ISSN No:-2456 –2165

Efficient Arithmetic Coder Design for SPIHT Image Compression

Rahila I. Mulla
M.E* Student: VLSI and Embedded Systems
ADCET, Ashta ,Dist: Sangli , India
Rahila1361@gmail.com

Prof. Rupali R. Jagtap
Asst. Professor: Dept. of Electronics & Telecomm.
ADCET, Ashta ,Dist: Sangli , India
rupajagtap@gmail.com

Abstract—A memory-efficient arithmetic coder design using MATLAB and Verilog HDL for the set partitioning in hierarchical trees (SPIHT) image compression is illustratedin this paper. In this wavelet transform is performed on the image first then coefficients are processed in SPIHT compression module. Compressed bitstream obtained from SPIHT module is applied to Arithmetic coder module whose output is an encoded bitstream. Output is tested for bits/pixel efficiency and compared with other compression models. This design of SPIHT with AC gives very good compression coding rate bits/pixel. MATLAB and Modelsim are used for simulation.

Keywords—Arithmetic Coding, Coding Efficiency (Bits/Pixel), Set Partitioning in Hierarchical Trees (SPIHT), Verilog, Wavelet.

I. INTRODUCTION

Set Partitioning in Hierarchical Trees (SPIHT) is image compression methodusing wavelet transform that givesa very good characteristics. These characteristics are:

- Good image quality
- Progressive image transmission
- Produces a fully embedded coded file
- Simple quantization algorithm
- Can be used for lossless compression
- Can code to exact bit rate or distortion
- Fast coding/decoding (nearly symmetric)
- Has wide applications, completely adaptive
- Efficient combination with error protection.

The main advantage of SPIHT is its fully progressive behavior, means it does not need complete file to see the image. The image's PSNR or its quality is directly related to the quantity of the file received from the transmitter, so quality of image will increase with the amount of the file received. After the SPIHT compression, some redundancy will exist in the file. This redundancycan be removed using efficient encoding stage. So addition of arithmetic compression to a SPIHT encoded image gives very good compression results. A few different compression results are compared.

II. OVERVIEW OF ARITHMETIC CODING

Arithmetic coding is a method for lossless data compression. Arithmetic coding is an encoding method that can give compression levels at or very near entropy. What this means is that if we have a message composed of symbols over some finite alphabet, we can generate the exact number of bits that corresponds to a symbol (e.g. 1.6 bits/symbol). This is opposed to Huffman encoding which must output an integer number of bits per symbol (e.g. 2 bits/symbol).Arithmetic coding achieves entropy (or very near it) by grouping symbols together until an integer value of bits can be outputted for a sequence of symbols (e.g. ABC may correspond to 1011).

In arithmetic coding a unique tag is generated for the sequence of symbolsthat is to beencoded. This tag is related to a binary fraction and becomes the binary code for the sequence. Actually the tag generation and the binary code are the same process.Infact, the arithmetic coding method is easyto understand if we dividethe process into two parts. In the first part a unique tag is generated for agiven sequence of symbols. This tag is then given a unique binary code. A unique arithmeticcode can be generated for a sequence of length m without the need for generating code wordsfor all sequences of length m.

Arithmetic coding works by using a probability interval defined with variables L and R, which are initially set to 0 and 1 respectively. The value of L represents the smallest binary value consistent with a code representing the symbols processed so far.

The value of R represents the product of the probabilities of those symbols. To encode the next symbol, which is the j^{th} of the alphabet, both L and R must be recomputed. L and R get the following values:

$$L = L + R \times \sum_{i=1}^{j-1} p_i$$

$$R = R \times p_i$$

ISSN No:-2456 –2165

III. RELEVANCE

As arithmetic coding (AC) can obtain optimal performance for its ability to generate codes with fractional bits, it is widely used by various image compression algorithms.Specially, the set partitioning in hierarchical trees (SPIHT) uses an AC method to improve its peak signal-to-noise ratio (PSNR) and compression ratio bits/pixel. Although the theory and program code of AC are mature, the complicated internal operations of AC limit its application for some real time fields, such as satellite image and high speed camera image compressions. In order to achieve performance gains, efficient architecture of AC in compression scenarios must be designed to satisfy the throughput requirement.

IV. LITERATURE REVIEW

D. Marpe, H. Schwarz, and T. Wiegand presented Context-Based Adaptive Binary Arithmetic Coding (CABAC) as a normative part of the new ITU-T/ISO/IEC standard H.264/AVC for video compression [2]. By combining an adaptive binary arithmetic coding method with context modeling, a high degree of adaptation and redundancy reduction is achieved.The CABAC framework also contain a low-complexity method for binary arithmetic coding.

A Variant of SPIHT image compression algorithm called No List SPIHT (NLS) is presented by F. W. Wheeler and W. A. Pearlman [3]. NLS works without linked lists and goodfor a fast, simple hardware implementation. Instead of Lists, a state table with four bits per coefficient keeps track of the set partitions and what information has been encoded. Performance of the algorithm on standard test images is nearly the same as in SPIHT.

H. Pan, W.C. Siu, and N.F. Law, proposed A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT [5] . In this paper, Author illustrates a listless modified SPIHT (LMSPIHT) method, it is a fast and low memory image coding algorithm based on the lifting wavelet transform. The LMSPIHT jointly gives the advantages of progressive transmission, spatial scalability, and includes human visual system (HVS) characteristics in the coding method.

Authors A. A. Kassim, N. Yan, and D. Zonoobi presented [6] The wavelet-based set partitioning in hierarchical trees (SPIHT) it is highly efficient in coding non textured images and the wavelet packet transform (WPT) is able to provide an optimal representation for textured images.

The proposed SPIHT-WPT coder achieves improved coding gains, especially for highly textured images.

V. METHODOLOGY

- High resolution Image captured using Camera.
- Input image is converted to pixel values in MATLAB.
- Pixel values are given as input to the Line based wavelet lifting module which produces wavelet coefficients.
- The transformed wavelet coefficients are placed in a buffer and they are accessed in a SPIHT-Breadth First Search way. Transformed wavelet coefficients are given to the arithmetic coder through internal bus.
- Output of arithmetic coder is applied to the internal bus and sent to output stream.

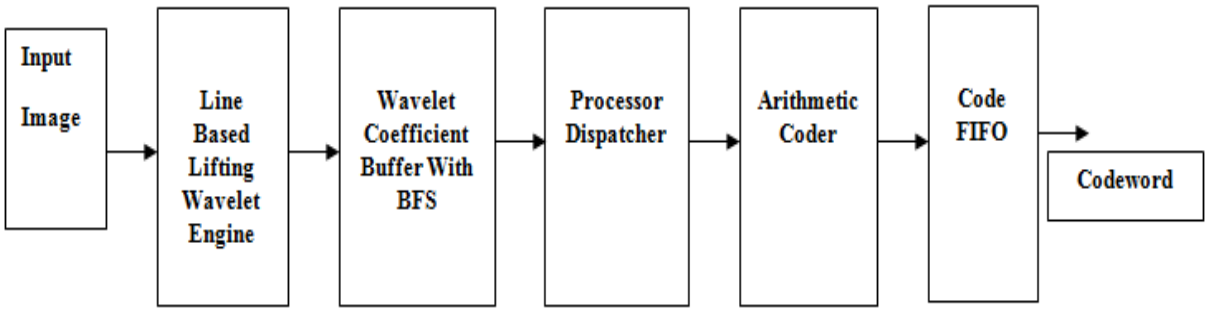

Fig.1 Block Diagram of Proposed Work

VI. SPIHT ALGORITHM

The algorithm for SPIHT makes use of tree partitioning in which itkeep insignificant coefficients together in large subsets. The partitioning decisions are in binary format that are send to the decoder and providesa significance map encoding which is more efficient than EZW.

The trees are again partitioned into four types of sets, which are sets of coordinates of the coefficients:**O(i , j) :** set of coordinates of alloffsprings of the wavelet coefficient node of location (i , j). As each node can either have four offsprings or no offspring ,O (i , j) will have size either zero or four. For example, in Fig.II,O (0, 1) consists of the coordinates of the coefficients b1, b2, b3, and b4.

D(i ,j) :set of all descendants of the coefficient at location (i ,j). Descendantsinclude offsprings and the offsprings of the offsprings, and so on.

H :set of all root nodesessentially band I in the case of Fig.2.
L (i , j) :set of coordinates of all the descendants of the coefficient at location(i ,j) it excludes the immediate offsprings of the coefficient at location (i ,j). It means,

$$L(i, j) = D(i, j) - O(i, j)$$

SPIHT algorithmuses three lists: the list of insignificant pixels (LIP), the list of significant pixels(LSP), and the list of insignificant sets (LIS).

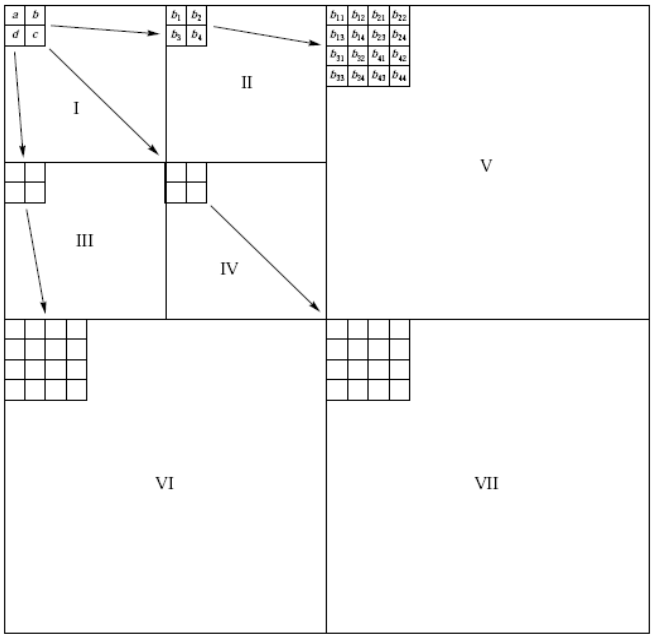

Fig. 2 Data Structure Used In SPIHT Algorithm

The lists LSP and LIS includes thecoordinates of coefficients, while the LIS will contain the coordinates of the roots of setsof type D or L. first determine the initial value of the threshold. We can do this bycalculating

$$n = \log_2 [Cmax]$$

Where, Cmax = maximum magnitude of the coefficients i.e. to be encoded.

The LIP list isinitialized with the set H. Those elements of H that have descendants are also placed inLIS as setD entries. The LSP list is initially empty. In significance map encoding step first in each pass process the members of LIP , then the members of LIS . We then process the elements of LSP inthe refinement step.

Algorithm begins by checking each coordinate in LIP. If the coefficient of thatcoordinate is significant i.e. it is greater than 2^n, we will transmit a 1 followed by a bitindicating the sign of the coefficient (here, 1 is for positive and 0 for negative).After this move the corresponding coefficient to the LSP list. If the coefficient at that coordinate is notsignificant, we transmit a 0.After examining each coordinate in LIP, we begin examining the sets in LIS. If the set atcoordinate **(**i , j) is not significant, we transmit a 0. If the set is significant, we transmit a 1.What we do after that depends on whether the set isD or L.If the set is D, we check each of the offsprings of the coefficient at that coordinate.In other words, we check the four coefficients whose coordinates are in O (i ,j). For eachcoefficient that is significant, we transmit a 1, the sign of the coefficient, and then move thecoefficient to the LSP. Transmit a 0 for all other coefficients i.e. insignificant and add their coordinates to the LIP.Here, we have removed the coordinates of O(i ,j) from the set, what is left is simply theset L(i , j). If this set is not empty, we move it to the end of the LIS and mark it to be oftype L. Note that this new entry into the LIS has to be examined during *this* pass. If the setis empty, we remove the coordinate (i , j) from the list.If the set is of type L, we add each coordinate in O(i ,j) to the end of the LIS as theroot of a set of type D. Again, note that these new entries in the LIS have to be examinedduring this pass. We then remove (i ,j) from the LIS.If we processed every set in the LIS, we can proceed to the refinement step. In the refinement step we examine each coefficient thatwas in the LSP prior to the current passand output the nth most significant bit of $|C_{ij}|$ We ignore the coefficients that have been added to the list in this pass because, by declaringthem significant at this particular level, we have already informed the decoder of the valueof the nth most significant bit.

This completes one pass. Depending on the availability of more bits or external factors,if we decide to continue with the coding process, we decrement n by one and continue.

VII. WORKING PROCESS

- START
- using MATLAB select desired image
- convert image into pixel values and write them rowwisen inimagetextfile.txt and coloumnwise in imagetextfilep.txt
- MATLAB call HDLdeamon socket Simulink handle to open modelsim for further process
- Modelsimruns SPIHT compression module
- save resulted coefficients in text file
- text files accessed in MATLAB and displayed as image to show SPIHT transformation in image format as well as coefficients are saved in arrays
- MATLAB initializes modelsim operation
- Modelsimruns arithmetic coder module
- SPIHT compressed coefficients are now processed in arithmetic coder module and encoded sequences are stored in memory designed using verilog code.
- compression efficiency bits/pixel is calculated in MATLAB and displayed
- End

ISSN No:-2456 –2165

VIII. RESULTS

Input image:
Input image is Lena (512 X 512)

Approximate Coefficient:

Detail Coefficient:

Horizontal Coefficients:

Vertical Coefficients:

Detail Coefficient 2:

Approximate Coefficient 2:

After SPIHT Transformation:

ISSN No:-2456 –2165

Here, we have compared coding efficiency of SPIHT with AC and SPIHT with Huffman coding, we can see no. of bits required to represent pixel is less in case of SPIHT with Arithmetic coder.

Image	Bits/pixel	
	SPIHT with Arithmetic coder	**SPIHT with Huffman coder**
Lena	0.1184	2.43583
Airport	0.2985	4.86931
Bike	0.0747	7.23228
Cafe	0.0339	2.66317
Pentagon	0.1985	4.66549
Woman	0.2791	9.18014
Peppers	0.2455	11.32722

Table 1: Coding Efficiency Comparison

IX. CONCLUSION

This SPIHT with AC (arithmetic coder) can meet many high speed image compression requirements.In Real time fields Satellite images , High Speed Camera Images it gives excellent compression results.

REFERENCES

[1]. Kai Liu, Evgeniy Belyaev, and Jie Guo, "VLSI Architecture of Arithmetic coder used in SPIHT," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol.20, No.4, April 2012.

[2]. D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. for Video Technol., vol. 13, no. 7, pp. 620–636, Jul. 2003.

[3]. F. W. Wheeler and W. A. Pearlman, "SPIHT image compression without lists," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Istanbul, Turkey, Jun. 2000, pp. 2047–2050.

[4]. A. Said and W. A. Pearlman, "A new ,fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. for Video Technol., vol. 6, no. 3, pp. 243–249, Mar. 1996.

[5]. H. Pan, W.-C. Siu, and N.-F. Law, "A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT," Signal Process.: Image Commun., vol. 23, no. 3, pp. 146–161, Mar. 2008.

[6]. A. A. Kassim, N. Yan, and D. Zonoobi, "Wavelet packet transform basis selection method for set partitioning in hierarchical trees," J. Electron.Imag., vol. 17, no. 3, p. 033007, Jul. 2008.

[7]. M. A. Ansari and R. S. Ananda, "Context based medical image compression for ultrasound images with contextual set partitioning in hierarchical trees algorithm," Adv. Eng. Softw., vol. 40, no. 7, pp. 487–496,Jul. 2009.

[8]. M. Akter, M. B. I. Reaz, F. Mohd-Yasin, and F. Choong, "A modified- set partitioning in hierarchical trees algorithm for real-time image compression," J. Commun. Technol. Electron., vol. 53, no. 6, pp. 642–650, Jun. 2008.

[9]. J. Bac and V. K. Prasanna, "A fast and area-efficient VLSI architecture for embedded image coding," in Proc. Int. Conf. Image Process., Oct.1995, vol. 3, pp. 452–455.

[10]. I. C.Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," Commun. ACM, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[11]. Khalid Sayood, "Introduction to Data Compression", Third edition.