ENHANCED PARTIAL-SPIHT FOR LOSSLESS AND LOSSY IMAGE COMPRESSION

Ahmed Abu-Hajar and Ravi Sankar
Department of Electrical Engineering
University of South Florida
Tampa, FL 33620
{abuhajar,sankar@eng.usf.edu}

ABSTRACT

This paper proposes an enhanced Partial- SPIHT (P-SPIHT) for lossless and lossy image compression. P-SPIHT uses three coding modes for each bit plane based of the probability of the significant coefficients (P_1) within each bit plane as proposed in [1]. In this paper, P-SPIHT is extended to support both lossy and lossless compressions. Also it sorts the coded data into three categories; sign bits (*SB*), tree bits (*TB*) and magnitude bits (*MB*). P-SPIHT sorts *TB* and *MB* into insignificant tree bits (*ITB)*, significant tree bits (*STB)*, insignificant magnitude bits (*IMB)* and significant magnitude bits (*SMB).* Sorting the data enhances the compression of the arithmetic coder, as each category uses its own frequency model within the coder. Sorting the data improves the compression of the arithmetic coder by a factor of two. Experimental results show that the compression of P-SPIHT is superior to SPIHT for all the tested images and it is comparable JPEG2000 especially in the lossless mode.

1. INTRODUCTION

Several wavelet image compression algorithms have been proposed to in the literature. SPIHT is considered to be one of the most popular wavelet image compression algorithms [2-4]. A simple block diagram of SPIHT is shown in Fig.1 which transforms the image using wavelet. Then, the coefficients in the transform domain are quantized and coded bit plane by bit plane. Arithmetic coder is used to remove the remaining redundant information. SPIHT uses zero tree coding scheme to code each bit plane [2]. SPIHT segments each subband into 2x2 sets in the transform domain. The 2x2 sets in the higher levels relate to the ones in the lower levels in a descendant order as shown in Fig.2. SPIHT codes one bit that reads “0” if all the descendants of a set are insignificant; otherwise, SPIHT codes one bit that reads “1” followed by four bits of the 2x2 set. The sign bit is coded after coding the coefficient for the first time.

The efficiency of SPIHT in terms of compression in *bpp* depends on the probability of the significant coefficients (P_1) within each bit plane. The compression of SPIHT decreases as the probability of the significant coefficients within the bit plane (P_1) increases [1]. P_1 increases in the lower bit planes for two reasons:

- The magnitudes of the high frequency coefficients are low, and they appear more frequent in the lower bit planes.
- The coefficients are likely to be significant once they are coded significant at least once.

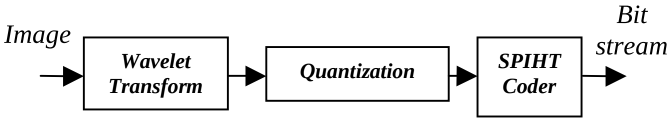

Fig.1. Block Diagram of SPIHT

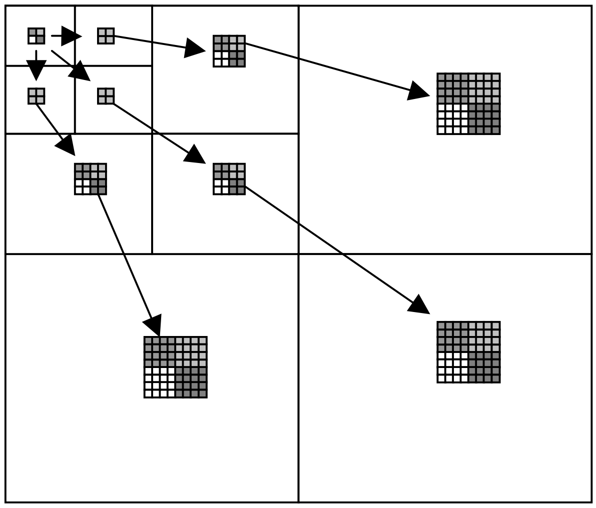

Fig.2 Spatial Zero Tree

To improve the coding compression, we have proposed Partial-SPIHT (P-SPIHT) which uses predetermined values of P_1 to select a particular scheme to code the image losslessly.

This paper is organized as follows: Section 2 presents review of P-SPIHT, and Section 3 presents sorting the data of P-SPIHT. Section 4 presents the development of the lossy mode P-SPIHT, and Section 5 presents the proposed P-SPIHT. Section 6 presents the experimental results and discussion. Finally Section 7 presents the conclusion of this paper.

2. REVIEW OF P-SPIHT

P-SPIHT is designed to code the image losslessly using three coding modes. First, SPIHT evaluates the probability of the significant coefficients, then P-SPIHT selects one of the proposed modes listed in the following:

- Mode_1: This mode is selected SPIHT as long as the probability of the significant pixels is less that 0.2.
- Mode_2: This mode is selected if the probability of the significant pixels is between 0.2-0.3. Mode_2 codes the 2x2 sets and it ignores the spatial correlation among the descendant coefficients. If all the 2x2 sets are insignificant it transmits "0" else it transmits "1" followed by the four bits of the set.
- Mode_3: This mode is selected if the probability of the significant pixels is less than 0.3. It codes the four bits of the 2x2 set without checking for the significance condition.

In this paper, P-SPIHT is enhanced by extending P-SPIHT to the lossy mode. But at low bit rates, P-SPIHT would select Mode_1 and the overall compression would be the same as SPIHT. However, compression may be improved if the performance of the arithmetic coder is improved. We are proposing here to sort the coded data so that the performance of the arithmetic coder can improved.

3. SORTING THE DATA OF P-SPIHT

Adaptive arithmetic coder is used in P-SPIHT to remove the remaining redundancy in the compressed data. However, the performance of the arithmetic coder as the one used in [5] is not optimum for SPIHT or P-SPIHT, because the probability of the coded symbols are near uniform and are uncorrelated. In general, the performance of the arithmetic coder, or any entropy coder, improves if few symbols have very high probabilities and the remaining symbols have very low probabilities. In this case, the entropy of the coded data is low and the performance of the arithmetic coder is relatively higher.

Here, we are proposing to code the data such that the symbols have higher correlation and the fewer higher peaks. This is achieved by sorting the coded data into categories such that each category is coded separately using its own frequency model within the arithmetic coder. The data are sorted into three main categories and they are listed as:

1. *Magnitude Bits (MB)*: This category contains the magnitude of the coefficients only.
2. *Tree Bits (TB)*: This category contains the bits responsible for checking the significance of each set.
3. *Sign Bits (SB)*: This category contains the sign of the coefficient. The sign bit is coded once after coding its magnitude bit as a significant bit for the first time.

Further, *TB* is sorted into two subcategories each, which are called the *Significant Magnitude Bits (SMB)* and *Insignificant Magnitude Bits (IMB)*, and *TB* is sorted into *Significant Tree Bits (STB)* and the *Significant Tree Bits (STB)*. Dividing the data into these subcategories is based on the observation that *TB* data are highly correlated within each bit plane. The correlation increases by dividing the *TB* into *STB* and *ITB*. This is because the sets that were found significant in earlier bit planes are highly likely to be significant in later bit planes. Also, the *MB* is separated into *IMB* and *SMB* based on the observation the coefficient it is highly likely to be significant in bit planes, once it was coded to be significant at earlier bit planes.

Sorting the data depends on the coding mode of P-SPIHT. If Mode_1 or Mode_2 are selected then the coded data are categorized into IMB, SMB, ITB, STB and SB. On the other hand, if Mode_3 is selected the coded data are categorized into IMB, SMB and SB. Mode_3 omits all the TB because it does not code the significant of the set..

The categorized data are coded in an embedded manner as shown Fig.3 as the whole image is coded bit plane by bit plane, and each bit plane is coded slot by slot. The slots are coded in a sequential order of ITB, STB, IMB, SMB then SB. Each slot contains data and End of Slot Symbol (ESS). The ESS synchronizes the information which indicated the end of coding a particular category within a bit plane.

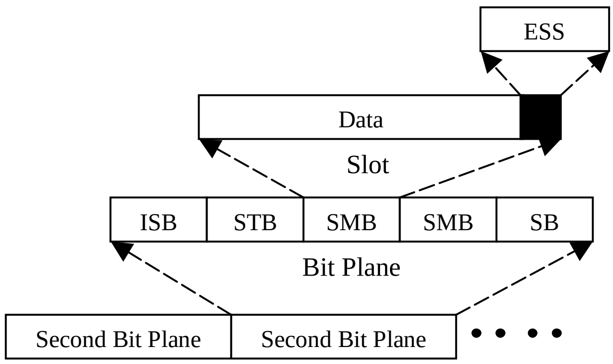

Fig. 3 Architecture of embedding the categorized data

4. EXTENDING P-SPIHT TO THE LOSSY MODE

P-SPIHT is capable of lossy as well as losslessly coding, and there are some differences between the two modes. The first one is that each mode uses different filter, as the lossless mode uses the reversible 5/3 filter and the lossy mode uses 9/7 filter. The second difference is that lossy mode normalizes the coefficients while lossless mode omits the normalization step. The normalization step multiplies the output of lowpass and the high pass filters of the analysis section of the filter bank by two constants k_1 and k_2, respectively. The normalization step approximates the transform to be orthonormal which means that all the coefficients in the same bit plane have the same energy. Ideally k_1 and k_2 are equal to $\sqrt{2}$, but they are approximated to be $k_1=2$ and $k_2=1$ [5]. If the image is decomposed into N levels, then all the coefficients in LL_N

subband are multiplied by 2^{N+1} and all the coefficients in the HL_j and the LH_j are multiplied by 2^N and all the coefficients in the HH_j are multiplied by 2^{N-1} where $j=1,2,\ldots N$.

5. THE PROPOSED ALGORITHM

Fig. 4 shows the overall proposed P-SPIHT coder in which the image is transformed using the reversible 5/3 wavelet filter for the lossless mode and the 9/7 wavlet filter for the lossy mode. The lossy mode normalizes the coefficient so that all the coefficients in all the subbands within a particular bit plane have the same energy. The normalization step is used for the lossy coding only and the lossless mode omits the normalization step. Then, the coefficients are quantized on a bit plane basis.

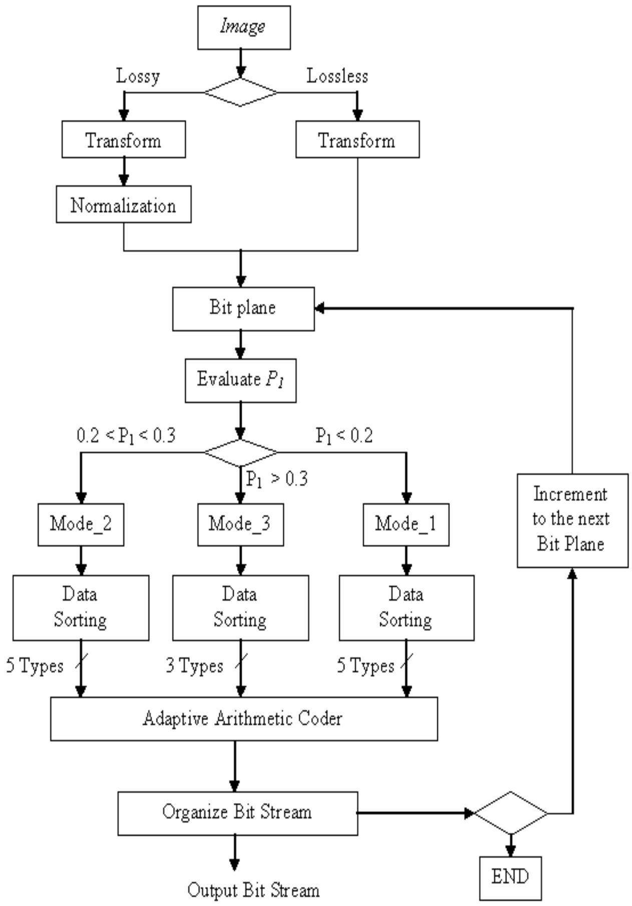

Fig.4 The overall proposed P-SPIHT

P_SPIHT evaluates P_1 of each bit plane indpendently then determines the coding mode to code each bit plane. Mode_1 is selected if $P_1<0.2$, or Mode_2 is selected if $0.2 \le P_1 < 0.3$, and finially Mode_3 is selected if $P_1 \geq 0.3$. Once the mode is selected, the coded data are sorted into *ITB*,*STB*, *IMB*,*SMB* and *SB*. Mode three does not code *ITB* and *STB* because it does not check for the significance condition. Finally sorted data are orginaized as in Fig. 3. If all the coefficients are transmitted or the coder reached the maximum bandwith or the quality of the image, the algorithm would end the process; otherwise, the coder would increment to the next bit plane.

6. EXPERIMENTAL RESULTS AND DISCUSSION

Two sets of images are used to evaluate the compression of the proposed P-SPIHT and the results are compared with SPIHT and JPEG2000. The first set uses ten 8-bit images and the second set uses fifteen 16-bit images. In addition, four versions of P-SPIHT are evaluated which are: P-SPIHT_1 which uses the three modes of the data but it does not sort the data, and P-SPIHT_2 which sorts the data but each data category uses the same frequency model through the entire coding process, and P-SPIHT_3 which resets the frequency model for each bit plane, and finally P-SPIHT_4 which is the same as P_SPIHT_3 but it uses 13/7T reversible integer transform instead of the 5/3 filter, or 9/7 filter in the lossy mode.

The results of the lossless compression in *bpp* of P-SPIHT are superior to SPIHT for all the tested images. Not only P-SPIHT_4 was superior to all the P-SPIHT, but also it was comparable to the JPEG2000. The compression using P-SPIHT_4 exceeds JPEG2000 for some of the 8-bit images. In addition, compression using P-SPIHT exceeds JPEG2000 for all the tested 16-bit images. This makes P-SPIHT the best wavelet-based lossless image compression for 16-bit images. Table.1 summarizes the average lossless compression in *bpp*.

Table 2 summarizes the PSNR of different algorithms at various bit rates for the 8-bit images. The results show that P-SPIHT is superior to SPIHT at all bit rates and it is comparable to JPEG2000. The same results are attained for the 16-bit images. Table 3 summarizes the PSNR of the 16-bit images at various bit rates.

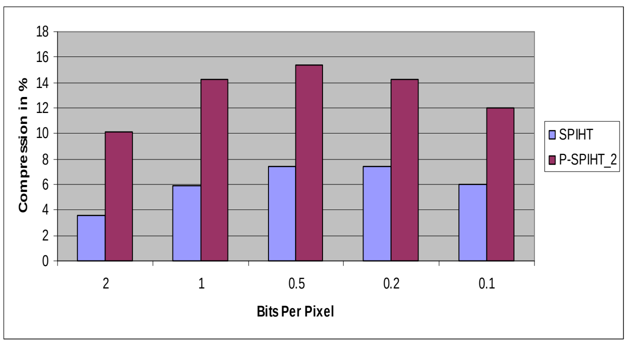

Fig 5 Compression in percent caused by the arithmetic coder for 8-bit images

Sorting the data causes a great improvement of P-SPIHT over SPIHT. Fig. 5 illustrates the compression in % at various bit rates caused by the arithmetic coder. Fig.5 shows that the compression of the arithmetic coder increased by a factor of 2 at all bit rates. Fig. 6 shows similar results for the 16-bit images.

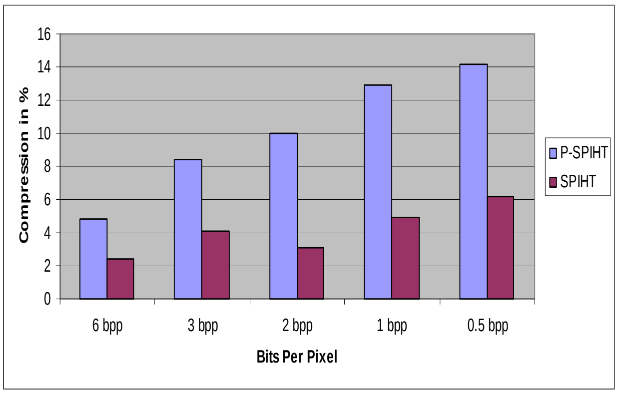

Fig. 5 The compression in Percent caused by arithmetic coder for 16-bit images

7. CONCLUSION

In this paper, an enhanced version of P-SPIHT is proposed. The proposed algorithm using three coding modes similar to the one proposed in [1]. The proposed P-SPIHT sorts the coded data into categories which enhances the performance of the arithmetic coder by a factor of two. In addition, P-SPIHT is extended to support lossy compression as well. The results show that P-SPIHT is superior to SPIHT for all the tested images. Also, P-SPIHT is found to be superior to the lossless JPEG2000 for all 16-bit images, and it is comparable to JPEG2000 for the 8-bit images. The lossy mode of P-SPIHT is superior to SPIHT and it is comparable to JPEG-2000.

REFERENCES

[1] A. Abu-Hajar and R. Sankar, "Wavelet Based Lossless Image Compression Using Partial SPIHT and Bit Plane Based Arithmetic Coder," ICASSP 2002, Vol. 4, pp. 3497-3500, Orlando, Florida, 2002

[2] [9] M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelets Coefficients," IEEE Trans. Signal Processing, Vol. 41, pp. 3445-3462, Dec. 1993.

[3] A. Said and W.A. Pearlman," A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. on Circuit and Systems for Video Technology, Vol. 6, No. 3, June 1996.

[4] A. Said and A. Pearlman, "An Image Multiresolution Representation for Lossless and Lossy Compression," IEEE Trans. Image Processing, Vol. 5,No.9, pp.243-250, Sep. 1996.

[5] I. Witten, R. Neal and J. Cleary, "Arithmetic Coding for Data Compression," Communication ACM, Vol. 30, No. 6 pp. 520-540, June 1987.

[6] JPEG2000 Part I: Final Draft International Standard (ISO/IEC FDISI 5444-1), ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.

Table 1 Lossless Image Compression in bpp on the Average

Image	*SPIHT*	*P-SPIHT_1*	*P-SPIHT_2*	*P-SPIHT_3*	*P-SPIHT_4*	*JPEG2000*
8-bit Set	5.2858	4.8752	4.792	4.775	4.7454	4.74196
16-bit Set	10.55807	9.6835	9.465913	9.4626	9.4632	9.6609

Table 2 PSNR at Different Bit Rates for 8-Bit Images

Rate	*SPIHT*	*P-SPIHT_2*	*JPEG2000*
2.0	39.108267	39.841256	40.869171
1.0	34.868311	35.459456	36.316183
0.5	31.716822	32.362089	32.848599
0.2	28.102156	28.487033	29.090472
0.1	26.066144	26.350722	26.730983

Table 3 PSNR at Different Bit Rates for 16-Bit Images

Rate	*SPIHT*	*P-SPIHT_2*	*JPEG2000*
6	69.18103	73.82173	-
3	55.96763	59.05263	59.97
2	53.28655	54.47028	55.68238
1	49.25195	49.6945	50.73325
0.5	46.56175	46.6425	47.51543