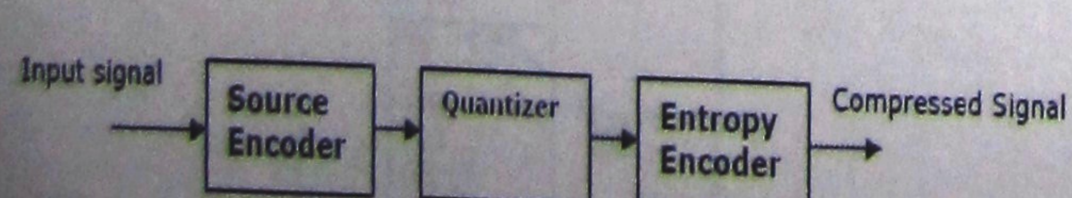

Figure:1 Image compression scheme

The objective of our paper is to compress image file and for compression we have used SPIHT-Set Partition in Hierarchical Tree Coding Technique [4].

For decompression of the image file we have used compression algorithm in reverse manner. The SPIHT algorithm we have to apply on color image file on its different color models RGB and HSI, with serial transmission and reception of gray level image using RS232. SPIHT-Set Partition in Hierarchical Tree Coding is a new digital image compression technique which is based on three concepts:

1) Partial ordering of the transformed image elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder. **2)** Ordered bit plane transmission of the refinement bits. **3)** Exploitation of self similarity of the image.

To apply SPIHT-Set Partition in Hierarchical Tree Coding we have to convert given image in to frequency domain (here we have used Discrete Cosine Transform) then we applied SPIHT algorithm and we achieved binary bit stream of compressed image. After that we used this binary bit stream in decoding algorithm and achieved reconstructed image.

3. WHY SPIHT?

Because of so many reasons like, Good image quality, High PSNR for color images, Optimized for progressive image transmission, Produces fully embedded coded files, Simple algorithm, Fast coding / decoding, Wide applications, Lossless compression, Efficient combination with error protection.

4. APPROACH FOR IMPLEMENTATION FOR COMPRESSION

The EZW algorithm was further extended by Amir et. al to give a new scheme called the Set Partitioning in Hierarchical Trees (SPIHT) [5]. SPIHT achieved better performance than the EZW without having to use the arithmetic encoder and so the algorithm was computationally more efficient. The SPIHT uses a more efficient subset partitioning scheme. Due to this, even binary encoded transmission achieves almost similar performance compared to EZW. The better performance of the SPIHT over EZW can be attributed to better wavelet filters (7/9 orthogonal wavelet filters instead of length 9 QMF filters), separation of the significance of the child nodes from that of the grand child nodes, and separation of the child nodes from the parent [6],[7].

Matlab offers a set of wavelet tools to be able to produce an image with the needed properties. The concept of wavelet transformation was not our focus in this project but in order to understand how the SPIHT algorithm works; the properties of wavelet transformation would need to be identified. Matlab is able to create adequate testing pictures for this project.

To adequately comprehend the advantages of the SPIHT algorithm, a top level understanding will be needed to identify its characteristics and differences from other algorithms.

5. SET PARTITIONING ALGORITHM

The SPIHT algorithm is unique in that it does not directly transmit the contents of the sets, the pixel values, or the pixel coordinates. What it does transmit is the decisions made in each step of the progression of the trees that define the structure of the image. Because only decisions are being transmitted, the pixel value is defined by what points the decisions are made and their outcomes, while the coordinates of the pixels are defined by which tree and what part of that tree the decision is being made on. The advantage to this is that the decoder can have an identical algorithm to be able to identify with each of the decisions and create identical sets along with the encoder.

The part of the SPIHT that designates the pixel values is the comparison of each pixel value to $2^n \leq | c_{i,j} | < 2^{n+1}$ with each pass of the algorithm having a decreasing value of n. In this way, the decoding algorithm will not need be passed the pixel values of the sets but can get that bit value from a single value of n per bit depth level. This is also the way in which the magnitude of the compression can be controlled. By having an adequate number for n, there will be many loops of information being passed but the error will be small, and likewise if n is small, the more variation in pixel value will be tolerated for a given final pixel value. A pixel value that is $2^n \leq | c_{i,j} |$ is said to be significant for that pass.

By sorting through the pixel values, certain coordinates can be tagged at "significant" or "insignificant" and then set into partitions of sets. The trouble with traversing through all pixel values multiple times to decide on the contents of each set is an idea that is inefficient and would take a large amount of time. Therefore the SPIHT algorithm is able to make judgments by simulating a tree sort and by being able to only traverse into the tree as much as needed on each pass. This works exceptionally well because the wavelet transform produces an image with properties that this algorithm can take advantage of. This "tree" can be defined as having the root at the very upper left most pixel values and extending down into the image with each node having four (2 x 2 pixel group) offspring nodes (see figure 2).

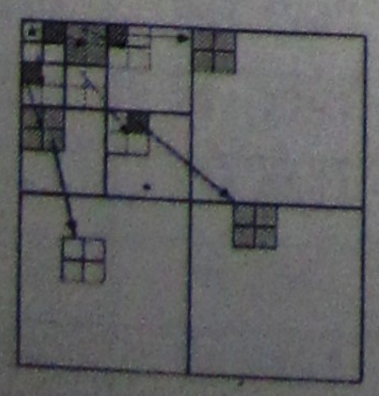

Figure 2: Partition in image

The following sets of coordinates are used to present the new coding method:

O(i,j): set of coordinates of all offspring of node *(i,j)*; *children only*

D (i,j): set of coordinates of all descendants of node *(i,j)*; *children, grandchildren, great-grand, etc.*

H (i,j): set of all tree roots (nodes in the highest pyramid level); *parents*

L (i,j): ***D** (i,j)* – ***O**(i,j)* (all descendents except the offspring); *grandchildren, great-grand, etc.*

For instance, except at the highest and lowest pyramid levels, we have

$O(i,j)=\{2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}$

We use parts of the spatial orientation trees as the partition- using subsets in the sorting algorithm. The set partition rules are simply the following.

1) The initial partition is formed with the sets ((2, *3*)) and *D(z, J)* , for all *(2 , J)* E *X*.

2) If *D(z, J)* is significant, then it i s partitioned into C(z, *J)* plus the four single-element sets with *(k , I)EO(z, 2)* .

3) If *C(z, J)* is significant, then it is partitioned into the four sets *D (k ,I)* , with *(k ,I) E O(z,J)* .

The following are the lists that will be used to keep track of important pixels:

LIS: List of Insignificant Sets, this list is one that shows us that we are saving work by not account ting for all coordinates but just the relative ones.

LIP: List of Insignificant Pixels, this list keeps track of pixels to be evaluated

LSP: List of Significant Pixels, this list keeps track of pixels already evaluated and need not be evaluated again.

The SPIHT algorithm consists of three stages:

(1) initialization,
(2) sorting and
(3) refinement

6. INITIALIZATION

n = ⌊log2 (max |coeff|)⌋
LIP = All elements in H
LSP = Empty
LIS = D's of Roots

Significance Map Encoding ("Sorting Pass") Process LIP

for each *coeff (i ,j)* in LIP
    check
  if {absolute value *(i, j)*} >=2n
      Output=1
    Output the sign of coefficient *(i,j)*
      for negative value output = 0
      for positive value output = 0
    move *(i, j)* to LSP
else
    Output=0

Process LIS

For each set *(i ,j)* in LIS
if entry of type *D*
    check
if max { absolute value *(i ,j)* } >= 2n
    Output=1
divide *D* into offsprings *O(i ,j)*
  for each offsprings of *D*
if {absolute value *(i, j)*} >=2n
    Output =1
    Output the sign of coefficient *(i,j)*
      for negative value output = 0
      for positive value output = 0

move (i, j) to LSP
else
Output=0
move (i, j) to LIP
if entry of type L
check
if max { absolute value (i, j) } >= 2n
Output = 1
divide L into descendents $D(i,j)$
for each descendents of L
move $D(i, j)$ to LIS and done same procedure as for the type D entry
else
Output=0
keep L entry as it is in LIS.

Refinement Pass

Process LSP

For each element of LSP in the previous pass (value of $n = n\text{-}1$)
Output the nth most significant bit of coeff.

Update

Decrement value of n by 1
so now $n = n - 1$
go to the step for processing LIP and repeat the procedure till $n = 0$.

7. WORK PERFORMED

Matlab, popular mathematical modeling software, is used for the software implementation. The image format used is the Portable Grey Map (PGM) format in which pixels are stored in the unsigned char type, providing a maximum of 256 gray scale levels or 8-bit data per pixel. The grayscale format was chosen to keep the algorithm fairly simple compared to processing for RGB 3 color or more planes. The Matlab programming environment provided all the necessary functions and tools needed to achieve this.
The SPIHT algorithm would need an adequate image sample to perform the encoding on so both images were used to explore the properties of the SPIHT algorithm.
Having the SPIHT algorithm encoded into matlab meant that matrix data structures, recursive algorithms, and a bit stream would be needed. The algorithm was encoded using multiple loops to be able to scan through all the pixel value and be able to evaluate the significance of each. At the same time, the sets would need to be repetitively access to be able to continually update the composition of each.

8. EXPERIMENTAL RESULTS

We use the computer: CPU Pentium IV, 256 RAM with windows XP operating system. Experimental environment MATLAB 7. We take Gray scale image cameraman and RGB image peppers as experimental object and use the objective evaluation standards PSNR to take this paper evaluation criterion. The experimental data is as follows:

Table I Result for Gray Scale image

Rate	CR	MSE	PMSE
0.1	79.5656	407.4655	22.0299
0.2	39.7312	291.5747	23.4833
0.4	19.8550	181.2791	25.5433
0.6	13.2343	131.4871	26.9420
0.8	9.9250	98.2241	28.2086
1.0	7.9394	76.6894	29.2835

Table II Result for RGB Color image

Rate	CR	MSE	PMSE
0.1	106.2887	0.0194	72.3499
0.2	53.1173	0.0192	73.9779
0.4	26.5519	0.0190	75.9143
0.6	17.6998	0.0189	77.2403
0.8	13.2743	0.0188	78.2861
1.0	10.6783	0.0189	79.4321

9. CONCLUSION

SPIHT is presently one of the most efficient known for lossy compression both in terms of speed and compression.
The SPIHT algorithm is very efficient for high compression rate when N is large but does not minimize memory nor bandwidth and is not designed to look at regions of interest, as opposed to JPEG 2000.
The run-length and Huffmann coding can quantize the analyzed image economically by the SPIHT is very effective on color images compare to gray level images [8], the compression ratio for gray level images varies from 7.93 to 79.5656 while for the same image if we look at RGB image then it will be 10.2342 to 75.1378 and his 10.02 to 68.13 hence at the cost of minor compression ratio we get good results in form of PSNR and MSE.

10. REFERENCES

1. *Sayood K. (2000), Introduction to data compression, Morgan Kaufmann Publrs, US, 2 revised edition, ISBN 1558605584.*
2. *Chui C. K. (1992), An introduction to wavelets, Boston, Academic Press, ISBN 0121745848.*
3. *Daubechies I., Barlaud M., Antonini M., Mathieu P. (1992), "Image coding using wavelet transform" in IEEE Transactions on image processing, Vol. 1, No. 2, pp. 205-220.*
4. *Forchheimer R. (1999), Image coding and data compression, Linköping: Department of electrical engineering at Linköpings University.*
5. *Said A., Pearlman W. A.(1996), "A new, fast, and efficient image codec based on set portioning in hierarchical trees" in IEEE Transactions on circuits and systems for video technology, Vol. 6, No. 3, pp. 243-250.*
6. *Brislawn C. M.. (1996), "Classification of nonexapnsive symmetric extension transforms for multi rate filter banks" in Applied and computational harmonic analysis, Vol. 3, pp. 337-357.*
7. *Azpiroz-Leehan J., Lerallut J.-F. (2000), "Selection of biorthogonal filters for image compression of MR images using wavelet packets" in Medcial engineering & physics, Vol. 22, pp. 225-243.*
8. *Christopoupolus C., Athanassios S., Ebrahimi T. (2000), "The JPEG2000 still image coding system: An overview" in IEEE Transactions on consumer electronics, Vol. 46, No. 4, pp. 1103-1127.*