Embedded Video Compression with Error Resilience and Error Concealment

Sungdae Cho[a] and William A. Pearlman[a]

[a]Center for Next Generation Video
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

ABSTRACT

This paper presents an embedded video compression with error resilience and error concealment using three dimensional SPIHT (3-D SPIHT) algorithm. We use a new method for partitioning the wavelet coefficients into spatio-temporal (s-t) blocks to get higher error resilience and to support error concealment. Instead of grouping adjacent coefficients, we group coefficients in some fixed intervals in the lowest subband to get interleaved trees. All of the sub-blocks correspond to a group of full frames within the image sequence, because each group of interleaved trees has coefficients dispersed over the entire frame group. Then we separate the stream into fixed length packets and encode every one with a channel code. Experiments show that our proposed method brings higher error resilience in noisy channels since the decoded coefficients associated with early decoding error would be spread out to the whole area along with the sequence, and we can conceal the lost coefficients with the surrounding coefficients even if some of substreams are totally missing. In addition to that, our proposed method gives higher coding performance in noiseless channels than the conventional grouping method of grouping contiguous trees.

Keywords: video compression, 3-D wavelet transform, video transmission, embedded bitstream, error resilience, error concealment, SPIHT

1. INTRODUCTION

Recently, communication has grown very rapidly, and image and video compression for noisy channels is becoming more important. Especially, the video has become a very important component of multimedia applications such as video over wireless or ATM channel, video on demand (VOD), video conferencing, and videophone over the internet. To transmit video data over noisy channels, we need to consider not only compression efficiency but also the effects of noise.

In order to achieve error resilience of embedded image bitstreams, partitioning of wavelet coefficients into many independent trees is widely used as in Ref..[1–3] This idea was first reported by Creusere.[1,4,5] Creusere used two methods to partition wavelet coefficients of EZW[6] algorithm. One is called zerotree preserving (ZP) partitioning, which gets contiguous wavelet coefficients in the lowest subband and get the other coefficients in the higher subbands while preserving zerotree structures. The other method is called offset zerotree (OZ) partitioning, which no longer preserves the conventional zerotree structure. He claims that the OZ partitioning method can be used where prematurely terminated bitstreams are existing, since the reduced-resolution coefficients will be surrounded by full-resolution coefficients at every scale. However, in both of the methods, performance degrades rapidly as number of substreams S grows due to increasing overhead and inefficient rate allocation.

In the case of embedded video bitstreams, we showed in Ref.[7] that the Spatio-Temporal Tree Preserving 3-D SPIHT (STTP-SPIHT) algorithm is highly error resilient against channel bit errors. The algorithm is based on the very effective and computationally simple 3-D SPIHT algorithm.[8,9] The idea of the STTP-SPIHT is that the more clean bits, the higher quality of video because of the embedded nature of SPIHT algorithm. To get more clean bits, we divided the wavelet coefficients into spatio-temporal (s-t) blocks, and concatenated with the RCPC coder[10] with

This material is based upon work supported by the National Science Foundation under Grant No. EEC-9812706, and IBM T. J. Watson Research Center under an IBM Faculty Award. The government has certain rights in this material.

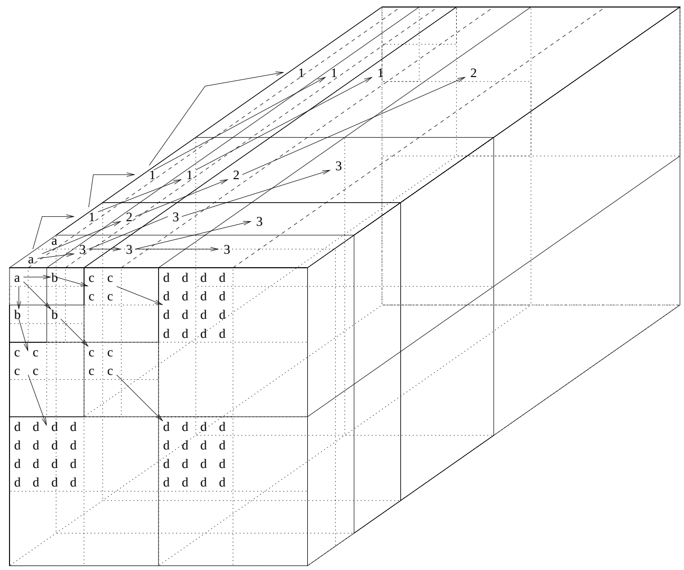

Figure 1. Structure of the spatio-temporal relation of 3-D SPIHT compression algorithm

a block interleaving scheme. As a result, the algorithm gets many more clean bits than that of the normal 3-D SPIHT algorithm. The useful property of this algorithm is that any decoding failure affects only a certain region, and doesn't propagate into any other.

However, there are two problems with the STTP-SPIHT algorithm. The first problem is the rate allocation among the spatio-temporal (s-t) tree blocks, because some s-t blocks correspond to the background which contains no motion or just little motion, and the others to high motion area. In this case, the rate should be allocated differently according to the characteristics of s-t blocks. The other problem is that very early decoding failure affects a certain region, and that region is decoded with lower resolution other regions. If the affected region contains very important motion, then the video quality would be worse.

In this paper, we use a different method for partitioning the wavelet coefficients into s-t block to solve those problems. Instead of grouping adjacent coefficients, we group coefficients at a fixed interval in the lowest subband, depending on the number of s-t blocks S. Then we track the spatio-temporal related trees of the coefficients, and merge them together. As a result, the s-t blocks of the STTP-SPIHT correspond to certain local regions, but the s-t blocks of our new grouping method correspond to the full group of frames with lower resolution. This grouping method supports error concealment of lost coefficients using surrounding coefficients in the event of decoding failure. We call this algorithm Error Resilient and Error Concealment 3-D SPIHT (ERC-SPIHT) algorithm. As with STTP-SPIHT, we separate the sub-bitstreams into fixed length packets, interleave them to obtain an embedded composite bitstream, and encode them with a rate-compatible, punctured convolutional (RCPC) error-correction code with cyclic redundancy check (CRC). This kind of channel code not only corrects errors, but also allows detection of of decoding failures, so that decoding can cease in substreams where decoding failures occur. Because the subbitsreams are embedded, the correctly received bits in each sub-bitstream can be decoded to provide a reconstruction at lower resolution or accuracy.

The organization of this paper is as follows: Section 2 shows embedded video compression with error resilience and error concealment using 3-D SPIHT algorithm. Section 3 provides simulation results. Section 4 concludes this paper.

2. EMBEDDED VIDEO COMPRESSION WITH ERROR RESILIENCE AND ERROR CONCEALMENT

2.1. Background

Figure 1 shows how coefficients in a three-dimensional (3-D) transform are related according to their spatial and temporal domains. Character 'a' represents a root block of pixels ($2 \times 2 \times 2$), and characters 'b', 'c', 'd' denote its

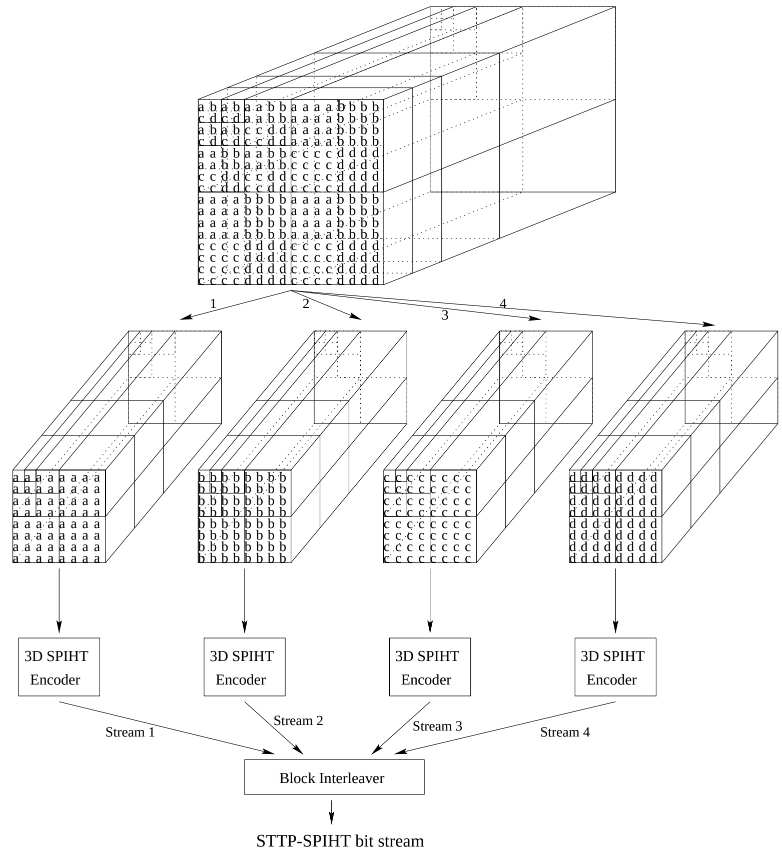

Figure 2. Structure of the Spatio-Temporal Tree Preserving 3-D SPIHT(STTP-SPIHT) compression algorithm

successive offspring progressing through the different spatial scales and numbers '1', '2', '3' label members of the same spatio-temporal tree linking successive generations of descendants. We used 16 frames in a GOF (group of frames), therefore we have 16 different frames of wavelet coefficients. We can observe that these frames not only have spatial similarity inside each one of them across the different scales, but also temporal similarity between frames, which will be efficiently exploited by the Error Resilient and Error Concealment SPIHT (ERC-SPIHT) algorithm.

In Ref.,[7] we reported the Spatio-Temporal Tree Preserving 3-D SPIHT(STTP-SPIHT) compression algorithm. Figure 2 shows the structure and the basic idea of the STTP-SPIHT compression algorithm. The STTP-SPIHT algorithm divides the 3-D wavelet coefficients into some number S of different groups according to their spatial and temporal relationships, and then to encode each group independently using the 3-D SPIHT algorithm, so that S independent embedded 3-D SPIHT substreams are created. These bitstreams are then interleaved in blocks. Therefore, the final STTP-SPIHT bitstream will be embedded or progressive in fidelity, but to a coarser degree than the normal SPIHT bitstream. In this figure, we show an example of separating the 3-D wavelet transform coefficients into four independent groups, denoted by a, b, c, d, each one of which retains the spatio-temporal tree structure of normal 3-D SPIHT,[8,9] and these trees correspond to the specific regions of the image sequences. The s-t block, which is denoted by a, matches the top-left portion in all frames of the sequence transform. The other s-t blocks correspond to the top-right, bottom-left, bottom-right fractions of the image sequences, and those s-t blocks are denoted by b, c, d, respectively. The normal 3-D SPIHT algorithm is just a case of $S = 1$, and we can flexibly choose S. When we choose the number of substream S for the image size of $X \times Y$, each of X and Y should be divisible by 2^{L+1}, where L is the number of decomposition levels. If the axis is not divisible by the number, we should extend the original image to be divisible. For the 352×240 Football sequence with 3 levels of decomposition, we can choose certain values of S from 1 up to 330.

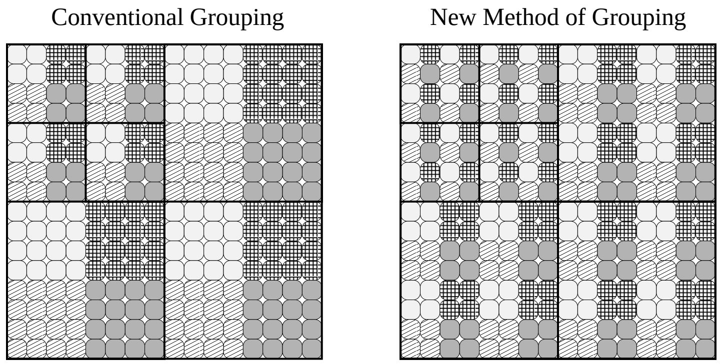

Figure 3. Graphical illustration of the 2 methods (STTP-SPIHT and ERC-SPIHT)

2.2. Proposed Method

STTP-SPIHT gave us excellent results in both noisy and noiseless channel conditions while preserving all the desirable properties of the 3-D SPIHT.[7] However, this method is also susceptible to early decoding error, and this error results in one or more small regions with lower resolution than the surrounding area. Sometimes, this artifact occurs in an important region.

We can use a different method for partitioning the wavelet coefficients into s-t blocks to solve the problem. The 3-D SPIHT compression kernel is independently applied to each tree composing the wavelet coefficients in the lowest subband and the spatially related coefficients in the higher frequency subbands. The algorithm produces sign, location and refinement information for the trees in each pass. Therefore, we need to keep the spatio-temporal related trees to maintain compression efficiency of the 3-D SPIHT algorithm. However, we do not have to keep together the contiguous wavelet coefficients in the lowest subband, since the kernel is independently applied to each tree rooted in a single lowest subband coefficients and branching into the higher frequency subbands at the same s-t orientation. In our proposed algorithm, therefore, we group the lowest subband coefficients at some fixed interval instead of grouping adjacent coefficients. This interval is determined by the number of s-t blocks S, the image dimensions, and number of decomposition levels. Then, we track the spatio-temporal related trees of the coefficients, and merge them together. Figure 3 graphically compares the two methods. The main advantage of the ERC-SPIHT is maintaining error resilience with coding efficiency. We also assign the same fixed rates to each substream. However, all of the sub-blocks contain similar information about each other, since each of the sub-blocks is composed of the coefficients not from a specific region, but from the whole region. Therefore, the fixed assignments of bitrates make more sense to our proposed method. Another nice feature of the ERC-SPIHT is that the very early decoding failure affects the whole region because the decoded coefficients would be spread out to the whole area along with the sequence, and the coefficients are concealed by the other surrounding coefficients which are decoded at a higher rate. When the decoding failure occurs in the same position, the quality of ERC-SPIHT is much better than that of STTP-SPIHT in visually and numerically (PSNR) because ERC-SPIHT algorithm itself has the function of error concealment. Therefore, the ERC-SPIHT no longer suffers from small areas which are decoded with a very low resolution.

Figure 4 shows the recovery capability of ERC-SPIHT in a worst-case example of decoding failure. We used 352×240 “Football” and “Susie” sequence, and coded at 1.0 bit/pixel with ERC-SPIHT ($S = 16$). We assumed the decoding error occurred in the beginning of the substream number 2 (second packet) for the “Football” sequence and the substream number 7 (seventh packet) for the “Susie” sequence, so that one of the substreams is totally missing. As a result, all of the wavelet coefficients which correspond to the missing substreams are set to zeros. When the inverse wavelet transform is applied to the decoder, the corresponding regions are filled with black pixels, because the decoded pixel values are zeros. In this figure, (a), (c) are the results from the ERC-SPIHT without error concealment, and (b), (d) are the results with error concealment. In this case, we just used the average values of surrounding coefficients for the missing coefficients only in the root subband. As we can see, in the case without error concealment, there are many black spots in the images. However, when we use concealment for the missing coefficients, we can recover the missing areas very well.

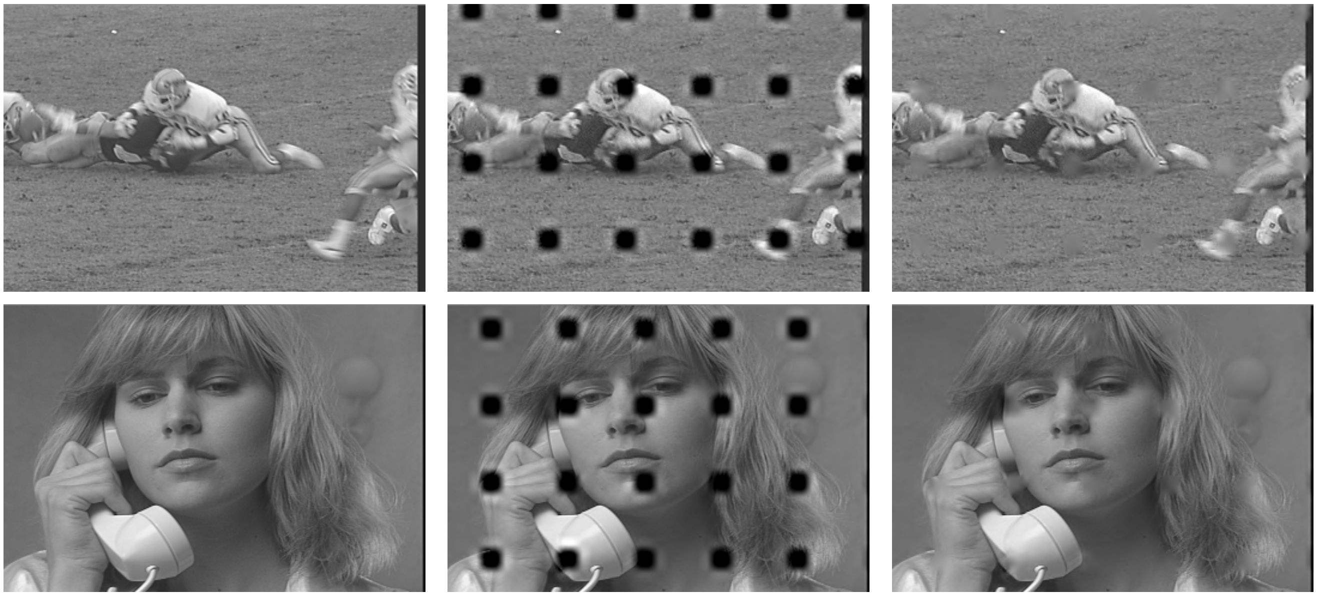

Figure 4. coded to 1.0 bit/pixel, with ERC-SPIHT ($S = 16$). (a)Top-left : 352×240 original "Football" sequence (frame 0) (b)Top-middle : without concealment after the second substream is missing (frame 0), PSNR = 17.96 (b)Top-right : with concealment (frame 0), PSNR = 29.74 dB, (d)Second row-left : 352×240 original "Susie" sequence (frame 16) (e)Second row-middle : without concealment after the seventh substream is missing (frame 16) , PSNR = 19.61 dB, (f)Second row-right : with concealment (frame 16) , PSNR = 33.83 dB

3. RESULTS

In our test of error resilience, we assume that the channel is binary symmetric (BSC). For MPEG-2, we use 15 frames in a GOP (Group Of Pictures), and the I/P frame distance is 3 (IBBPBBP...). For the normal 3-D SPIHT, STTP-SPIHT, and ERC-SPIHT, sixteen frames in a GOF (Group Of Frames) are used, and a dyadic three level transform using 9/7 biorthogonal wavelet filters[11] is applied to the image sequences. For the 3-D SPIHT and MPEG-2, we send the bitstream sequentially in 200 bit packets, and for the ERC-SPIHT, we interleave the streams in 200 bit packets to maintain embeddedness, and the receiver de-interleaves the bitstream to a series of substreams, each one of which is decoded independently. The algorithm is then tested using the $352 \times 240 \times 48$ monochrome "Football" (frame number 0-47) and "Susie" (frame number 16-63) sequences. The distortion is measured by the peak signal to noise ratio (PSNR)

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) dB, \tag{1}$$

where MSE denotes the mean squared error between the original and reconstructed image sequences. All PSNR's reported for noisy channels are averages over fifty (50) independent runs.

	Football		Susie	
MPEG-2	33.27		42.90	
3-D SPIHT	34.20		44.66	
	STTP-SPIHT	ERC-SPIHT	STTP-SPIHT	ERC-SPIHT
$S = 4$	34.04	34.04	44.37	44.50
$S = 10$	33.54	33.97	44.19	44.44
$S = 16$	33.35	33.79	43.13	44.26
$S = 55$	32.51	33.35	43.13	43.65

Table 1. Comparison of average PSNRs among 3-D SPIHT, ERC-SPIHT (S = 4, 10, 16, 55), STTP-SPIHT (S = 4, 10, 16, 55), and MPEG-2 at total transmission rate of 2.53 Mbps in noiseless channel

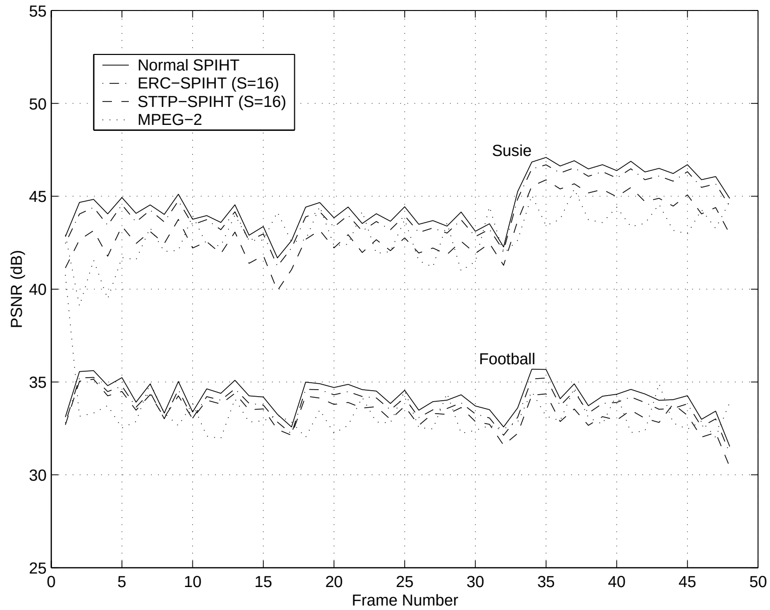

Figure 5. Comparison of frame by frame PSNR(dB) of "Football" and "Susie" sequence coded to 1.0 bit/pixel with normal 3-D SPIHT, ERC-SPIHT ($S = 16$), STTP-SPIHT ($S = 16$), and MPEG-2 without channel bit errors.

Figure 5 represents the frame by frame comparison of PSNR's of "Football" and "Susie" sequences coded with 1.0 bit/pixel (2.53 Mbps) without FEC. The solid line on top shows the PSNR values of normal 3-D SPIHT with "Susie" sequence, and the second solid line means PSNR value with "Football" sequence. The dashed dot lines represent ERC-SPIHT with $S = 16$, and dashed lines mean STTP-SPIHT with $S = 16$, and dotted lines show the MPEG-2 coded sequence. As we can see, there are just small losses (0.1-0.4 dB) with ERC-SPIHT with partitioning to 16 sub-groups, and this curve follows the normal 3-D SPIHT very closely. In "Susie" sequence, the PSNRs in STTP-SPIHT with $S = 16$ is about 1 dB worse than those of the normal 3-D SPIHT in every frame. The big difference of PSNRs is mainly due to the inefficient rate allocation among substreams, because the "Susie" sequence contains large portion of background which does not have any motion along with entire frames and the rates of these substreams should be allocated differently. However, ERC-SPIHT with $S = 16$ successfully follows the PSNR values of the normal 3-D SPIHT with just little PSNR differences.

Table 1 shows the comparison of average PSNRs among 3-D SPIHT, ERC-SPIHT ($S = 4$, 10, 16, 55), STTP-SPIHT ($S = 4$, 10, 16, 55), and MPEG-2 at total transmission rates of 2.53 Mbps of "Football" and "Susie" sequences with bit error rates (BER) of 0. As we can see in this table, the average PSNRs of ERC-SPIHT with $S = 16$ are 0.44-1.13 dB higher than those of the STTP-SPIHT with $S = 16$, and still 0.07-0.25 dB higher than those of the STTP-SPIHT with $S = 10$.

In our simulation of error resilient video transmission with error correction capability, all of the normal 3-D SPIHT, ERC-SPIHT, STTP-SPIHT and MPEG-2 bitstreams were protected identically. As in previous works,[12–15] we protected the 200 bit packets with the CRC, $c = 16$ bit parity check using generator polynomial $g(x) = X^{16} + X^{14} + X^{12} + X^{11} + X^8 + X^5 + X^4 + X^2 + 1$, and RCPC channel coder with constraint length $m = 6$. We simulated the transmission of the bitstreams over the bit error rates (BER) of $\epsilon = 0.01$ and 0.001, because the BER's of most wireless communication channels are within the range. We set the total transmission rate R_{total} to 2.53 Mbps, $r = 2/3$ for $\epsilon = 0.01$ and 8/9 for $\epsilon = 0.001$. For example, if we use a $352 \times 240 \times 16$ frames, the size of the bitstream is $R_{total} = 1,351,680$ bits (equivalently total transmission rate of 1.0 bpp with $352 \times 240 \times 16$ frames), therefore we have effective number of packets $M_1 = R_{eff}/N = \frac{Nr}{(N+c+m)N} R_{total} = (2/3)/222 \times 1351680 \approx 4060$ packets for $\epsilon = 0.01$, and $M_2 = R_{eff}/N = \frac{Nr}{(N+c+m)N} R_{total} = (8/9)/222 \times 1351680 \approx 5413$ packets for $\epsilon = 0.001$.

At the destination, the Viterbi decoding algorithm[16,17] is used to convert the packets of the received bit-stream into a SPIHT or MPEG-2 bit-stream. In the Viterbi algorithm, the "best path" chosen is the one with the lowest path metric that also satisfies the checksum equations. In other words, each candidate trellis path is first checked by computing a $c = 16$ bit CRC. When the check bits indicate an error in the block, the decoder usually fixes it by finding the path with the next lowest metric. However, if the decoder fails to decode the received packet within a certain depth of the trellis, the packet is known to decoding failure. For the ERC-SPIHT, we stop decoding for the substream where decoding failure occurs. In our test, the number of survival path was set to 100, and if none of these paths is satisfied by the CRC, then the decoder stops decoding for the substream. For MPEG-2, which is not embedded and needs the full bitstream to see the whole frames, when decoding failure occurs, we can use one of two schemes. One is just to use the corrupted packet, and the other is to put all 0's to the corrupted packet. In this paper, we use the corrupted packet itself.

Table 2 shows the comparison of average PSNRs among ERC-SPIHT ($S = 16$), MPEG-2, STTP-SPIHT ($S = 16$), and normal 3-D SPIHT with and without ARQ at the same total transmission rate of 2.53 Mbps. We can see that the average PSNR of the ERC-SPIHT in the case of $\epsilon = 0.01$ is about 5 dB higher than that of the normal 3-D SPIHT, 3-4 dB higher than that of the MPEG-2, and still 0.1-0.5 dB higher than that of the STTP-SPIHT, and in the case of $\epsilon = 0.001$ the average PSNR of the ERC-SPIHT is about 3.5 dB higher than that of the normal 3-D SPIHT, 2 dB higher than that of the MPEG-2, and also 0-0.2 dB higher than that of the STTP-SPIHT. In addition to the PSNR differences from the STTP-SPIHT, the ERC-SPIHT no longer suffers from the lower resolution in certain region that results from an early decoding error. When compared with the normal 3-D SPIHT with ARQ, the average PSNR is fairly close. However, the ARQ strategy is often inapplicable to real time scenarios.

	Football		Susie	
	0.01	0.001	0.01	0.001
ERC-SPIHT ($S = 16$)/RCPC	29.90	31.75	39.87	41.10
STTP-SPIHT ($S = 16$)/RCPC	29.61	30.77	39.67	40.27
MPEG-2/RCPC	26.35	28.98	36.66	38.87
3-D SPIHT/RCPC	24.50	28.20	34.46	37.64
3-D SPIHT/RCPC+ARQ	32.10	32.80	41.71	43.23

Table 2. Comparison of average PSNRs (dB) of "Football" and "Susie" sequence with ERC-SPIHT, STTP-SPIHT, and MPEG-2 and normal 3-D SPIHT at total transmission rate of 2.53 Mbps with Bit Error Rates (BER) of 0.01 and 0.001

Figure 6 shows the typical examples of decoded sequences for each case. From the left, each column represents in turn the 352×240 "Football" sequence (frame 15) at BER = 0.001, the 352×240 "Susie" sequence (frame 29) at BER = 0.001, and 352×240 "Football" sequence (frame 10) at BER = 0.01. In order from the top, the rows show the original sequence, the ERC-SPIHT decoded sequence, the STTP-SPIHT decoded sequence, the MPEG-2 decoded sequence and the normal 3-D SPIHT decoded sequence. As we can see, the ERC-SPIHT no longer has lower resolution region as had commonly occurred in STTP-SPIHT.

4. CONCLUSION

We have implemented embedded video compression with error resilience and error concealment. The result shows that this different method of grouping wavelet coefficient tree roots at a fixed interval and encoding these interleaved tree blocks independently provides high degrees of error resilience, error recovery, and coding efficiency.

REFERENCES

1. C. D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Transactions on Image Processing* **6**, pp. 1436–1442, October 1997.
2. J. K. Rogers and P. C. Cosman, "Wavelet zerotree image compression with packetization,," *IEEE Signal Processing Letters,* **5**, pp. 105–107, May 1998.
3. P. Cosman, J. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Transactions on Image Processing* **9**, pp. 982–993, June 2000.
4. C. D. Creusere, "Robust image coding using the embedded zerotree wavelet algorithm," *Proc. Data Compression Conference* , p. 432, March 1996.

Figure 6. First column : 352×240 "Football" sequence (frame 15), Second column : 352×240 "Susie" sequence (frame 29), Third column : 352×240 "Football" sequence (frame 13), First row : Original sequence, Second row : ERC-SPIHT decoded sequence, Third row : STTP-SPIHT decoded sequence, Forth row : MPEG-2 decoded sequence, Fifth row : normal 3-D SPIHT decoded sequence, (a)Top-left (b)Top-middle (c)Top-right (d)Second row-left : BER = 0.01, PSNR = 30.00 dB (e)Second row-middle : BER = 0.01, PSNR = 39.81 dB (f)Second row-right : BER = 0.001, PSNR = 31.82 (g)Third row-left : BER = 0.01, PSNR = 29.74 dB, (h)Third row-middle : BER = 0.01, PSNR = 38.53 dB (i)Third row-right : BER = 0.001 , PSNR = 30.44 dB, (j)Forth row-left : BER = 0.01, PSNR = 27.45 dB, (k)Forth row-middle : BER = 0.01, PSNR = 32.84 dB (l)Forth row-right : BER = 0.001 , PSNR = 29.28 dB,(m)Fifth row-left : BER = 0.01, PSNR = 24.41 dB, (n)Fifth row-middle : BER = 0.01, PSNR = 32.68 dB (o)Fifth row-right : BER = 0.001 , PSNR = 28.22 dB

5. C. D. Creusere, “A family of image compression algorithms which are robust to transmission errors,” *Proc. SPIE* **2668**, pp. 82–92, January 1996.
6. J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficient,” *IEEE Transactions on Signal Processing* **41**, pp. 3445–3562, December 1993.
7. S. Cho and W. A.Pearlman, “A full-featured, error resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (spiht) algorithm,” *to be published, IEEE Trans. on Circuits and Systems for Video Technology* .
8. B.-J. Kim and W. A. Pearlman, “An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees,” *Proc. of Data Compression Conference* , pp. 251–260, 1997.
9. B.-J. Kim, Z. Xiong, and W. A. Pearlman, “Low bit-rate scalable video coding with 3d set partitioning in hierarchical trees (3d spiht),” *IEEE Trans. Circuits and Systems for Video Technology* **10**, pp. 1374–1387, December 2000.
10. J. Hagenauer, “Rate-compatible punctured convolutional codes (rcpc codes) and their applications,” *IEEE Transactions on Communications* **36**, pp. 389–400, April 1988.
11. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Transactions Image Processing* **1**, pp. 205–220, 1992.
12. P. G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Siginal Processing Letters* **4**, pp. 189–191, July 1997.
13. P. G. Sherwood and K. Zeger, “Progressive image coding on noisy channels,” *Proc. DCC ’97* , pp. 72–81, April 1997.
14. B.-J. Kim, Z. Xiong, and W. A. Pearlman, “Progressive video coding for noisy channels,” *Proc. 1998 IEEE Int. Conf. On Image Processing (ICIP’98)* , October 1998.
15. B.-J. Kim, Z. Xiong, W. A. Pearlman, and Y. S. Kim, “Progressive video coding for noisy channels,” *Journal of Visual Communication and Image Representation* **10**, pp. 173–185, 1999.
16. G. D. Forney, “The viterbi algorithm,” *Proc. IEEE* **61**, pp. 169–176, February 1984.
17. N. Seshadri and C. Sundberg, “List viterbi docoding algorithm with applications,” *IEEE Transactions on Communications* **42**, pp. 313–323, 1994.