



Volume: 2, Issue: 5, 452-455
May 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Nandini Dhole

Department of Electronics &
Telecommunication,
RMDSSOE, Savitribai Phule
Pune University, Warje,
Pune, India

Kunal Deshmukh

Department of Electronics &
Telecommunication,
RMDSSOE, Savitribai Phule
Pune University,
Warje, Pune, India

DWT based set partitioning in hierarchical trees algorithm for image compression

Nandini Dhole, Kunal Deshmukh

Abstract

Embedded zero-tree wavelet (EZW) coding, proposed by J. M. Shapiro, is a very computationally simple and effective technique for image compression. Now we express an alternative description of the principles, so that the purpose for its better performance can be well understood. These principles are ordered bit plane transmission, an image wavelet transform self-similarity across different scales and a set partitioning sorting algorithm as a partial ordering by magnitude. Also we present an optional and efficient implementation based on set partitioning in hierarchical trees (SPIHT), image compression algorithm which is very effective, less complexity and generates an embedded compressed bit-stream that can be effectively decoded at several data rates. The results of image coding, obtained from reconstructed images by the decoding algorithm and actual file sizes, are comparable to or surpass previous results obtained through much more computationally complex and sophisticated methods. In addition, this new coding and decoding techniques are very fast.

Keywords: EZW, SPIHT, DWT

Introduction

Compression is a process of reducing the number of data bits necessary for representing information, to properly utilize the available bandwidth and reduce storage space. There are two types of compression namely lossless compression and lossy compression. Image compression reducing the amount of data required to represent a digital image with no significant loss of information. For Image Compression capability of data storage, bandwidth of communication, consumption of power and time for processing are critical parameters. With the rapid development of digital technology in consumer electronics, the requirement to save raw image data for future editing or repeated compression is increasing. In the context of image processing, compression process are aimed to reduce the images transmission rate, with a better level of visual quality. General aim of compression technique is to compress images, but the result is less than required².

Lossless Compression

In lossless compression, data can be completely recovered after decompression. Recovered data is perfectly identical to original data, utilize redundancy in data. Lossless Image compression is the only acceptable amount of data reduction. It provides low compression ratio while compared to lossy. In Lossless Image compression techniques are composed of two relatively independent operations.

- Formulate an optional representation of the image in which its inter-pixel redundancies are reduced.
- Coding the representation to eliminate coding redundancies.

Lossless Image compression is useful in applications such as medical imaginary, business documents and satellite images³.

Lossy Compression

Data cannot be recovered after decompression; some information is lost forever, provides more compression as compare to lossless and discards “insignificant” data components. Lossy compression gives higher levels of data reduction but result in a less than perfect reproduction of the original image. The compression ratio obtained is much better. Lossy image compression is most widely useful in applications such as videoconferencing, broadcast television, in which some amount of error is an acceptable trade-off for increased compression performance.

Correspondence:

Nandini Dhole

Department of Electronics &
Telecommunication,
RMDSSOE, Savitribai Phule
Pune University, Warje,
Pune, India

There are many algorithms based on image compression. One of the most efficient algorithms is the Set Partitioning in Hierarchical Trees (SPIHT) algorithm. SPIHT (Set Partitioning In Hierarchical Trees) has been most widely applied for image coding which is based on wavelet transform, was proposed by Said and Pearlman¹. SPIHT as an efficient method for lossy and lossless coding of still images was an advance over the innovative wavelet based image coding method of which employed a tree representation of zeroes of wavelet coefficients for the coding of these coefficients. Zero-tree image coding itself yielded several dB's of signal-to-noise-ratio improvement for most real world still images at low bit rates over the DCT (Discrete Cosine Transform) based older JPEG still image coding standard⁵. This paper organized as follows. In next section II, the Discrete Wavelet Transform is described, as the SPIHT is dependent on DWT. Section III explains SPIHT algorithm in detail with flow chart. At last the conclusion in section IV.

Discrete Wavelet Transformation Overview Wavelet Image Compression

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each component with a resolution matched to its scale. The basic idea of the wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets is basis function. These starting point functions are baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by amplification or contractions and translations.

Discrete Wavelet Transformation

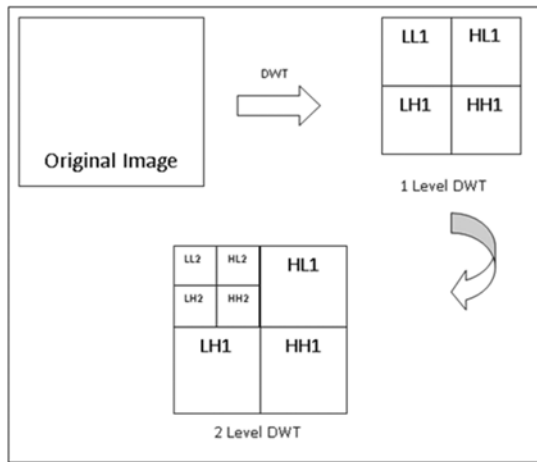


Fig: 1 Frequency Distribution of DWT

The second method of this mechanism uses 2-D Discrete Wavelet Transformation (DWT). DWT also converts the image to the frequency domain from its spatial domain. As shown in the Fig. 1, the image is divided by vertical and horizontal lines and represents the first-order of DWT, and the image is transferred into four subparts those are LL1, HH1, HL1, and LH1. These four parts are shows four frequency region in the image. For the band LL1 is sensitively with human eyes. In the frequency band HH1, HL1, and LH1 have more important information as compare to frequency domain LL1.

1-D Discrete Wavelet Transform

The discrete wavelets transform (DWT) convert a discrete time to discrete wavelet signal. The first step is to discretize the wavelet parameters that reduce the continuous basis set of wavelets to a discrete and orthogonal set of basis wavelets.

2-D Wavelet Transform

The 1-D DWT can be enhanced to 2-D transform using separable wavelet filters. The one-dimensional transform to all the rows of the input and then repeating on all of the columns can process the 2-D transform, by the use of separable filters. When one-stage 2nd dimensional DWT is applied to an image, four sets of coefficient as LL, HH, HL, and LH, are created, Where L corresponds to Low and H corresponds to High.

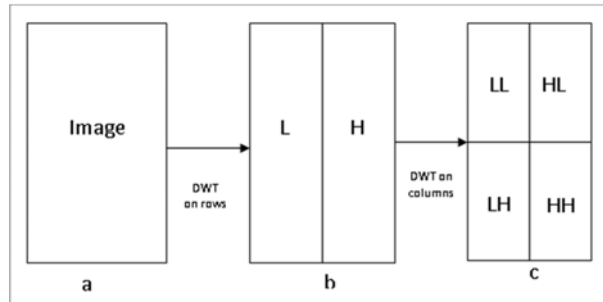


Fig: 2 Block Diagram of DWT (a) Input Original Image (b) Image after the 1-D applied on Row (c) Image after the second 1-D applied on row

The Two-Dimensional DWT transforms images to frequency domain from spatial domain. At every level of the wavelet decomposition, using a one-dimensional vertical analysis filter-bank, column is first changed. After that the filter-bank is applied horizontally to every row of the filtered and sub-sampled data. To gather data energy in the lowest sampled bands, we also use multiple levels of wavelet transforms. Specifically, the LL sub band in fig. 2(c) can be alter again to form sub-bands LL1, HH1, HL1, and LH1, producing a two-level wavelet transform.

SPIHT

SPIHT Introduction

The data structure used by the SPIHT algorithm is similar to that used by the EZW (Embedded Zero tree Wavelet) algorithm-although not the same. The wavelet coefficients are again divided into trees originating from the lowest resolution band (band I in our case). The coefficients are grouped into 2 × 2 arrays that, except for the coefficients in band I, are off-springs of a coefficient of a lower resolution band. The coefficients in the lowest resolution band are also divided into 2 × 2 arrays. The coefficient in the top-left corner of the array does not have any off-springs. The data structure is shown pictorially in Fig. 3, for seven-band decomposition.

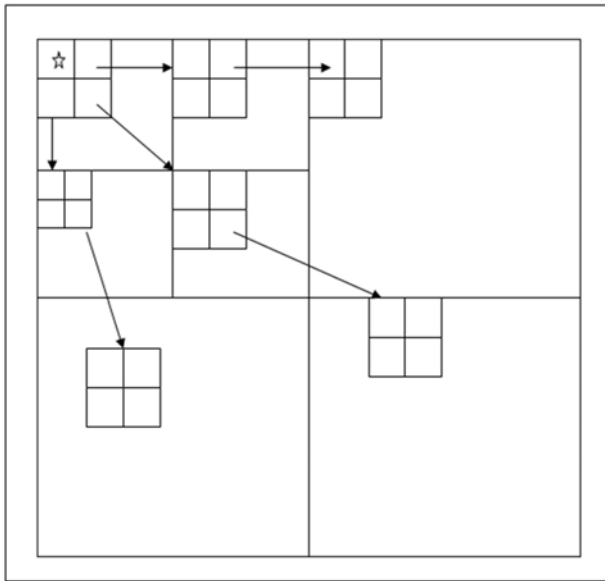


Fig: 3 Data structure used in the SPIHT algorithm

SPIHT Algorithm

SPIHT consists of two passes, the ordering pass and the refinement pass. In the ordering pass SPIHT attempts to order the coefficients according to their magnitude. In the refinement pass the quantization of coefficients is refined. The ordering and refining is made relative to a threshold. The threshold is appropriately initialized and then continuously made smaller with each round of the algorithm. SPIHT maintains three lists of coordinates of coefficients in the decomposition. These are the List of Insignificant Pixels (LIP), the List of Significant Pixels (LSP) and the List of Insignificant Sets (LIS). To decide if a coefficient is significant or not SPIHT uses the following definition. A coefficient is deemed significant at a certain threshold if its magnitude is larger than or equal to the threshold. Using the notion of significance the LIP, LIS and LSP can be explained. The LIP contains coordinates of coefficients that are insignificant at the current threshold; The LSP contains the coordinates of coefficients that are significant at the same threshold. The LIS contains coordinates of the roots of the spatial parent-children trees.

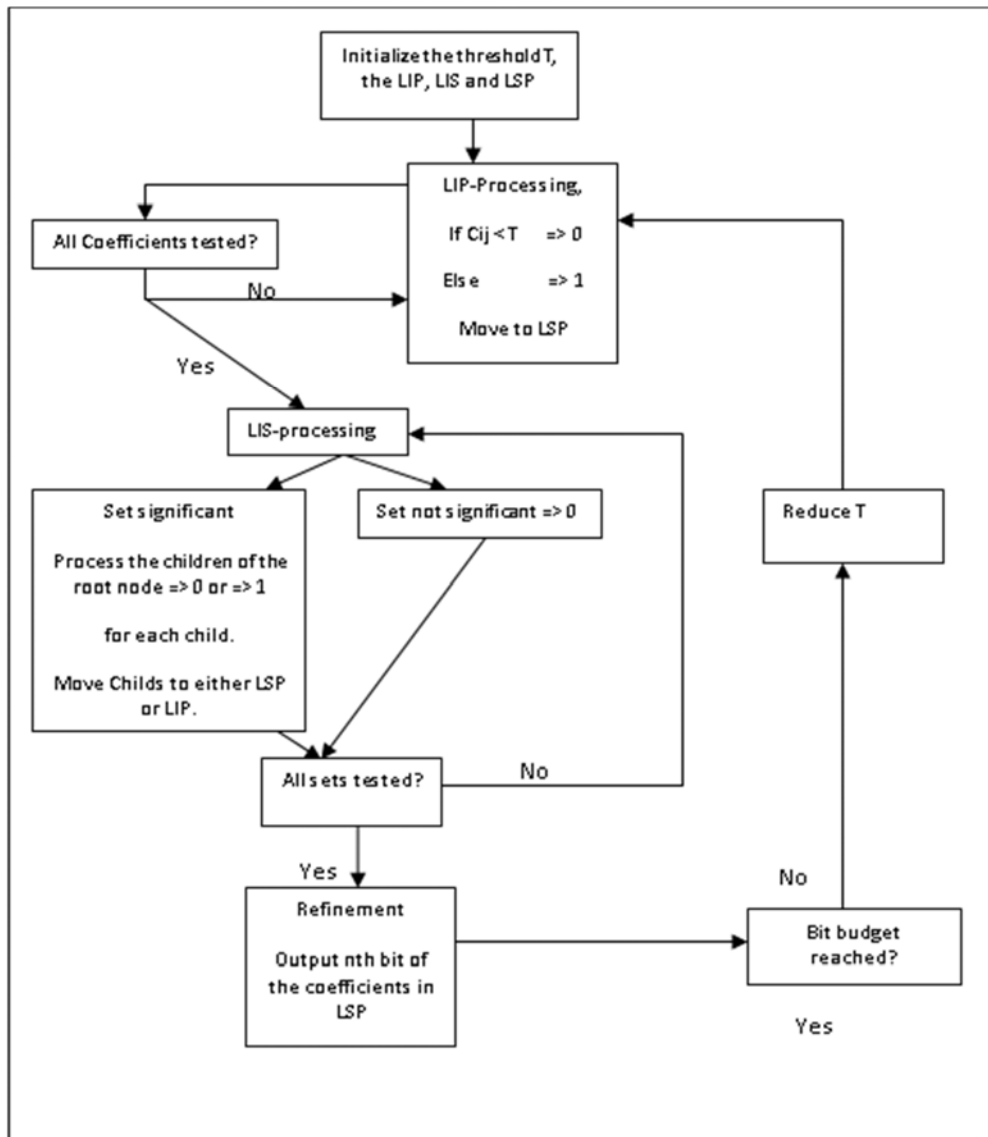


Fig: 4 SPIHT Algorithm

The trees are further partitioned into four types of $D(i, j)$ sets, which are sets of coordinates of the coefficients $O(i, j)$. This is the set of coordinates of the off-springs of the wavelet coefficient at location (i, j) . As each node can either have four off-springs or none, the size of $O(i, j)$ is either zero or four. For example, in Fig. 4, the set $O(i, j)$ consists of the coordinates of the coefficients b_1, b_2, b_3 and b_4 . $D(i, j)$ This is the set of all descendants of the coefficient at location (i, j) . Descendants include the off-springs, the off-springs of the off-springs, and so on. For example, in Fig. 4, the set $D(i, j)$ consists of the coordinates of the coefficients $b_1, \dots, b_4, b_{11}, \dots, b_{14}, \dots, b_{44}$. because the number of off-springs can either be zero or four, the size of is either zero or a sum of powers of four. H This is the set of all root nodes- essentially band I in the case of Fig. 4. $L(i, j)$ This is the set of coordinates of all the descendants of the coefficient at location (i, j) except for the immediate off-springs of the coefficient at location (i, j) .

In other words $L(i, j) = D(i, j) - O(i, j)$.

The algorithm makes use of three lists: the list of insignificant pixels (LIP), the list of significant pixels (LSP) and the list of significant sets (LIS). The LSP and LIS lists will contain the coordinates of coefficients, while the LIS will contain the coordinates of the roots of sets of type D or L. We start by determining the initial value of threshold. We do this by calculating,

$$n = \lfloor \log_2 c_{max} \rfloor$$

Where c_{max} is the maximum magnitude of the coefficients to be encoded. The LIP list is initialized with the set H. Those elements of H that have descendants are also placed in LSP as type D entries. The LSP list is initially empty.

SPIHT shows exceptional characteristics over several properties all at once and these are:

- 1 Fast coding and decoding.
- 2 Good image quality with a high PSNR.
- 3 Can be used for lossless compression.
- 4 May be combined with error protection.
- 5 Ability to code for exact bit rate or PSNR.
- 6 A fully progressive bit-stream.

Conclusion

We have presented an algorithm that operates through set partitioning in hierarchical trees (SPIHT) and fulfill complete embedded coding. This SPIHT algorithm uses the principles of set partitioning by significance of magnitudes with respect to a sequence of actively decreasing thresholds, ordered bit plane transmission, partial ordering by magnitude and self-similarity across scale in an image wavelet transform. The realization of these principles is matched coding and decoding algorithm is a new one and is shown to more effective than the previous implementation of EZW coding.

References

1. Amir Said, William A. Pearlman, "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Trans. on Circuits and Systems for Video Technology, June 1996, Vol. 6, No. 3, pp. 243-250.
2. Kai Liu, Evgeniy Belyaev, Jie Guo, "VLSI Architecture of Arithmetic Coder Used in SPIHT", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, April 2012, Vol. 20, No. 4, pp. 697-710.

3. Venkata Anjaneyulu, P. Rama Krishna, "FPGA Implementation Of DWT SPIHT Algorithm", for Image Compression International Journal of Technology Enhancements and Emerging Engineering Research, 2014, Vol. 2, Issue 3, pp. 20-24.
4. R. Vanaja, N. Lakshmi Praba, "FPGA Implementation of Pipelined Architecture For SPIHT Algorithm", IEEE International Conference For Emerging Trends in Computing, Communication and Nanotechnology, 2013, pp. 456-461.
5. "Algorithmic Modifications to SPIHT" by Ulug Bayazitl and William A. Pearlman.
6. Dipalee Gupta, Siddhartha Choubey, "Discrete Wavelet Transform for Image Processing", International Journal of Emerging Technology and Advanced Engineering, March 2015, Vol. 4, Issue 3, pp. 598-602.
7. Chandandeep Kaur, Sumit Budhiraja "Improvements of SPIHT in Image Compression- Survey", International Journal of Emerging Technology and Advanced Engineering, January 2013, Vol. 3, Issue 1, pp. 652-656.
8. Pasquale Corsonello, Stefania Perri, Paolo Zicari, Giuseppe Cocorullo, "Microprocessor-based FPGA implementation of SPIHT image compression subsystems", Elsevier Microprocessors and Microsystems, 2005, pp. 299-305.
9. Anilkumar V. Nandi, Dr. R. M. Banakar, "Throughput efficient Parallel Implementation of SPIHT algorithm", IEEE 21st International Conference on VLSI Design, 2008, pp. 718-723.
10. Ali Kadhim Al-Janabi, "Low Memory Set-Partitioning in Hierarchical Trees Image Compression Algorithm", International Journal of Video and Image Processing and Network Security, April 2013, Vol. 13, No. 02, pp. 12-18.
11. R. Vanaja, N. Lakshmi Praba, DR. N. Stalin, "Efficient Architecture for SPIHT Algorithm in Image Compression", International Journal of Advanced Research in Computer Science Engineering and Information Technology, June 2013, Vol. 1, Issue 1, pp.37-44.