TABLE III
PERFORMANCE COMPARISON OF PVQ USING THE PROPOSED MLP PREDICTOR AND JPEG FOR THE TEST IMAGE LENA

PVQ		JPEG	
Bit Rate	PSNR	Bit Rate	PSNR
0.181	29.54	0.191	28.25
0.240	30.57	0.233	29.54

the training data. In Table II, the images "crowd" and "man" were taken from inside the training data. As we mentioned earlier, the test image Lena was not used for training; however, it was used as the test image to stop the training process. We also used two additional test images: "Marie" and "couple." These images were *completely* outside the training process. The test image Marie has similar statistics to those images used in the training process; however, the test image couple has quite different statistics from the training data. Table II shows that the neural network predictors performed better than the linear predictor for the test image Marie; whereas the performance of all the predictors was almost the same for the test image couple. This implies that the neural network predictors perform well for the data that has statistics similar to the training data. Therefore, this property of the neural network predictors makes them attractive for designing dedicated predictors for different kinds of data, forming an adaptive system in order to achieve better overall performance. Finally, Table III shows the performance comparison of a PVQ using the proposed MLP predictor and the Joint Photographers Expert Group (JPEG) for the test image Lena at low bit rates. The PVQ not only gives better PSNR but also improves the perceptual quality of the reconstructed image when compared with JPEG.

IV. CONCLUSION

In this correspondence, we presented several neural network architectures for designing the nonlinear predictors, which include the multilayer perceptron, the radial-basis function network, and the functional link network. The neural network approach provides a solution to the problem of designing a nonlinear vector predictor. The neural network predictors are found to have improved performance, in comparison to that of an optimal linear predictor, both numerically and perceptually. We demonstrated that the neural network predictors predict the blocks that contain edges with greater accuracy than a linear predictor. This capability of the neural networks can be better taken advantage of by using a modular approach in which a dedicated predictor is used to learn particular statistics of the data. The use of neural network predictors as expert modules in an adaptive vector prediction scheme is the focus of the future research.

REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer, 1992.
[2] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, pp. 957–971, Aug. 1988.
[3] H. M. Hang and J. W. Woods, "Predictive vector quantization of images," *IEEE Trans. Commun.*, vol. COMM-33, pp. 1208–1219, Nov. 1985.
[4] N. Mohsenian, S. A. Rizvi, and N. M. Nasrabadi, "Predictive vector quantization using a neural network approach," *Opt. Eng.*, vol. 32, pp. 1503–1513, July 1993.
[5] S. A. Rizvi and N. M. Nasrabadi, "Predictive residual vector quantization," *IEEE Trans. Image Processing*, vol. 4, pp. 1482–1495, Nov. 1995.
[6] S. Wang, E. Paksoy, and A Gersho, "Nonlinear prediction of speech with vector quantization," in *Proc. Int. Conf. Spoken Language Processing*, Kobe, Japan, Nov. 1990, pp. 29–32.
[7] N. Tishby, "A dynamical systems approach to speech processing," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Albuquerque, NM, Apr. 1990, pp. 365–368.
[8] S. Dianat, N. M. Nasrabadi, and S. Venkataraman, "A nonlinear predictor for DPCM encoder using an artificial neural network," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Toronto, Ont., Canada, 1991.
[9] C. N. Manikopoulos, "Neural network approach to DPCM system design for image coding," *Proc. Inst. Electr. Eng.*, vol. 139, pp. 501–507, Oct. 1992.
[10] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, vol. 1, 1986.
[11] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.
[12] S. S. Haykin, *Neural Networks: A Comprehensive Foundation.* New York: Macmillan, 1994.
[13] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

A New Method of Robust Image Compression Based on the Embedded Zerotree Wavelet Algorithm

Charles D. Creusere

***Abstract*—We propose here a wavelet-based image compression algorithm that achieves robustness to transmission errors by partitioning the transform coefficients into groups and independently processing each group using an embedded coder. Thus, a bit error in one group does not affect the others, allowing more uncorrupted information to reach the decoder.**

***Index Terms*—Coefficient partitioning, embedded bitstream, error resilience, image compression, low complexity, wavelets.**

I. INTRODUCTION

Recently, the proliferation of wireless services and the internet along with consumer demand for multimedia products has spurred interest in the transmission of image and video data over noisy communications channels whose capacities vary with time. In such applications, it can be advantageous to combine the source and channel coding (i.e., compression and error correction) processes from both a complexity and an information theory standpoint [1]. In this work, we introduce a form of low-complexity joint source-channel coding in which varying amounts of transmission error robustness can be built directly into an embedded bit stream. The approach taken here modifies Shapiro's embedded zerotree wavelet (EZW) image compression algorithm [2], but the basic idea can be easily applied to other wavelet-based embedded coders such

Manuscript received January 21, 1996; revised January 24, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Amy R. Reibman.

The author is with the Naval Air Warfare Center Weapons Division, China Lake, CA 93555 USA (e-mail: chuck@wavelet.chinalake.navy.mil).

Publisher Item Identifier S 1057-7149(97)07025-5.

as those of Said and Pearlman [3] and Taubman and Zakhor [4]. Some preliminary results using this approach have previously been presented in [5] and [6].

This paper is organized as follows. In Section II, we discuss the conventional EZW image compression algorithm and its resistance to transmission errors. Next, Section III develops our new, robust coder and explores the options associated with its implementation. In Section IV, we analyze the performance of the robust algorithm in the presence of channel errors, and we use the results of this analysis to perform comparisons in Section V. Finally, implementation and complexity issues are discussed in Section VI, followed by conclusions in Section VII.

II. EZW Image Compression

After performing a wavelet transform on the input image, the EZW encoder progressively quantizes the coefficients using a form of bit plane coding to create an embedded representation of the image—i.e., a representation in which a high resolution image also contains all coarser resolutions. This bit plane coding is accomplished by comparing the magnitudes of the wavelet coefficients to a threshold T to determine which of them are significant: if the magnitude is greater than T, that coefficient is significant. As the scanning progresses from low to high spatial frequencies, a 2-b symbol is used to encode the sign and position of all significant coefficients. This symbol can be a $+$ or $-$ indicating the sign of the significant coefficient; a "0" indicating that the coefficient is insignificant; or a zerotree root (ZTR) indicating that the coefficient is insignificant along with all of the finer resolution coefficients corresponding to the same spatial region. The inclusion of the ZTR symbol greatly increases the coding efficiency because it allows the encoder to exploit interscale correlations that have been observed in most images [2]. After computing the "significance map" symbols for a given bit plane, resolution enhancement bits must be transmitted for all significant coefficients; in our implementation, we concatenate two of these to form a symbol. Prior to transmission, the significance and resolution enhancement symbols are arithmetically encoded using the simple adaptive model described in [7] with a four symbol alphabet (plus one stop symbol). The threshold T is then divided by two, and the scanning process is repeated until some rate or distortion target is met. At this point, the stop symbol is transmitted. The decoder, on the other hand, simply accepts the bitstream coming from the encoder, arithmetically decodes it, and progressively builds up the significance map and enhancement list in the exact same way as they were created by the encoder.

The embedded nature of the bitstream produced by this encoder provides a certain degree of error protection. Specifically, all of the information which arrives before the first bit error occurs can be used to reconstruct the image; everything that arrives after is lost. This is in direct contrast to many compression algorithms where a single error can irreparably damage the image. Furthermore, we have found that the EZW algorithm can actually detect an error when its arithmetic decoder terminates (by decoding a stop symbol) before reaching its target rate or distortion. It is easy to see why this must happen. Consider that the encoder and decoder use the same backward adaptive model to calculate the probabilities of the five possible symbols (four data symbols plus the stop symbol) and that these probabilities directly define the codewords. Not surprisingly, the length of a symbol's codeword is inversely proportional to its probability. If a completely random bit sequence is fed into the arithmetic decoder, then the probability of decoding any symbol is completely determined by the initial state of the adaptive model—i.e., the probability weighting defined by the model is not, on the average, changed by a random input.

In our implementation of the Witten *et al.* arithmetic coder [7], we set Max_frequency equal to 500 and maintain the stop symbol probability at 1/cum_freq[0]. Because cum_freq[0] (the sum of the frequency counts of all symbols) is divided by two whenever it exceeds Max_frequency, the probability of decoding a stop symbol stays mostly between 1/250 and 1/500. Thus, if a random bitstream is fed into the decoder after training it to this point, an average of 250 to 500 symbols will be processed before the stop symbol is decoded. The bitstream is correctly interpreted as long as the decoder is synchronized with the encoder, but this synchronization is lost shortly after the first error occurs. Once this happens, the incoming bitstream looks random to the decoder (the more efficient the encoder, the more random it will appear). Since each symbol is represented in the compressed image by between one and two bits, the decoder should self-terminate between 31 and 125 bytes after an error occurs. Experimentally, we have found that the arithmetic decoder overrun is typically between 30 and 50 bytes, which is consistent with the theoretical range, since most of these terminations took place while decoding the highly compressed significance map. If the overrun is small compared to the number of bits correctly decoded, it does not significantly affect the quality of the reconstructed image. While some erroneous information is incorporated into the wavelet coefficients, the bit plane scanning structure ensures that it is widely dispersed spatially, making it visually insignificant in the image.

III. Robust EZW (REZW) Algorithm

A. Basic Approach

As shown in Fig. 1, the basic idea of the REZW image compression algorithm is to divide the wavelet coefficients up into S groups and then to quantize and code each of them independently so that S different embedded bitstreams are created. These bitstreams are then interleaved as appropriate (e.g., bits, bytes, packets, etc.) prior to transmission so that the embedded nature of the composite bitstream is maintained. In the remainder of this paper we assume that individual bits are interleaved. For the REZW approach to be effective, each group of wavelet coefficients must be of equal size and must uniformly span the image. A similar method has been proposed in [8] to parallelize the EZW algorithm, but that method instead groups the coefficients so that data transmission between processors is minimized.

What do we gain by using this new algorithm over the conventional one? As has been pointed out in Section II, the EZW decoder can use all of the bits received before the occurrence of the first error to reconstruct the image. By coding the wavelet coefficients with multiple, independent (and interleaved) bit streams, a single bit error truncates only one of the streams—the others are still completely received. Consequently, the wavelet coefficients represented by the truncated stream are reconstructed at reduced resolution while those represented by the other streams are reconstructed at the full encoder resolution. If the set of coefficients in each stream spans the entire image, then the inverse wavelet transform in the decoder evenly blends the different resolutions so that the resulting image has a spatially consistent quality.

B. Zerotree Preserving (ZP) Partitioning

Fig. 2 graphically illustrates this wavelet coefficient partitioning for $S = 4$ bitstreams and three wavelet scales. In the figure, each coefficient with the same shade of gray maps to the same group and is, therefore, processed by the same encoder. Furthermore, boxes with X's in their middles are used to define the elements of one

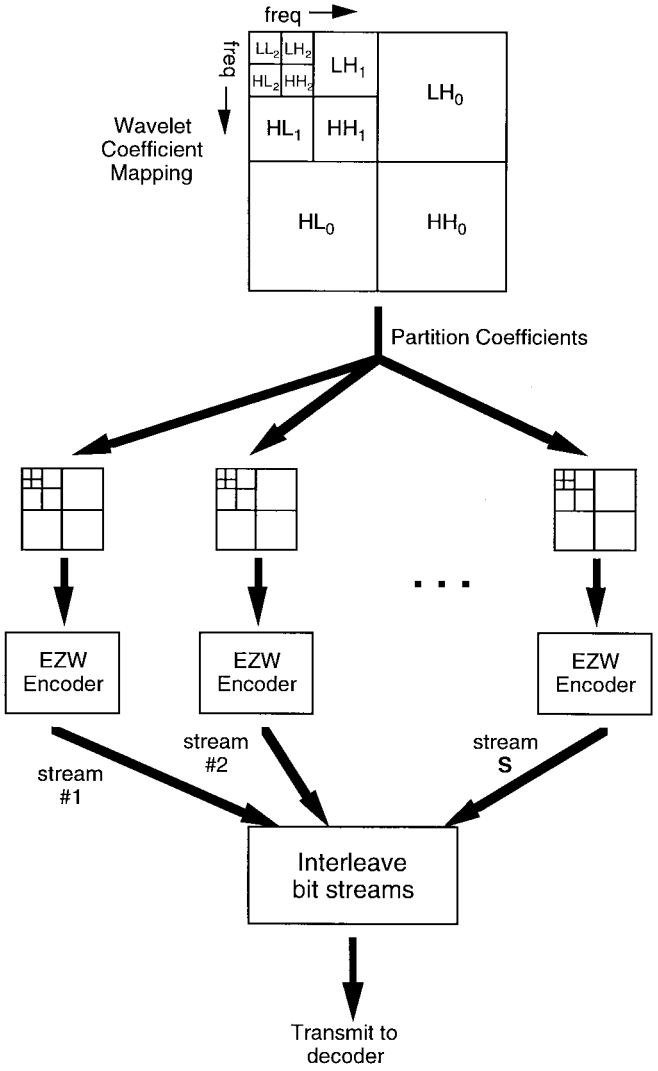

Fig. 1. Structure of the robust embedded zerotree wavelet algorithm.

zerotree, and we note that this zerotree is identical to those used by the conventional EZW coder [2]—hence the name "zerotree preserving." It is clear from this example that all of the elements from a given zerotree are fed into the same encoder and, thus, that the correlation of insignificant coefficients between scales is fully exploited. Note that S can be increased by powers of four until the encoder in each stream processes just one zerotree. If the image is of size $X \times Y$ (assuming for simplicity that both of these are powers of two) and NS scales of wavelet decomposition are used, then the single zerotree limitation implies that the maximum number of independent bit streams allowed is $S = X \cdot Y/4^{NS}$. While introducing more bitstreams is advantageous in terms of robustness to errors, it also results in reduced rate-distortion performance when no transmission errors occur (see Section V). The partitioning of scale j (as labeled in Fig. 1) of the wavelet coefficient mapping $W_j(x, y)$ into $S = 4^K \leq X \cdot Y/4^{NS}$ groups, is given by

$$W_{\Psi,j}(x,y) = W_j\left(\begin{matrix}2^{NS-j-1}\left\{\left\lfloor \frac{x}{2^{NS-j-1}} \right\rfloor \cdot (2^K-1)+n\right\}+x, \\ 2^{NS-j-1}\left\{\left\lfloor \frac{y}{2^{NS-j-1}} \right\rfloor \cdot (2^K-1)+m\right\}+y\end{matrix}\right) \quad (1)$$

where $\Psi = n + 2^k \cdot m$ specifies the stream number for $\{n, m\} \in [0, 2^K - 1]$ and $\lfloor \cdot \rfloor$ is the largest integer less than the argument. Note that the scale j is inversely proportional to the frequency: e.g., $j = 0$ and $j = NS - 1$ reference the highest and lowest frequency subbands, respectively.

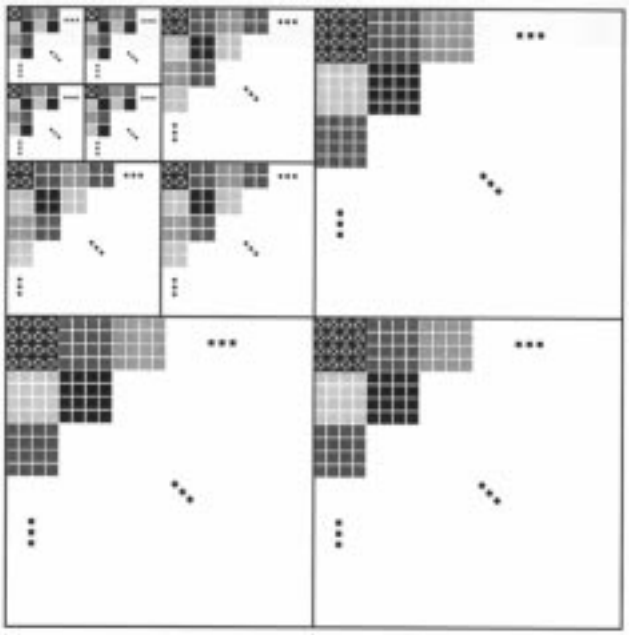

Fig. 2. Zerotree preserving wavelet coefficient partitioning for $S = 4$ and 3 wavelet scales. All coefficients with the same shade map to the same stream, and the X's denote one complete zerotree.

C. Offset Zerotree (OZ) Partitioning

A second partitioning of the wavelet coefficients is illustrated by Fig. 3 when $S = 4$ and $NS = 3$. In this case, the parent-child relationships of the conventional zerotree structure are no longer preserved (as indicated by the boxed X's in the figure). Instead, these relationships are staggered between wavelet scales in such a way that no adjacent wavelet coefficients are input to the same encoder stream. In the more general case where $S = 4^K \leq X \cdot Y/4^{NS}$, every coefficient in a given stream is separated from any others in the same stream by at least $2^K - 1$ wavelet coefficients. As with the zerotree preserving partitioning of Section III-B, every 2^Kth sample (in both the horizontal and vertical directions) of the four coarsest scales is mapped to the same stream. In the offset zerotree partitioning, however, this 2^Kth resampling is applied at all other wavelet scales as well—i.e., (1) is applied to all scales with $j = NS - 1$. The advantage of this method over ZP partitioning is that if one bitstream is terminated prematurely, its reduced-resolution coefficients will be surrounded by full-resolution coefficients at every scale. Thus, the image reconstructed by the inverse wavelet transform retains more of its high-frequency information when selective transmission errors occur. Unfortunately, the zerotrees are now spatially staggered between scales, reducing the interscale correlation between wavelet coefficients and, consequently, the rate-distortion performance of the coding algorithm.

IV. Stochastic Analysis

To evaluate the effectiveness of this family of robust compression algorithms, we assume that the coded image is transmitted through a binary symmetric, memoryless channel with a probability of bit error given by ε. We would like to know the number of bits

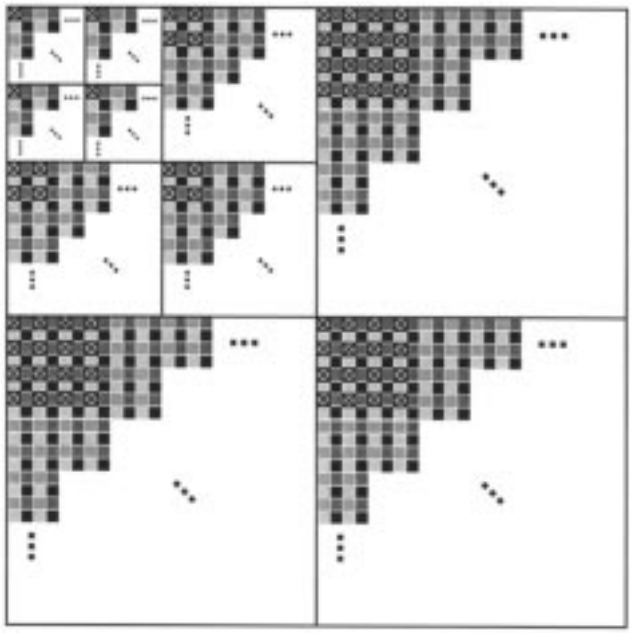

Fig. 3. Offset zerotree wavelet coefficient partitioning for $S = 4$ and three wavelet scales. All coefficients with the same shade map to the same stream, and the X's denote one complete zerotree.

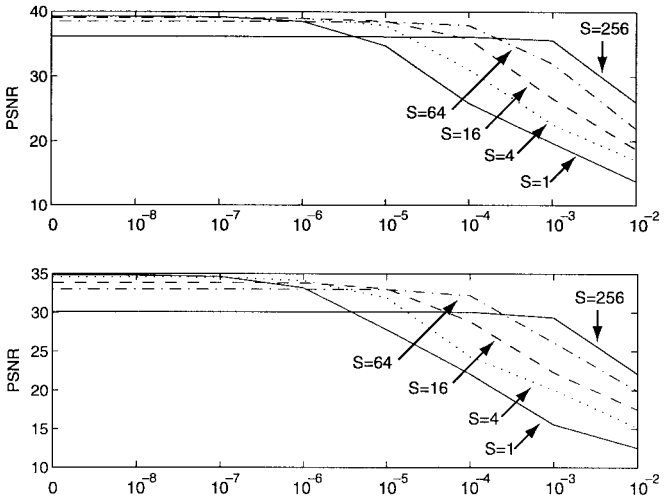

Fig. 4. PSNR (in dB) versus the probability of a bit error. (a) Lena and (b) Barbara coded at 1.0 b/pixel using ZP partitioning.

correctly received in each of the S streams. Since this quantity is itself a random variable, we use its mean value to characterize the performance of the different algorithms. Because the channel is memoryless, streams terminate independently of each other, but the mean values of their termination points are always the same for a specified ε. Assuming that the image is compressed to B total bits and that S streams are used, then the probability of receiving k of the B/S bits in each stream correctly is given by

$$p(k) = \begin{cases} \varepsilon \cdot (1-\varepsilon)^k, & 0 \le k < \dfrac{B}{S} \\ (1-\varepsilon)^k, & k = \dfrac{B}{S} \end{cases} \tag{2}$$

which is a valid probability mass function as one can easily verify by summing over all k. In (2), $(1-\varepsilon)^k$ is the probability that the first k bits are correct while ε is the probability that the $(k+1)$th bit is

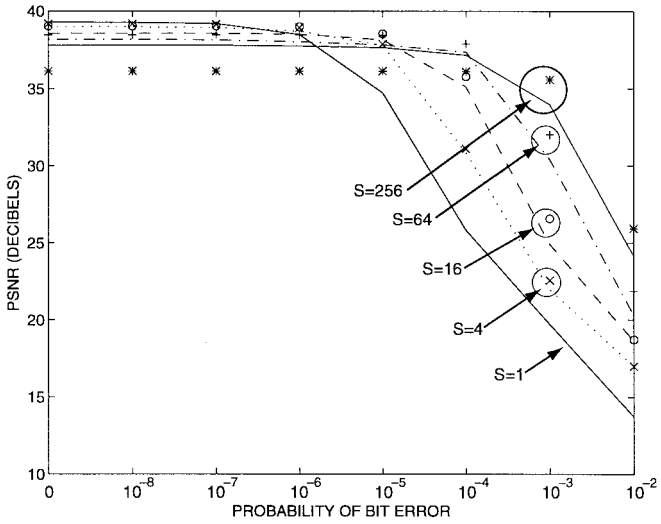

Fig. 5. Comparison of partitioning for Lena image coded at 1.0 b/pixel. Connected lines correspond to OZ partitioning while symbols correspond to ZP partitioning. Note that ZP has slightly better rate-distortion performance in most cases with the major exception being when $S = 256$ and the error rate is low.

in error. Note that a separate term conditioned on B/S is necessary to take into account the possibility that all of the bits in the stream are correctly received. The mean value can now be calculated as

$$m_s = \sum_{k=0}^{B/S} k \cdot p(k). \tag{3}$$

On the average, the total number of bits correctly received is $S \cdot m_s$. If B/S is large relative to $1/\varepsilon$, then $m_s \approx m_1$ for $\forall S$, and, therefore, approximately S times more bits are correctly decoded for the robust algorithm than for the conventional one (i.e., $S = 1$). Generally, the gain actually achieved is not this high, but it is nonetheless significant. In Section V, we use (3) to analyze the impact of transmission errors on the average quality of the reconstructed image for all possible values of S.

V. Results

For these comparisons, we introduce an error into all streams simultaneously according to (3), and we allow each stream to self-terminate. Furthermore, we assume that the size of the image file is known by the decoder or, equivalently, that synchronization information is inserted after each image in a sequence. A five-level transform using the 9/7 biorthogonal wavelet [9] is then applied to the image, and all possible robust partitionings are evaluated ($S = 1$ is, again, the conventional EZW coder). From the objective results summarized in Fig. 4, and the subjective results, illustrated by Figs. 6–10, we note that considerable robustness is achieved at the expense of perfect channel rate-distortion performance. For the Lena image, the peak signal-to-noise ratio (PSNR) drops from 39.28 dB ($S = 1$) to 36.13 dB ($S = 256$), while for Barbara the equivalent drop is from 34.84 dB to 30.09 dB. With an error rate of $\varepsilon = 10^{-2}$, however, the average improvement of the REZW algorithm over a conventional EZW coder is more than 15 dB for the same images. Examining Fig. 5, we note that the objective performance of the zerotree preserving partitioning is generally superior to that achieved using offset zerotrees. This superiority is even more evident in the subjective quality as shown by Figs. 7 and 8. Most of the artifacts in Fig. 8 are actually introduced by the arithmetic decoder overrun, and if this is eliminated (e.g., the position of the error is known), the ZP and OZ partitionings result in almost identical subjective quality at all

Fig. 6. Lena coded to 1.0 b/pixel using conventional EZW algorithm with probability of error $= 10^{-3}$. PSNR $=$ 19.71 dB.

Fig. 8. Lena coded to 1.0 b/pixel using OZ partitioning with $S = 4$ and probability of error $= 10^{-3}$. PSNR $=$ 21.98 dB.

Fig. 7. Lena coded to 1.0 b/pixel using ZP partitioning with $S = 4$ and probability of error $= 10^{-3}$. PSNR $=$ 22.78 dB.

Fig. 9. Lena coded to 1.0 b/pixel using ZP partitioning with $S = 256$ and probability of error $= 10^{-3}$. PSNR $=$ 35.06 dB.

error rates. Overrun distortion becomes less visually significant when more streams are used as is illustrated by Figs. 9 and 10, although it still reduces the PSNR more dramatically for the OZ partitioning. For a multichannel transmission method such as orthogonal frequency division multiplexing, the offset zerotree partitioning is perceptually superior to the zerotree preserving partitioning because the error rates in the different channels can be radically different. The advantages of the offset zerotree partitioning in this situation are clearly shown in Figs. 11 and 12, where the truncation point in each stream has been separately calculated using (3). Note that here we stop the coding process in each stream immediately after the first bit error occurs to prevent the decoder overrun from obscuring the comparison.

VI. Implementation and Complexity

The robust compression algorithm illustrated by Fig. 1 can be implemented efficiently either on a conventional sequential processor or on an array of multiple instruction multiple data (MIMD) parallel processors. A sequential implementation first performs a wavelet transform and then executes one of the vertical branches of Fig. 1 until the appropriate number of bits (based on the interleaving structure) are produced. At that point, execution control is passed to the next branch, and the process continues in a round-robin fashion until the bit allocation is exhausted. In the parallel implementation, a parallel wavelet decomposition is performed and each branch in

Fig. 10. Lena coded to 1.0 b/pixel using OZ partitioning with $S = 256$ and probability of error $= 10^{-3}$. PSNR $=$ 33.95 dB.

Fig. 11. Lena coded to 1.0 b/pixel using ZP partitioning with $S = 4$ and probability of errors of $10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} in each stream. PSNR $=$ 20.33 dB.

the figure is executed independently by separate processors. In this case, either the outputs of the processors have to be synchronized to achieve proper interleaving or an additional output processor must be used to organize the data.

Neither implementation of the robust algorithm significantly affects the complexity of the complete system. Parallel wavelet transforms do have higher computational complexity [10], but this increase is also incurred in a parallel implementation of the conventional algorithm [8]. Furthermore, the search complexity of the REZW coder actually decreases slightly with S since the number of significant coefficients

Fig. 12. Lena coded to 1.0 b/pixel using OZ partitioning with $S = 4$ and probability of errors of $10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} in each stream. PSNR $=$ 22.70 dB.

trends downward. The only quantifiable disadvantage of the robust algorithm is that it requires more temporary storage space since the coders operating in each stream must have their own thresholds and adaptive models. In reality, however, its most significant drawback is that it is simply more intricate than EZW, making it harder to efficiently program. Once programmed, it is just as fast as the conventional coder and requires only a small amount of additional memory.

VII. Conclusion

We have shown here how robustness to transmission errors can be added to an embedded image compression algorithm with little or no definable increase in its complexity. As the number of partitions, S, increases, the resilience of the coded image to transmission errors also increases, but the perfect channel rate-distortion performance of the codec decreases. If the bit error rate of the channel is greater than 10^{-6}, however, the performance of our robust compression algorithm is generally better than that of the conventional embedded zerotree wavelet coder for some value of S.

Acknowledgment

The author thanks Dr. A. Reibman and the anonymous reviewers for their invaluable comments and suggestions.

References

[1] T. M. Cover, "Broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 2–14, Jan. 1972.

[2] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, Dec. 1993, pp. 3445–3462.

[3] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning into hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[4] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, Sept. 1994, pp. 572–588.

[5] C. D. Creusere, "Robust image coding using the embedded zerotree wavelet algorithm," in *Proc. Data Compression Conference*, Snowbird, UT, Mar. 1996, pp. 432.
[6] ____, "A family of image compression algorithms which are robust to transmission errors," *Proc. SPIE*, vol. 2825, pp. 890–900, Aug. 1996, .
[7] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[8] C. D. Creusere, "Image coding using parallel implementations of the embedded zerotree wavelet algorithm," *Proc. SPIE*, vol. 2668, pp. 82–92, Jan. 1996.
[9] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia, PA: SIAM, 1992.
[10] J. Lu, "Parallelizing Mallat algorithm for 2-D wavelet transforms," *Inform. Process. Lett.*, vol. 45, 1993, pp. 255–259.

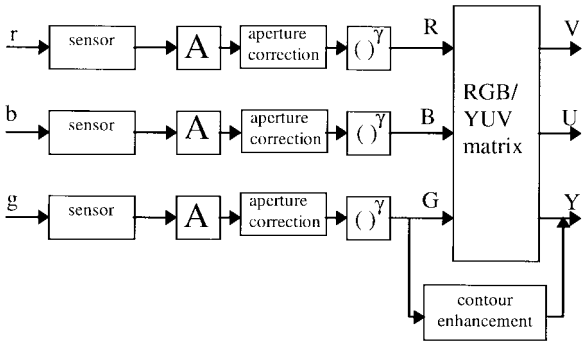

Fig. 1. Simplified color camera model.

An Adaptive Order-Statistic Noise Filter for Gamma-Corrected Image Sequences

Richard P. Kleihorst, Reginald L. Lagendijk, and Jan Biemond

***Abstract*— Original video signals are often corrupted by a certain amount of noise originating from the camera electronics. As a result of the gamma correction in cameras, the observed noise is signal dependent. In this correspondence, we present a spatio-temporal order-statistic (OS) noise filter that takes into account the gamma correction in the camera. The calculation of the filter coefficients requires higher-order order-statistics (HOOS) of the noise process. We make use of a range test (RT) to determine locally from which neighboring signal values an estimate should be formed. The noise filter that we arrive at is adaptive and computationally efficient.**

***Index Terms*—Image filtering, image sequence processing, noise filtering, order statistics.**

I. Introduction

Most digital video signals that we consider as being the original and perfect recordings of a natural scene, are often corrupted by a certain amount of noise. Surprisingly, the amount of noise in original video signals is much higher than one would expect from the quantization of the luminance and chrominance components in 8 to 12 b. The dominant cause of the noise lies usually in thermal effects in the electronic circuitry. For instance, the well-known "CalTrain" video often used in compression literature has an estimated signal-to-noise ratio (SNR) of 28 dB. Reducing the amount of observed noise is desirable to improve the visual quality, but is also important as a preprocessing stage in many signal processing applications of interest, such as video compression and analysis.

A complicating factor in the filtering of camera noise is the gamma correction that is present in any camera to compensate for the nonlinearity of the display's cathode ray tube (CRT). The gamma correction results in signal-dependent observation noise [1]. A typical noise filter for video signals, however, usually ignores the signal dependency of the observation noise. We propose an order-statistic (OS) spatio-temporal finite impulse response (FIR) estimator for removing signal-dependent camera noise from image sequences [2], [3]. Various forms of order statistics estimators have been introduced in the literature, ranging from simple median, minimum, and maximum filters to adaptive estimators using linear combinations of ordered observations [4]–[6]. Due to the ordering of the observations prior to the actual filtering operation, OS estimators often outperform linear estimators.

In this correspondence, we propose a new technique to estimate the weights of an adaptive OS estimator. The applicability of this technique is not limited to the noise removal problem that we consider in this correspondence, but can be used for any nonlinear signal estimation problem [7]. The resulting estimator weights are optimally adapted to the probability density function (pdf) of the noise and to the spatio-temporally localized properties of the video signal. The estimator weights are a function of *higher-order order-statistics (HOOS)* of the noise process. Furthermore, we make use of a *range test (RT)* to remove outliers from the spatio-temporal estimation window so as to reduce the estimator's variance. The result is an optimal and computationally efficient nonlinear filter for camera noise.

II. Signal Model

The properties of the observation noise depend on the camera structure. A simplified model for a color camera is shown in Fig. 1. The electrical current generated by the red, green, and blue (RGB) sensors depends directly on the intensity of the imaged object, and takes on values between 0.0 and 1.0. After suitable amplification and possibly aperture correction, the RGB video signals pass a gamma-correction stage to compensate for the CRT's nonlinearity. Finally, an RGB/YUV matrix may transform the RGB signals into luminance and chrominance (YUV) components. As a postprocessing step, the luminance signal is sometimes spatially enhanced (sharpened). In this correspondence, we assume that the sharpening and aperture correction are switched off or have a negligible effect on the video signal's statistical properties. Further, we assume that the RGB/YUV matrix is also switched off and that the RGB video signals are quantized in 8 b.

The above model can be mathematically summarized as follows. The ideal noise-free spatio-temporal camera signal $f(i,j,k)$ is cor-

Manuscript received March 24, 1995; revised November 1, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael T. Orchard.

R. P. Kleihorst is with the Philips Research Laboratories, Eindhoven, The Netherlands.

R. L. Lagendijk and J. Biemond are with the Delft University of Technology, Department of Electrical Engineering, Information Theory Group, Delft, The Netherlands.

Publisher Item Identifier S 1057-7149(97)07029-2.