ON THE ROBUSTNESS OF VECTOR SET PARTITIONING IMAGE CODERS TO BIT ERRORS*

Debargha Mukherjee and Sanjit K. Mitra

Image Processing Laboratory
Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106.
Tel: (805) 893-8312, FAX: (805) 893-3262, Email: (debu,mitra)@iplab.ece.ucsb.edu

ABSTRACT

A vector enhancement of Said and Pearlman's Set Partitioning in Hierarchical Trees (SPIHT) methodology, named VSPIHT, has recently been proposed for embedded wavelet image compression. A major advantage of vector based embedded coding with fixed length VQs over scalar embedded coding, is its superior robustness to noise. In this work we show that vector set partitioning can effectively alter the balance of bits in the bit stream so that significantly fewer critical bits carrying significance information is transmitted, thereby improving inherent noise resilience. Additionally, the degradation in reconstruction quality caused by errors in non-critical quantization information, can be reduced by appropriate VQ indexing, or designing channel optimized VQs for the successive refinement systems. For very noisy channels unequal error protection to the critical and non critical bits with either block codes or convolution codes are used. Extensive simulation results are presented.

1. INTRODUCTION

Inspired by the success of Shapiro's Embedded Zerotree Wavelet (EZW) algorithm [1], and Said and Pearlman's improved Set Partitioning in Hierarchical Trees (SPIHT) algorithm [2], several researchers have recently proposed vector extensions (VSPIHT) to these scalar algorithms, both based on lattice VQ [3]-[5], and trained VQ [6]. While all these zerotree based wavelet coders, either scalar- or vector-based, are more efficient in the rate-distortion sense than traditional DCT based coders in the absence of noise, a pervasive reason for concern over such coders is their poor inherent resilience to errors in the bit stream. Lack of robustness to bit flips can be a serious constraint in many applications where bit errors in the channel between the encoder and the decoder are unavoidable. Sherwood and Zeger [7] showed that by the use of proper error correction techniques to protect equally all the bits generated by a scalar SPIHT coder, very effective noise resilient coders can be obtained. Their scheme is more appropriate for channels where the bit error rate (BER) remains approximately constant. In this work, we first present an evaluation of the degree of inherent robustness of scalar and vector SPIHT coders, by dividing the bitstream they generate into *critical* and *non-critical* bits, and comparing the proportion of each in the bitstream. It is shown that in low noise channels, high dimensional VSPIHT can produce bitstreams vastly more noise resilient than those produced by scalar SPIHT, even without any protection. In high noise situations, typical of wireless channels, unequal error protection for the critical and non-critical bits can be used to accommodate bursty channels. We present results for both block codes and convolution codes for unequal error protection.

Section 2 presents an overview of scalar and vector SPIHT. Section 3 introduces an approach to classifying the generated bitstream as either critical or non-critical, and discusses noise resilience within this framework. Section 4 shows the noise advantage of vector SPIHT over scalar SPIHT by experimental verification of the distribution of bits in both. Section 5 presents some actual coding results comparing scalar SPIHT to various VSPIHT coders in the presence of noise. Section 6 concludes this paper.

2. SCALAR AND VECTOR SPIHT

The essence of set-partitioning is to first classify the elemental coding units based on their magnitude, and then to quantize them in a successive refinement framework. The elemental coding units are scalar wavelet coefficients in original SPIHT [2], and more generally vectors of wavelet coefficients in VSPIHT [5]-[6]. The classification is performed in multiple passes using octavely decreasing thresholds. Each pass yields a new set of elements in the image that have magnitudes higher than the threshold associated with the pass (i.e., are *significant*), but lower than that associated with the previous pass. These elements are roughly quantized in the same pass, and gradually refined in all successive passes.

Said and Pearlman [2] divides each pass into two subpasses: the *Sorting pass*, and the *Refinement pass*. The Sorting pass is concerned with deciding, which, among the insignificant elements, become newly significant in the pass, and transmitting that information to the decoder. Additionally, the first stage quantization of the elements newly decided as significant is also performed in the sorting pass. The Refinement pass, which follows the Sorting pass, is used to refine the estimate for the elements already decided as significant in previous Sorting passes.

The difference between scalar and vector SPIHT is only in the type of elements used in each. In scalar SPIHT, the elemental coding units are the scalar wavelet coefficients, and their scalar magnitude is used as a basis for classification. As soon as a coefficient becomes significant in a Sorting pass, its sign is also transmitted. Its magnitude is thereafter refined in all Refinement passes following this pass. In VSPIHT, the elemental coding units are vectors of size *HV* obtained by grouping the wavelet

---

* This work was supported by the University of California MICRO grant with matching support from Lucent Technologies, Raytheon Missile Systems, Tektronix Corporation and Xerox Corporation.

coefficients in windows of size $H \times V$ in each subband. The classification boundaries are concentric hyperspheres in HV dimensional space. All significant vectors are quantized using successive refinement systems consisting of tree-structured VQs, multistage VQs, or a combination of both [8]. The first stage vector quantization is performed in the Sorting pass as soon as a vector becomes significant, while the later refinements are made in the Refinement passes following this pass.

3. NOISE RESILIENCE ISSUES IN SET PARTITIONING CODERS

3.1 Critical and Non-Critical Bits

In general any coder producing variable length codes, are inherently vulnerable to bit stream errors.The set-partitioning coders are no exceptions. It is possible however to discuss their degree of vulnerability to bit errors by analyzing the bitstream they generate. The bit stream can be separated into two parts: *critical* and *non-critical* bits. The critical bits are those which cause desynchronization between the encoder and the decoder. In the set-partitioning context, they consist of the bits carrying significance information. A single bit error in the critical bits cause catastrophic failure of the reconstruction process beyond that point. The non-critical bits, on the other hand cause less severe errors. The effect of these errors are limited to single coding units, scalars or vectors, and do not disturb the progression of the decoding process after they occur. Non-critical errors consist of errors in the actual quantization information. For scalar SPIHT, they consist of the sign information in the Sorting pass for coefficients that newly become significant, and all the Refinement information in the Refinement passes. For VSPIHT, they comprise the first stage VQ index for the vectors newly found to be significant in the Sorting pass, and the later stage VQ indices in all of the Refinement passes, assuming that all the VQ indices are fixed length coded.

3.2 Effect of Adaptive Arithmetic Coding

Adaptive arithmetic coding is often used to reduce the number of significance information bits transmitted in set-partitioning coders [2]. Additionally, for VSPIHT, adaptive arithmetic coding for the VQ indices can be used. An entire arithmetic coded bitstream is represented by an interval between 0 and 1. As a result, the critical and non-critical bits are not really separable. Although it is possible under certain conditions for certain bit errors not to affect the decoding of information subsequently transmitted following the error, we do not consider such situations in this work. We concentrate solely on the non-arithmetic coded versions of scalar and vector set-partitioning, where the separation between critical and non-critical bits is distinct. Furthermore, as will be apparent shortly, inherent robustness to bit-errors requires high dimensional vectors for VSPIHT, in which case, there is little to be gained by adaptive arithmetic coding anyway.

3.3 Effect of Bit-Errors on Rate-Distortion Curve

We now present a typical rate-distortion curve obtained by decoding the bit stream received from a set-partitioning coder. Figure 1 shows the typical progression of the reconstruction PSNR as more and more bits from the embedded bit stream are used. The dashed curve shows the noiseless case. In the presence of noise, the curve degrades to the solid one. As long as there are no critical bit errors, and the errors in non-critical bits are below reasonable bounds, the PSNR curve continues to rise.

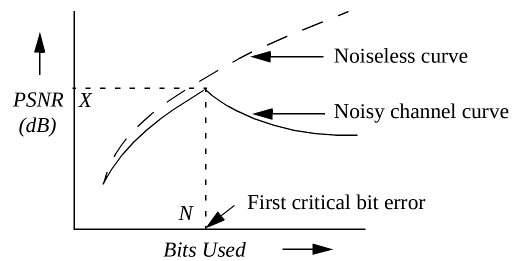

Figure 1. Typical SNR vs. Bits used curve for SPIHT/VSPIHT in noisy channels

The rise is of course not as much as in the noiseless case due to non-critical bit errors affecting the reconstruction. However, as soon as a critical bit error is encountered, say at bit number N, synchronization is lost. The PSNR progression becomes highly uncertain beyond this point, and typically degrades. The bit N, at which the first critical error occurs, therefore determines to a large extent, the quality of the reconstruction in a low noise channel. A major consideration in the design of an inherently noise resilient set-partitioning coder then is to increase the expected value of N as much as possible.

3.4 A Simplistic Computation of $E(N)$

A simplistic computation of the expected number of bits that are decoded in a received bitstream, before the first critical failure occurs, is now presented. Critical bits only occur in Sorting passes. The Refinement pass is entirely non-critical if fixed length codes are used. Assume α to be the fraction of the bits in the Sorting pass that are critical. Since the critical and non-critical bits are inseparably mixed in the Sorting passes, α can be regarded as the probability of a Sorting pass bit being critical. If the Sorting pass bit number at which the first critical bit error occurs is denoted by P, and if the bitstream is transmitted over a binary symmetric channel with a bit error rate (BER) of β, the expected value of P, is given as:

$$E(P) = \sum_{m=1}^{\infty} m \cdot (1-\alpha\beta)^{m-1} \cdot \alpha\beta = \frac{1}{\alpha\beta} \qquad (1)$$

Furthermore, let the proportion of bits used for the Sorting and the Refinement passes, as seen by an arriving sorting pass bit, be on an average, σ and $(1-\sigma)$ respectively. The expected value of N, $E(N)$, is then roughly equal to:

$$E(N) = \frac{E(P)}{\sigma} = \frac{1}{\alpha\beta\sigma} \qquad (2)$$

Thus, for a given channel BER β, the product $\gamma = \alpha\sigma$, determines the expected waiting time for a critical bit error to occur. This quantity can be defined as the average fraction of total bits used that are critical, as seen by an arriving sorting pass bit. Note that this quantity is close to the overall average fraction of bits that are critical, though not exactly the same. The lower the value of γ, the larger the value of $E(N)$, and vice versa.

4. VECTOR SET PARTITIONING ADVANTAGE

4.1 Distribution of Bits

We now show that vector SPIHT with fixed length VQ indices can effect a shift in the balance of bits between the critical and non-critical bits in set-partitioning, so that a smaller value of γ is obtained. To support the claim we compare the actual value of γ obtained from the scalar SPIHT coder of [2], with that from VSPIHT coders of various dimensions for several images up to various bitrates. The VSPIHT coders use successive refinement VQ systems with tree-structured first stages, followed by several stages of multistage VQ. The bit allocation is balanced to ensure that an error in the VQ indices do not induce loss in synchronization, and the codes used are fixed length. Table 1(a)-(c) shows the actual γ values obtained from various coders for the 512×512 *Lena*, *Goldhill* and *Barbara* images, computed up to several bitrates. The VSPIHT coder of dimension 4 is similar to that presented in [6]. The VSPIHT coders of dimension 8 and 16 considered in the table, splits the larger vectors into smaller vectors of dimension 4, for tree-multistage successive refinement. From the table, it is apparent that as the dimensionality of vectoring in VSPIHT increases, the balance of bits shifts so that lower and lower values of γ are obtained. This implies that the expected number of bits to traverse before the first critical bit error occurs is inverse proportionally increased. It is this property of VSPIHT that makes it inherently noise resilient.

We compare in Figure 1, the typical rate distortion curves obtained for a scalar SPIHT coder and a vector SPIHT coder, in noisy channels. Note that the expected number of bits at which critical failure occurs is much higher for the VSPIHT case than for scalar SPIHT. The improvement in reconstruction PSNR at critical failure is also shown.

4.2 Effect of Non-Critical Bit Errors

Figure 1 presents an interesting insight into the behavior of scalar and vector SPIHT. Although the VSPIHT curve is shown to perform better than SPIHT in the noiseless case, in presence of noise, the degradation due to non-critical errors is more severe than in scalar SPIHT. This however, is expected, because a single bit error in a VQ index affects as many coefficients as is the dimension of the VQ, rather than a single coefficient in the scalar case. One reason for using split-VQ in high dimension

Table 1(a). Lena Image γ values

BPP	SPIHT	VSPIHT (dim 4)	VSPIHT (dim 8)	VSPIHT (dim 16)
0.2	0.731	0.313	0.160	0.088
0.4	0.714	0.302	0.162	0.085
0.6	0.739	0.259	0.183	0.098

Table 1(b). Goldhill Image γ values

BPP	SPIHT	VSPIHT (dim 4)	VSPIHT (dim 8)	VSPIHT (dim 16)
0.2	0.794	0.303	0.186	0.092
0.4	0.769	0.329	0.175	0.090
0.6	0.754	0.273	0.197	0.104

Table 1(c). Barbara Image γ values

BPP	SPIHT	VSPIHT (dim 4)	VSPIHT (dim 8)	VSPIHT (dim 16)
0.2	0.772	0.332	0.168	0.089
0.4	0.760	0.313	0.159	0.083
0.6	0.740	0.257	0.160	0.089

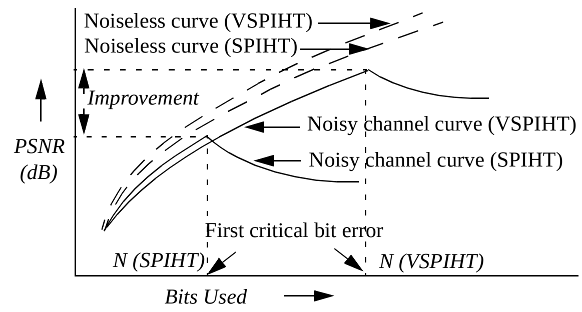

Figure 2. Comparing typical PSNR vs. Bits used curves for SPIHT and VSPIHT in noisy channels

VSPIHT is to reduce this number while retaining the same low value of γ. The situation can be improved further in the low noise case, without losing quantization efficiency, by using an appropriate indexing scheme on the VQs, so that the more likely errors lead to less severe distortion. However, in noisier situations, it is more advantageous to use channel-optimized VQs (COVQs) at each stage of refinement. Standard Euclidean vector quantization obtains the best encoding index i from a size N_{cb} codebook with codevectors y_i, $i = 0, 1, \ldots, N_{cb} - 1$, for a given input source vector x, as:

$$i = \arg min(\|x - y_i\|^2), over\{0, 1, \ldots, N_{cb} - 1\} , \qquad (3)$$

whereas, Euclidean channel-optimized vector quantization obtains the best encoding index i as:

$$i = \arg min\left(\sum_{j=1}^{N_{cb}} P_{j/i} \cdot \|x - y_j\|^2\right), over\{0, 1, \ldots, N_{cb} - 1\} , \qquad (4)$$

where the $P_{j/i}$ are the transition probabilities of receiving index j over the channel, given index i is transmitted. COVQ strives to minimize the end-to-end expected distortion between the input and the reconstruction, in the presence of noise. The Lloyd's VQ design algorithm is appropriately modified to minimize the new distortion metric [9]. For a given bit error rate, the transition probabilities for the VQ indices are computed and used to design the COVQs at every stage using the algorithm in [9]. To obtain the tree-structured multistage codebooks for each VSPIHT class, after the first stage COVQ has been designed, the training set is partitioned, and the smaller training sets are then used to design the multistage codebooks in a greedy manner.

4.3 Unequal Error Protection

As the noise level in the channel increases, it becomes necessary to protect both the critical bits and the non-critical bits using error correction codes. However, unlike [7], our approach is based on unequal error protection for the critical and non-critical bits. The critical bits are protected heavily so that no errors are made. Since the critical bits are few in high dimensional VSPIHT, the bandwidth expended by this heavy protection remains small. The non-critical bits are protected with less powerful codes on an average. A few errors in non-critical information are tolerated in favor of using high rate error correction codes. This approach is especially suited to bursty channels where the BER can show occasional spikes. The critical bits are conservatively protected, so that critical failure does not occur even at the peak BER. The protection for the non-critical bits will be sufficient to withstand errors at a little more than the average BER for the channel, but during occasional BER peaks, non-critical bit errors will allow graceful degradation. In practice, it is advantageous to protect the non-critical bits from the first few passes more heavily than those from the later passes. Since the first few passes carry more important VQ indices, this guarantees a minimum image quality at all times.

In order to allow unequal error protection, the source coder outputs critical and non-critical bits to two separate buffers. The critical bit buffer is checked after every bit addition to see if the number of bits accumulated therein has reached the block size of the block coder or convolution coder to be used. As soon as this number is reached, the critical bit buffer is channel coded and transmitted over the channel. Following the critical block transmission, the bits accumulated in the non-critical bit buffer are blocked, coded and transmitted, possibly multiple times, until the bits left therein come to be less than the blocksize required for coding. The bitstream over the channel thus consists of repetitive patterns of single critical blocks followed by an arbitrary number of non-critical blocks. This protocol of transmitting the significance information ahead of the quantization information, allows the decoder to perfectly synchronize itself to the encoder, as long as the critical information is received error-free. The additional flexibility of allowing different channel coders for different passes of the VSPIHT coder is easily incorporated within this framework. Whenever bits from the non-critical bit buffer are to be transmitted, the channel coder corresponding to the pass to which the first bit in the buffer belongs is used. Both convolution coders and block coders can be used for error correction. While convolution coders are more powerful, they require larger block lengths, which increase the granularity of embedding. Block codes, with smaller block lengths, allow better preservation of the embedding property.

5. EXPERIMENTAL RESULTS

5.1 Low Noise Channels: No Error Protection

To demonstrate the inherent robustness of high-dimensional vector SPIHT, we now compare the actual PSNR vs. bitrate results obtained by scalar SPIHT and vector SPIHT of dimension 16, both without arithmetic coding, at the end of a low noise channel. No error protection is used in either case, but the VQs are appropriately indexed in the VSPIHT case, to reduce the effect of errors. If a critical bit error occurs, the PSNR value is frozen at the point when the critical failure occurs (shown as *X* in Figure 1). The evolution of the PSNR beyond that point is highly unpredictable and does not yield any insight. In practice, it is possible to detect loss in synchronization soon after it occurs, by incorporating periodic redundancy checks in the bitstream syntax. Once critical failure is detected, the decoder stops decoding the bitstream, because if continued, the reconstruction is likely to become worse.

We use the *Lena* image for our experiments. Scalar and vector set-partitioning coding runs are made for several bitrates at several bit-error-rates. The SNR values obtained for a total of 50 runs for each case is averaged to obtain a final representative SNR for the image at that bitrate and BER. The results thus obtained are tabulated in Table 2. In the absence of noise, the SPIHT coder without arithmetic coding perform a little better than the VSPIHT(16) coder also without arithmetic coding. However, as the BER becomes non-trivial, the VSPIHT coder demonstrates more and more robustness.

Table 2. Average SNR values for various bitrates and various bit-error-rates for the *Lena* Image coded by SPIHT and VSPIHT (dim 16).

BPP	Algorithm	BER=10^{-5}	BER=10^{-4}
0.2	SPIHT	31.54	25.15
0.2	VSPIHT(16)	32.35	30.46
0.4	SPIHT	32.68	25.33
0.4	VSPIHT(16)	34.36	31.69
0.6	SPIHT	33.90	25.90
0.6	VSPIHT(16)	36.21	32.99

5.2 High Noise Channels: Unequal Error Protection

We present results for four implementations of unequally protected VSPIHT, two with block codes and the other two with convolution codes. In the first block code implementation for a BER of 0.001, all the critical bits are protected with a (15, 11) 1-error correcting BCH code. For the non-critical bits, the same BCH code is used for the first three passes, a (31, 26) 1-error correcting BCH code is used for the fourth and fifth passes, a (63, 57) 1-error correcting BCH code is used for the sixth pass, and the remaining passes are left unprotected. The SNR results obtained on an average by VSPIHT of dim 16 in conjunction with this error protection scheme, from several runs at various bit rates, is presented in Table 2(a). In the second block code implementation for a BER of 0.01, all the critical bits are protected with a (23, 12) 3-error correcting Golay code. For the non-critical bits, the same Golay code is used for the first three passes, a (31, 21) 2-error correcting BCH code is used for the next three passes, and the (15, 11) BCH code is used for the remaining passes. The SNR results are presented in Table 2(b).

Table 2(a). Average SNR (dB) values at BER = 0.001 for images coded by VSPIHT (dim 16) and unequal block codes.

BPP	*Lena*	*Goldhill*	*Barbara*
0.2	31.36	29.02	25.69
0.4	34.72	31.42	28.44
0.6	36.18	32.51	30.74

Table 2(b). Average SNR (dB) values at BER = 0.01 for images coded by VSPIHT (dim 16) and unequal block codes.

BPP	*Lena*	*Goldhill*	*Barbara*
0.2	30.27	28.09	24.63
0.4	33.38	30.07	27.30
0.6	35.29	31.65	29.39

In the first convolution code based implementation for a BER of 0.01, all the critical bits are protected with a (3, 2, 3) convolution code with a block size of 100 information bits, terminated with 6 zerobits to flush the memory, so that the overall channel block length is 159 bits. The non-critical bits from the first three passes are protected with the same convolution code but with a block size of 180 information bits (279 channel bits with zero termination). Those from the remaining passes are coded with a (4, 3, 2) convolution code with a block size of 180 information bits (248 channel bits with zero termination). The average SNR results for VSPIHT (dim 16) in conjunction with the above unequal protection scheme for various images are presented in Table 2(c). In the second convolution code implementation at a BER of 0.1, all the critical bits are protected with a (4, 1, 6) convolution code with a block size of 120 information bits, terminated with 6 zerobits to flush the memory, so that the overall channel block length is 504 bits. The non-critical bits from the first three passes are protected with the same convolution code but with a block size of 180 information bits (744 channel bits with zero termination). Those from the next three passes are coded with a (3, 1, 7) convolution code with a block size of 180 information bits (561 channel bits with zero termination). Those from the remaining passes are protected with a (2, 1, 8) convolution code with a block size of 180 information bits (376 bits with zero termination). The average SNR results for VSPIHT (dim 16) in conjunction with the above unequal protection scheme for various images are presented in Table 2(d).All the convolution codes used are taken from the tables in [10].

Table 2(c). Average SNR (dB) values at BER = 0.01 for images coded by VSPIHT (dim 16) and unequal convolution codes.

BPP	***Lena***	***Goldhill***	***Barbara***
0.2	30.67	28.44	25.07
0.4	33.58	30.23	27.62
0.6	35.35	31.78	29.81

Table 2(d). Average SNR (dB) values at BER = 0.1 for images coded by VSPIHT (dim 16) and unequal convolution codes.

BPP	***Lena***	***Goldhill***	***Barbara***
0.2	27.11	26.23	22.96
0.4	29.88	27.93	24.49
0.6	31.53	29.06	26.03

The SNR results presented above are a little lower than those presented in [7], especially for the *Lena* image. This difference is attributed solely to the loss in source coding efficiency that results from the use of a sub-optimal split VQ based 16-dimensional VSPIHT scheme without arithmetic coding, as opposed to the very efficient arithmetic coded scalar SPIHT scheme [2]. While this loss in source coding performance is offset partially by the unequal error protection scheme, the true advantage of high dimensional VSPIHT with unequal error protection is over bursty channels, which we next demonstrate. Consider a dual-state channel over which the BER is 0.01, 95% of the time, and 0.1, the remaining 5% of the time, with the average burst length in the high BER state being 100 channel bits. To transmit VSPIHT coded images reliably over such a channel, an unequal protection scheme with interleaved block codes was designed. The critical bits and the non-critical bits from the first pass are protected with a (63, 18) 10-error correcting BCH code with interleaving degree 10 (effective block length 630 channel bits). The non-critical bits from the second and third passes are protected with a (31, 11) 5-error correcting BCH code with interleaving degree 20 (effective block length 620 channel bits), those from the fourth pass are protected with a (63, 30) 6-error correcting code with interleaving degree 10 (effective block length 630 channel bits), those from the fifth pass are protected with a (63, 39) 4-error correcting BCH code with interleaving degree 10 (effective block length 630 channel bits), those from the sixth pass are protected with a (63, 45) 3-error correcting BCH code with interleaving degree 10 (effective block length 630 channel bits), and those from the remaining passes are protected with the BCH (15, 11) 1-error correcting code with no interleaving. The codes and the interleavers are so chosen that the effective channel block length never exceeds 630 bits, thus preserving sufficiently fine granularity of embedding. The SNR results are shown in Table 3. An equal protection scheme (like [7]), applied on an inherently vulnerable bitstream as that obtained from arithmetic-coded scalar SPIHT [2], needs to designed for the worst-case scenario, and therefore loses efficiency.

Table 3. Average SNR (dB) values over a dual-state channel with BER = 0.01 (95% of the time), and = 0.1 (5% of the time) with average burst length 100 bits, for images coded by VSPIHT (dim 16) and unequal interleaved block codes.

BPP	***Lena***	***Goldhill***	***Barbara***
0.2	28.82	27.49	23.78
0.4	32.50	29.47	26.33
0.6	34.07	30.89	27.93

6. CONCLUSION

The inherent resilience of non-entropy coded vector set-partitioning wavelet image coders to bit errors, is demonstrated. High dimensional VSPIHT is shown to be especially suitable for unequal error protection over bursty channels because of fewer critical bits.

7. REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-62, December 1993.

[2] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-50, June 1996.

[3] E. A. B. Da Silva, D. G. Sampson, M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 299-310, February 1996.

[4] J. Knipe, X. Li, and B. Han, "An improved lattice vector quantization scheme for wavelet compression," *IEEE Trans. Signal Processing*, vol. 46, no. 1, pp. 239-43, January 1998.

[5] D. Mukherjee and S. K. Mitra, "Vector set partitioning with successive refinement Voronoi lattice VQ for embedded wavelet image coding," *Proc. International Conference on Image processing*, Chicago, Illinois, vol. 1, pp. 107-11, October 1998.

[6] D. Mukherjee and S. K. Mitra, "Vector set partitioning with classified successive refinement VQ for embedded wavelet image and video coding," *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Seattle, Washington, vol. 5, pp. 2809-12, May 1998.

[7] P. G. Sherwood and K. Zeger, "Progressive Image Coding on Noisy Channels," *Proc. Data Compression Conference*, Snowbird, Utah, pp. 72-81, March 1997.

[8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[9] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Information Theory*, vol. IT-37, no. 1, pp. 155-60, January 1991.

[10] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.