VERY LOW BIT RATE COLOR IMAGE COMPRESSION BY USING STACK-RUN-END CODING*

Min-Jen Tsai
Institute of Information Management, National Chiao Tung University, Hsin-Chu, 300
Taiwan, Republic of China
E-mail: mjtsai@cc.nctu.edu.tw

ABSTRACT

A low complexity wavelet based image coding algorithm for color image compression is presented in this paper. The key renovation of this algorithm is that a small number of symbol set was designed to convert the information from the wavelet transform domain into a compact data structure for each subband. The scheme works first by color space conversion, followed by uniform scalar quantization and data conversion where raster scanning order is performed for individual subband and quantized coefficients are converted into the symbol stream according to the designed symbol representation. Two different context lists in each subband are specified from the symbol stream and alternatively compressed by adaptive arithmetic coder with high efficiency. Unlike zerotree coding or its variations which utilize the intersubband relationship into its own data representation, our work is a low complexity intrasubband based coding method which only addresses the information within the subband. The only extension is the termination symbols which carry the zero value information towards the end of the subband or across the subbands till the end of the image. Compared with the zerotree refined schemes, this algorithm results in competitive PSNR values and perceptually high quality images at the same compression ratio for color image compression.

1. INTRODUCTION

Color image compression is an important technique to reduce the communication bandwidth consumption, especially for congested network like Internet or low bandwidth communication for wireless multimedia transmission. The techniques could also benefit the applications for storage and archiving purposes as well. The popularly used JPEG [1] is the current standard which is based on the block discrete cosine transform with run-length encoding procedures. When high compression ratio is desired, the images have been highly degraded because of the significant block artifacts which result unacceptable images in certain applications.

Wavelet transform [2][3] based image compression algorithms [4][5][6] have achieved good compression performance and been expected to be the core technique for the next generation image communication standard [7]. Among all the approaches, Dr. Shapiro's zerotree [4] data structure has been widely used and extended to different variations and refinements [6][8][9]. Further improvement based on rate-distortion optimization [8] or statistical analysis [9] is also developed under zerotree structure. In essence, they are intersubband based approaches where intersubband relationship has been explored from the location similarities and the ancestor and children dependency prediction. Contrast to the zerotree approach, this new algorithm is an intrasubband based scheme where only information within the subband is needed. The only exception is that the termination symbol is designed to represent the information across the subbands.

The algorithm is conceptually simple without addressing the relationship across the subbands and easily to implement because of the small number of symbol set. Beyond its simplicity, its performance is significantly better than JPEG standard and very competitive with the zerotree refinements. This paper is organized as follows. In Section 2, we explain the Stack-Run-End (SRE) compression algorithm in detail. In Section 3, we summarize the experiment results and outline its characteristics with discussion. Finally, we conclude our renovation with summary.

2. THE ALGORITHM

Wavelet based color image compression includes following stages: color space conversion, wavelet transform, quantization, data conversion and entropy coding. Stack-Run-End (SRE) compression basically covers the quantization, data conversion and entropy coding procedures.

The selection of the working color domain will be discussed in Sec 3. for the color space conversion. After the color space conversion and before the application of the stack-run-end coding, it is free of choices of the decomposition structures for wavelet transform or so called wavelet packets[10] to create the information in the wavelet domain. Generally speaking, the hierarchical decomposition structure of the wavelet transform and uniform scalar quantization in each subband are the most

*The material included in this paper was partially presented at the ICASSP98 and this work was supported by the National Science Council in Taiwan, Republic of China, under Grant NSC88-2416-H009-021 and NSC89-2416-H009-015.

Contributed Paper
Original manuscript received March 28, 2000

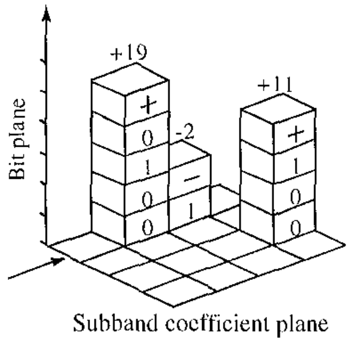


Figure 1. Illustration of the transform coefficients from he perspective view. The raster scanning follows the arrow direction. The values of the significants are +19, -2 and +11. The run values are 1, 0 and 8 with one.

commonly performed operations before the application of the proposed algorithm. With slight modification, stack-run-end algorithm could be implemented for different wavelet decomposition format and different scalar quantization approach as well.

Based on the observation that the quantized transform coefficients are either zero values or nonzero values (called significants) with either positive or negative sign, efficient grouping scheme and representation for those meaningful information are necessary during the data conversion procedure. Our approach is different from JPEG's run grouping and similar to the more advanced approach "Stack-Run Coding" [11][12]. The refinement of Stack-Run-End coding in this color image compression extends the original stack-run symbol set by the consideration of grouping zero value coefficients into the termination symbol representation, especially near the end of the subband. Since the termination symbols perform the "Ending" action which terminates the information accumulation within the subband or the image, it is the reason that we suffix "End" with "Stack-Run" for the naming and distinguish their functions in the following.

To make a concise introduction of our algorithm, an example in Figure 1 illustrates the (stack, run, end) concept and Figure 2 shows the mapping relationship for stack and run. From Figure 1, only three nonzero transform coefficients with integer values exist after the uniform quantization. The nonzero coefficients are called "stack" and the zero values between the stacks are grouped as "run" value. For the zero values towards the end of the subband after the last significant are specified by the representation of the "end-of-subband" symbol. For certain situation that many high frequency subbands contain no significant coefficients, several consecutive "end-of-subband" symbols towards the end of the image could be simplified by an "end-of-image" symbol for the encoding.

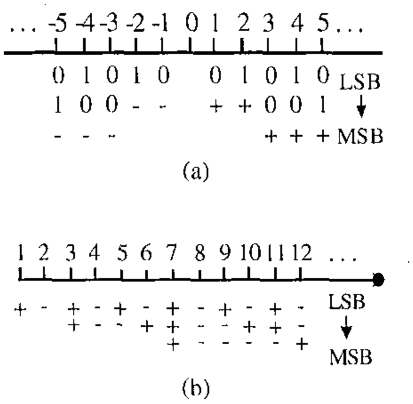


Figure 2. Mapping rule for (a) stack (b) run values

The symbol alphabets in our context for the "stack" value are denoted as the following:

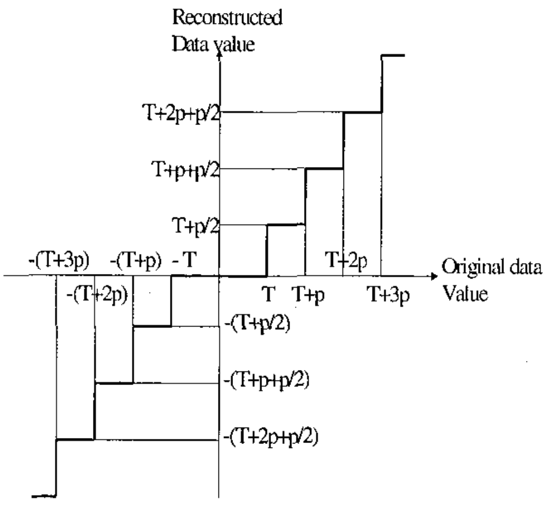
- "+": the binary value 1 with the sign value "+", always used in the stack's MSB.
- "-": the binary value 1 with the sign value "-", always used in the stack's MSB.
- "1": the binary value 1 of the significant coefficient.
- "0": the binary value 0 of the significant coefficient.

Two symbols "+" and "-" are used to represent the "run" value. The definitions are:

- "+": the binary value 1 of the run value.
- "-": the binary value 0 of the run value.

The exception for the run value located at the end of the subband is used by the termination symbol "EOB". Several consecutive EOBs towards the end of the image will be represented by another termination symbol "EOI". They are summarized as the following:

- "EOB": the symbol for those zero values which are between the last nonzero significant coefficient and the end of the subband.
- "EOI": the symbol for those zero values which are between the last nonzero significant coefficient and the end of the image.



(a)

Original Value	Quantized value	Symbol Mapping (LSB->MSB)	Decimal Value
[T,T+p)	1	0+	2
[T+p,T+2p)	2	1+	3
[T+2p,T+3p)	3	00+	4
......	......	......	......

(b)

Figure 3. (a) The relationship between the transform coefficients and the reconstructed values. T is the threshold and p is the stepsize.
(b) Several positive coefficient mappings are listed. Decimal value is acquired by assuming "+" is the binary value 1. The mapping relationship is the same for negative value by substituting "+" by "- ".

From Figure 3, it shows the relationship between the original transform coefficients and quantization value. "T" is the threshold and "p" is the stepsize. The value of threshold is usually bigger than the stepsize value in order to have wider deadzone which usually depends on the fidelity preservation and bandwidth constraints. Transformed coefficient will be encoded only if the value is greater than the threshold. Truncation is basically performed during the scalar quantization. For values between "T" and "T+p", the quantized value is 1 with actual value "T". From the symbol mapping at Figure 2 and Figure 3, "0+" is constructed because there is a need for the binary 1 (labeled as "+") for the threshold and binary 0 for the step size to symbolize the actual value. Since the decimal value 2 is the initial value for its value representation, the rest of the quantized value will also need this shift for the symbol conversion.

So, the significants +19, -2, +11 are represented as 20, 3, 12 decimal value with the sign information on the top of the stacks. For example, +19 could be labeled as 00101 (from LSB to MSB order) in the binary representation. With the positive sign information at the MSB, the total representation for +19 is "0010+". The representation for the negative value is the same for "-2" as labeled "1-".

We have noticed at Figure 2 that the run values are always positive integers. To avoid the confusion in context representation, the MSB "+" of the run binary representation would be redundant except for the value equivalent to 2^k-1 where k is an integer. This is a very important observation to further facilitate the compression efficiency and reduce the number of the symbol representation.

From the initial scanning point, there is a zero value before "+19", the run value is 1. Since there is no zero value between "+19" and "-2", the run value 0 is not encoded. Another run value "8" exists between the "-2" and "+11". After the significant value "+11", four zero values are left till the end of the scanning. If this is the end of the subband, an EOB symbol is used which is regardless of the run value. If the situation is no further significant values towards the end of the image, an EOI symbol is used. Assuming this is the end of the subband, the whole symbol stream could be represented as "+0010+1----001+EOB". At this point, we already successfully convert the meaningful information in the subband into a more concise data representation. All the conversions are based on the information for either stack, run or termination symbols.

Since the information created from above example is either stack, run or the termination symbols, it has been examined that directly applying one entropy coding scheme for the whole symbol stream could not efficiently compress those symbols. To utilize the information created from different types of data with different symbol distributions, a partition of the symbol stream for the best entropy coding is necessary. The "location list" and "stack list" are generated after the partition according to the following rules:

- Stack list is the list of stack symbols ordered sequentially from LSB to MSB. However, the LSB of the stack is not included.
- Location list is the list of the run values with the LSB of the adjacent stacks and the termination symbols EOB and EOI.

According to the above rules, the symbol stream of the example in Figure 1 will be easily separated from "+0010+1----001+EOB" into the stack list with the stream "010+-01+" and the location list with the stream "+01---0EOB" respectively. After this separation, a more compact stack and run values are closely related within the similar context list. Since the symbol is well defined and uniquely distinguishable with the prior information about the image size and the decomposition structure at both encoder and decoder, there is no confusion about the information presentation.

A zeroth order adaptive arithmetic coder [13] is applied to further compress those two lists for each subband. The coder's counter of the symbol appearance is always reset whenever a new location or stack list occurs. The coder itself is very simple and the adaptability is very fast given the small setting of the maximum frequency for the symbol appearance. It also considers the local variation of the probability distribution to adjust the frequency order. Alternately encoding the stack list and location list for the stack-run-end compression, a subband based embedded stream could be created for the progressive transmission and display.

3. DISCUSSION

A huge amount of color image data have been tested under this new stack-run-end compression approach. In this image set which contains over hundreds of color images ranging from 64x64 to 736x576, we adopt the advice from [14] to exclude some over-used images like Lena and pepper. We try to extend the fields and include more new images with wide varieties which are suitable for Internet communication and wireless transmission or other similar purpose uses. Since the images come from different available resources, we expect they can cover wide area of applications for color image compression evaluation. Even the judgment of compression performance is not mainly based on the numerical metrics like the peak signal noise ratio (PSNR), those numbers are by far the most common used and the easiest way for evaluation comparison. In our experiments, current standard JPEG and the refinement of zerotree scheme from Said & Pearlman [6] (SPIHT) are also compared. To make the fair comparison, both SRE and SPIHT use the 9-tap/7-tap filter [15] in the wavelet transform and perform the hierarchical decomposition. JPEG standard [16] is tested under the environment where the setting is at the optimized mode.

Given the testing environment is the personal computer with Pentium processor under Window NT 4.0, BMP image format with R, G, B channels is the most popular format for PC applications. From statistics, the R, G, B color channels have high correlation relationship. To decorrelate the interrelation, the whole computation has been performed under the YUV color domain to increase the efficiency. After the compression, the reconstructed images are stored in the BMP image format for display and evaluation. The overall color PSNR performance of various target compression ratio at low bit rate from our testing image database are summarized and tabulated at Table 1.

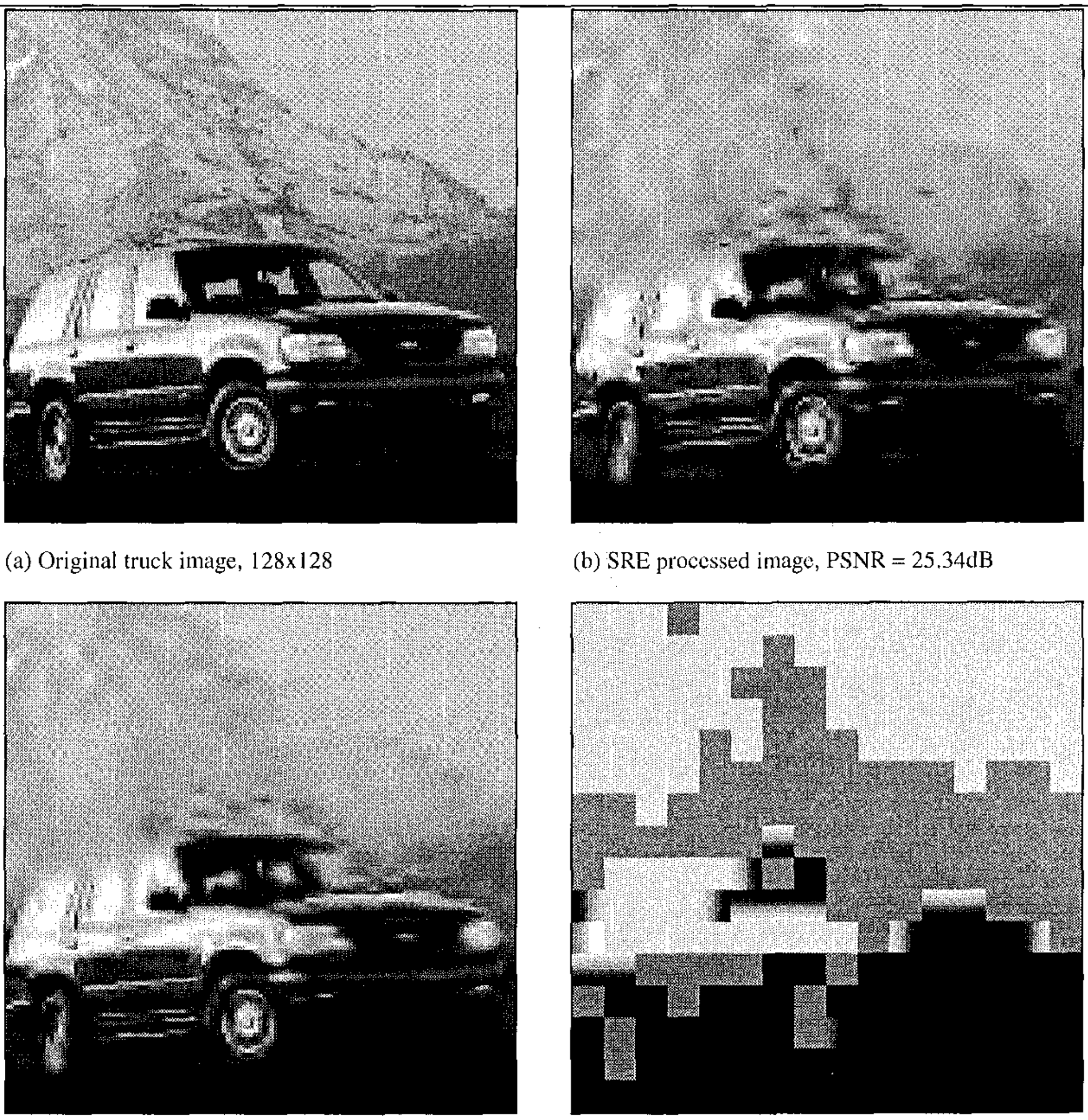
In Table 1, the average color PSNR are calculated for each target bit rate for each tested image. For each compression technique, we get the final PSNR value by summing all the color PSNR numbers and then divided by the number of the images. Even those numbers are the overall results which do not tell the compression performance for each individual image, they still disclose the characteristics of the compression techniques in the numerical sense. We notice that SRE and SPIHT are far superior to JPEG method from the color PSNR metric. The PSNR difference between SR and SPIHT is between a few tenth of a dB variation which is generally negligible.

Besides PSNR metric evaluation, we also requested two experienced technicians to visually compare the reconstructed images and give the perceptual quality score for those three methods at high compression ratio. The conclusion is that JPEG produces significant block artifacts which is the apparent noise from the reconstructed images. SRE and SPIHT have much better image fidelity maintenance and both algorithms are very comparable. The interesting behaviors from the SRE and SPIHT reconstructed images are that SRE has the better capability to maintain the details of the object and SPIHT seems to keep the color distribution better.

From Table 1, it can not distinguish the individual image compression result under different algorithm. We demonstrate an example in Figure 4 and Figure 5 to illustrate the point from the perceptual observation. From the pictures at both Figures, the reconstructed image from JPEG has obvious block artifacts and color distortion. SRE and SPIHT have blurred textures but still maintain the

Algorithm and bit rate	0.25bpp (96:1)	0.5bpp (48:1)	0.75bpp (36:1)	1.0bpp (24:1)
SRE	24.63dB	27.46dB	29.70dB	31.61dB
SPIHT	24.81dB	27.72dB	29.90dB	31.88dB
JPEG	23.36dB	26.18dB	27.76dB	29.01dB

Table 1: Compression results from SRE, S&P and JPEG methods at the bit rate 0.25, 0.5, 0.75 and 1.0 bpp with the compression ratio at 96:1, 48:1, 36:1 and 24:1 respectively. The average of color PSNR from R, G, B channels are calculated over 100 color images.



(a) Original truck image, 128x128

(b) SRE processed image, PSNR = 25.34dB

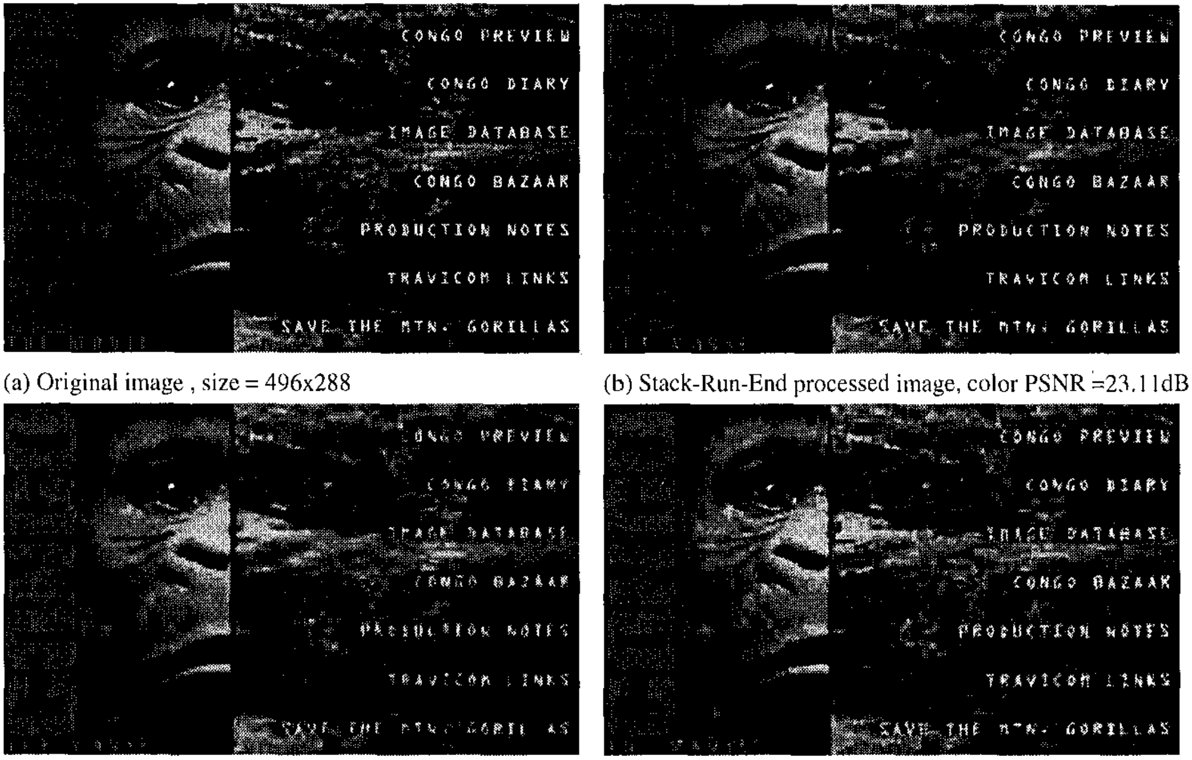
(c) SPIHT processed image, PSNR = 25.61dB

(d) JPEG processed image, PSNR = 17.42dB

Figure 4. Demonstration of the original 24 bit truck image and the reconstructed images by SRE, SPIHT and JPEG methods at the compression ratio 96:1 which is equivalent to 0.25 bits per pixel (bpp).

(a) is the original truck image. (b),(c),(d) are reconstructed image by SRE, SPIHT and JPEG method respectively.

object structure. In Figure 4, JPEG processed image has serious color mismatch and degraded to block based grayscale image since limited available information for each transformed block which could not provide substantial information to recover the object within the whole frame. But the truck features are maintained from SRE and SPIHT and there is basically no noticeable difference between SRE and SPIHT processed images. If we emphasize the text information from the reconstructed images in Figure 5, it is very encouraging that SRE has much stronger edge details than SPIHT and JPEG. This observation agrees with the conclusion from the visual test.



(a) Original image , size = 496x288

(b) Stack-Run-End processed image, color PSNR =23.11dB

(c) S&P processed image, color PSNR = 23.03dB

(d) JPEG processed image, color PSNR = 21.50dB

Figure 5. Demonstration of the original 24 bit congo image and the reconstructed images by SRE, SPIHT and JPEG methods at the compression ratio 96:1 which is equivalent to 0.25 bits per pixel (bpp).

(a) is the original congo image. (b),(c),(d) are reconstructed images by SRE, SPIHT and JPEG method respectively. Color PSNR values for all three techniques are also included in (b), (c) and (d).

There is also a common critique to refer the complexity issue during the algorithm development. Since the SRE codes are not optimized yet, it is still too early to jump to the conclusion about the complexity comparison. In addition, there are many other consideration about the issues in CPU, memory or compiler during the realization. From our preliminary analysis, SRE compression has the similar complexity to the SPIHT method. Both methods are on the same scale of the speed with the potential for improvement, which are about 2~3 times slower than the JPEG method at the current version.

The advantage of the termination symbol design for color image coding in SRE compression is important at least two ways. The first one is that Y channel constructs the most of color content information than U, V channels. EOB and EOI symbols will be more often used for U, V channels which can benefit the reduction of the number of symbols in conversion. The other one is to speed up the reconstruction procedure for the receiver which can spare the minimum waiting time at the decoder. However, it is very flexible for SRE coding if EOB symbol is appended after coding each subband for other purposes (for example, synchronization) or consecutive EOBs are used without EOI symbols. The codecs already have the sufficient knowledge to specify the information at both sender and receiver without any confusion. In addition, it is possible to modify the SRE algorithm with error control capability to generalize its applications for the error prone channel communications.

4. CONCLUSION

In this paper, we introduce a new algorithm: Stack-Run-End compression for color image coding. The technique is an intrasubband wavelet based approach which is different from the zerotree type based schemes. The algorithm uses a small symbol set to convert the meaningful information of the wavelet transform coefficients into a concise data structure during the (stack, run, end) conversion. Two types of context lists for each subband are alternatively compressed by the zeroth order adaptive arithmetic coder with high efficiency. The bit stream has the progressive transmission property since it is organized at the subband order. Our experiment results shows that our approach is very competitive to the refinement of zerotree type schemes. From perceptual viewing test, high detail fidelity maintenance of the color images is achieved by our techniques.

5. REFERENCES

[1] W.B. Pennebaker, J.L. Mitchell, "JPEG - Still Image Data Compression Standard", Van Nostrand Reinhold, New York, 1993.

[2] Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. XLI, pp. 909-996, 1988.

[3] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Pattern Analysis Machine Intell.* 7:674-693, 1989.

[4] M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Trans. On Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[5] A. Zandi, J. Allen, E. Schwartz, M. Boliek, "CREW: Compression with reversible embedded wavelets", *Proc. IEEE Data Compression Conference* 212:221, 1995.

[6] 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, pp. 243-250, June 1996.

[7] ISO/IEC JTC1/SC29/WG1 (ITU-T SG8) Coding of Still Pictures, 1997.

[8] J. Li and S. Lei, "An Embedded Still Image Coder with Rate-Distortion Optimization", *IEEE Trans. Image Processing*, vol. 8, pp. 913-924, July 1999.

[9] R. W. Buccigrossi and E. P. Simoncelli, "Image Compression via Joint Statistical Characterization in the Wavelet Domain", *IEEE Trans. Image Processing*, vol. 8, pp. 1688-1701, December 1999.

[10] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense", *IEEE Trans. Image Processing*, vol. 1, pp. 160-175, February 1993.

[11] M.J. Tsai, J.D. Villasenor and F. Chen, "Stack-Run image coding for low bit rate image communication", *Proc. IEEE International Conference on Image Processing (ICIP'96)*, vol. 1. pp. 681-684, Sep. 16-19, 1996, Lausanne, Switzerland.

[12] M.J. Tsai, J.D. Villasenor and F. Chen, "Stack-Run image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 519-521, Oct. 1996.

[13] T.C. Bell, J.G. Cleary, I.H. Witten, "Text Compression", Prentice Hall, New Jersey, 1990.

[14] D.C. Munson, *IEEE Trans. on Image Processing*, vol. 5, No. 1, pp. 3, January 1996.

[15] M. Antonini,M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. Image Processing*, vol. 1, No. 2, pp. 205-220, Apr. 1992.

[16] Version 6a of the Independent JPEG Group's free JPEG software.

6. BIOGRAPHY

Min-Jen Tsai received the B.S. degree in electrical engineering from National Taiwan University in 1987, the M.S. degree in industrial engineering and operations research from University of California at Berkeley in 1991, the engineer and Ph.D. degrees in Electrical Engineering from University of California at Los Angeles in 1993 and 1996, respectively. He served as a second lieutenant in Taiwan army from 1987 to 1989. From 1996 to 1997, he was a senior researcher at America Online Inc. In 1997, he joined the institute of information management at the National Chiao Tung University in Taiwan and is currently an assistant professor. His research interests include multimedia system and applications, image compression and processing, digital watermarking and authentication, mobile agent and computing for electronic commerce applications.

Dr. Tsai is a member of IEEE, ACM and Eta Kappa Nu.