International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 7, July - 2013

Progressive Coding For Image Compression

R. D. Kekade
Department of Electronics & Telecommunication
Maharashtra Institute of Technology, Pune

Abstract - **New algorithms for image compression based on wavelets have been recently developed. One of the most efficient algorithms is the Set Partitioning in Hierarchical Trees (SPIHT) algorithm. The SPIHT algorithm orders the wavelet coefficients according to a significance test and stores this information in three separate lists. This paper describes an implementation of discrete wavelet transform based image compression using SPIHT coding in the MATLAB environment. Although, SPIHT is well known for its simplicity and efficiency, it requires a large amount of memory for storing the coordinates of wavelet coefficients. A great number of operations to manipulate the memory are also required in the codec scheme, which greatly reduces the speed of coding procedure. This becomes a big drawback for realizing the algorithm on hardware platform. In this paper, a Modified SPIHT (MSPIHT) method suitable for image compression, an improved version of original SPIHT is also discussed.**

Keywords-**image compression, MSPIHT, SPIHT**

I. INTRODUCTION

Recently the massive use of digital images generates increasingly significant volumes of data. Compressing these digital images is thus necessary in order to store them and simplify their transmission. Wavelet-based coding provides substantial improvements in image quality at higher compression ratios. Over the past few years, several very competitive wavelet based image compression algorithms with an embedded bit stream have been developed, such as Shapiro's embedded zerotree wavelet compression (EZW) algorithm[1], Said and Pearlman's set partitioning in hierarchical trees (SPIHT) algorithm[2], and Taubman's embedded block coding with optimized truncation (EBCOT) algorithm.

SPIHT is an improved version of EZW .It improves the coding performance by exploiting the self-similarity of the coefficients across sub-bands more efficiently than EZW. Although, SPIHT is well known for its simplicity and efficiency, it requires a large amount of memory to maintain three lists that are used for storing the coordinates of wavelet coefficients and tree sets in the coding and decoding processes.

The SPIHT algorithm is unique in that it does not directly transmit the contents of the coefficient tree sets, the pixel values, or the pixel coordinates [4]. 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. We use SPIHT algorithm for image compression and discuss the results in this paper.

In the original SPIHT algorithm, after a wavelet decomposition to the image, most of its energy amasses at the low-frequency subband, i.e., most of the big magnitude wavelet coefficients amass at the low-frequency subband, while the coefficients in high-frequency subbands are usually small and their numbers are huge, making LIP and LIS expand quickly and decrease slowly, therefore wasting time and space. A great number of operations to manipulate the memory are also required in the codec scheme, which greatly reduces the speed of coding procedure. This becomes a big drawback for realizing the algorithm on hardware platform in real-time and low memory applications. So a high-speed and low memory image compression algorithm is desired.

MSPIHT is the low memory solution to SPIHT algorithm. This algorithm uses two state mark bitmaps to replace three lists of SPIHT, so the coordinates of the coefficients are not stored in the high, variable and data dependant lists, the MSPIHT becomes a low-memory solution to SPIHT algorithm. Furthermore, in order to enhance the algorithm's iteration efficiency, the MSPIHT also uses two strategies: firstly, the number of error bit is proposed and used to merge the sorting pass and refinement pass, the number of error bit indicates the number of bits that will be omitted finally, in implementation, if a coefficient found to be significant, its last error bits will be omitted and the rest of the bits will be outputted directly, this improved the algorithm's implement efficiency; secondly, an array is used to store the maximum value coefficient of every zerotree sets, when a zerotree set is going to be analyzed, the related value in the array is first compared with the scan threshold, if the value is smaller than the threshold, directly pass this zerotree analysis and go to next one, this greatly reduces the comparing and judging times, especially in low bit rate applications.

II. METHODOLOGY

Wavelet transform provides a compact multi-resolution representation of the image [3]. It has excellent energy compaction property which suitable for exploiting redundancy in an image to achieve compression.

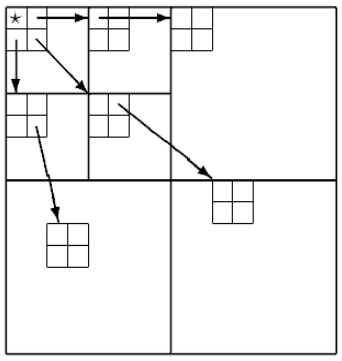

Fig. 1 .Spatial Orientation Tree.

A.SPIHT Coder

The success of the SPIHT is in part due to the organization of the wavelet coefficients in to the spatial orientation trees as showed in Fig.1.The SPIHT coder uses three types of sets: D(i,j) denoting the set of all descendants of a node (i,j), O(i,j) representing the set of all offspring of the node (i,j), and L(i,j) representing the set of all descendants excluding the immediate four offspring of the node (i,j), that is, L(i,j) =D(i,j)–O(i,j). They are represented as C(i,j). C(i,j) is called significant with respect to given threshold T(T=2^n ,the initial n=$\lfloor\lfloor \log_2 C_{max} \rfloor\rfloor$) if C(i,j)≥T, otherwise, it is called insignificant. To indicate the significance of a set U, we use the following notation: S_n (U) = 1, if max{C(i , j)}≥ 2^n, (i,j)€U, otherwise, S_n(U)=0. In practical implementation, the important information is stored in three ordered lists: a list of insignificance sets (LIS), a list of insignificance pixels (LIP), and a list of significant pixels (LSP). In all lists, each entry is identified by a coordinate (i, j), which represents individual pixels in the LIP and LSP, and indicates either the set D (i,j) (a type A set) or L(i,j)(a type B set) in the LIS. At the initialization step, the step, coefficients in the highest level are added to LIP, and only those with descendants are added to LIS as type A entries. The LSP is set as an empty list. The SPIHT coder starts with the most significant bit plane. At every bit plane, it tests the three lists in order, starting with LIP, followed by LIS and LSP. The coefficients in the LIP are coded firstly, when a coefficient in LIP becomes significant it is moved to the end of the LSP and their signs are coded. Similarly, sets D (i,j) and L(i,j) are sequentially coded following the LIS order, and those that become significant are partitioned into subsets. The set partitioning rules are as follows: (1) if D (i,j) is significant, it is partitioned into L(i, j) plus four single elements sets with (k, l) € O(i, j) . (2) if L(i,j) is significant, it is partitioned in to four sets D(k, l) with (k, l) € O(i, j) .Newly formed set L(i,j) and D(k,l) are ended to the end of LIS to be coded again before the same sorting pass ends. Finally, each coefficient in LSP except the ones added in the last sorting pass is refined in each refinement pass. The algorithm then repeats the above procedure for the next resolution, and then stopped at desired bit rates. The decoding algorithm can be obtained by duplicating the encoder's execution path.

B. *Modified SPIHT Description*

In SPIHT, the use of three temporary lists is powerful way to improve the codec's efficiency. However, they are quite memory consuming, and in addition, during coding, we often insert or delete the elements in the lists. These frequent operations also greatly increase the coding time with the expansion of the lists [6]. All of this makes the original SPIHT algorithm very complex that becomes difficult for hardware implementation, especially in low memory and real-time situations. In order to realize the implementation of the SPIHT in real-time and low memory situations on hardware platform, a successful fast, low memory and simple method must be provided. In this algorithm, the sorting pass and refinement pass are combined as one scan pass. Moreover, in the MSPIHT [5], the coordinates of wavelet coefficients with important information are never stored in the LSP, LIP or LIS. There are no such lists in the algorithm. We use two state mark bitmaps, a maximum value array and number of array bits to modify the original SPIHT algorithm.

The information of insignificant pixels and insignificant sets are not stored in LIP and LIS anymore, but in two corresponding state mark bitmaps respectively: called state mark bitmap of insignificant pixel (SMBIP) and state mark bitmap of insignificant set (SMBIS) respectively.

At the algorithm's initialization period, SPIHT need to travel all the wavelet coefficients to find the maximum value of the coefficients. For successive image coding methods, their coding process is a gradual scanning procedure along with the decrease of threshold. At every scanning pass, an effective binary bit of significant coefficient is outputted, when the scanning procedure achieve the required bit rate, the procedure stops, and significant coefficients' low bits are ignored. During implementation, when a wavelet coefficient found to be significant, its last error bits will be omitted and the rest of the bit is outputted directly. Using this way, the MSPIHT combines the sorting and refinement pass, accordingly the information of significant coefficients' position does not need to be stored for further process. Besides memory saving, this also reduces the scanning time, especially in low-bit rate situations.

III. RESULTS

Table I & Table II lists the experimental results SPIHT based image compression codec. Table I lists the experiment results of encoding on 256X256 gray-scale image Lena, using eight-level wavelet decomposition. It is a good idea to try various wavelet families and see their influence. Results confirm what has been expected; generally introduced fact that bior 4.4 filter bank produces best results in image compression. In the next step influence of level settings were discovered using group of images. We simply measured the time (in seconds) necessary to decode the image. At level 1 some artifacts are seen giving poor PSNR as shown in Fig.5.Contrastly,at level 5 we get good PSNR and no image artifacts as shown in Fig. 3.

A) Tables and Figures.

TABLE I
OBJECTIVE ANALYSIS OF SPIHT CODEC FOR VARIOUS WAVELET FILTERS

Wavelet	lena.bmp	
	PSNR(dB)	Time elapsed (sec) (decoding)
haar	30.04	1.906
db7	29.68	2.047
bior1.1	30.04	1.937
bior3.7	28.79	1.765
bior4.4	31.56	2.032
bior5.5	30.80	2.156

TABLE II
COMPRESSION TESTING AT DIFFERENT RESOLUTION LEVELS FOR PROGRESSIVE CODING

Level	PSNR(dB)		
	Cameraman	Rice	Circles
1	5.58	9.88	7.07
2	20.29	20.79	17.51
3	30.54	29.96	30.16
4	34.61	30.00	41.35
5	31.36	31.55	41.23
6	31.52	31.65	41.69
7	31.55	31.67	41.79
8	31.56	31.68	41.80

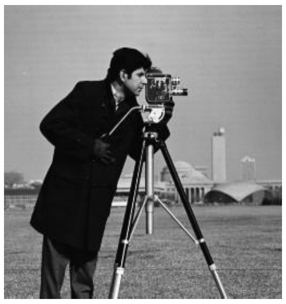
Fig. 2. Original Image
Cameraman

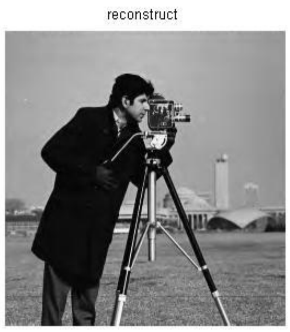

Fig. 3: Recovered image at 5th resolution level

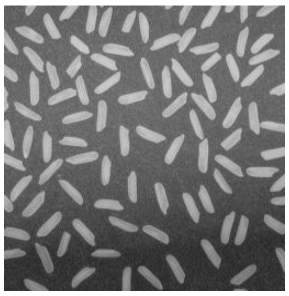

Fig. 4 Original Image
Rice

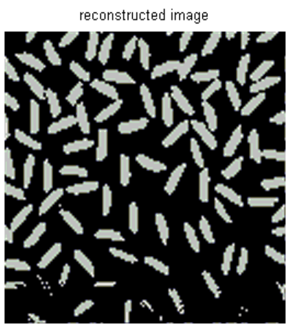

Fig. 5 Recovered image at **1st level**

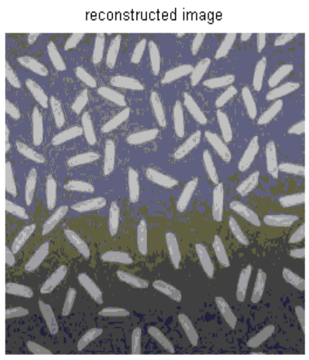

Fig. 6 Recovered image at **6th level**

IV. CONCLUSION

The SPIHT codec was implemented in MATLAB environment. SPIHT is a simple and efficient algorithm with many unique and desirable properties. The outstanding efficiency of the DWT-SPIHT image compression codec has been demonstrated in the terms of PSNR and decoding time. The simulations confirm the proposed ideas of proper wavelet filters usage. The key to the algorithm is to properly and efficiently sort sets of transform coefficients according to their maximum magnitude with respect to a sequence of declining thresholds. Theoretical analysis shows that the MSPIHT codec is improved version of SPIHT codec.

REFERENCES

[1] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[2] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250,June 1996.

[3] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[4]A.V. Nandi and Dr. R. M. Banakar “Hardware modeling and implementation of modified SPIHT algorithm for compression of images,” *Second International Workshop on Industrial and Information Systems,* August 2007

[5]Jianjun Wang, “Modified SPIHT Based Image Compression Algorithm for Hardware Implementation” *IEEE Second International Workshop on Computer Science and Engineering*, 2009.

[6]Yin-hua Wu, “An improved fast parallel SPIHT algorithm and its FPGA implementation,” *IEEE Second International Workshop on Future of Computer and Communication,* 2010.