Multilayered Protection of Embedded Video Bitstreams over Binary Symmetric and Packet Erasure Channels

Sungdae Cho and William A. Pearlman

Center for Image Processing Research

Dept. of Electrical, Computer and Systems Engineering

Rensselaer Polytechnic Institute

110 Eighth Street

Troy, NY 12180-3590

Corresponding Author:

Prof. William A. Pearlman

Center for Image Processing Research

Dept. of Electrical, Computer and Systems Engineering

Rensselaer Polytechnic Institute

110 Eighth Street

Troy, NY 12180-3590

Tel: 518-276-6082

Fax: 518-276-8715

E-mail: pearlman@rpi.edu

Preprint submitted to Elsevier Science *9 April 2004*

Abstract

This paper presents a multilayered protection of embedded video bitstreams over bit errors and packet erasure channels using Error Resilient and Error Concealment 3-D SPIHT (ERC-SPIHT) algorithm, which is based on the 3-D SPIHT concepts. A robust source coder is created to give error resilience in source level of the codestream. This robustness is achieved by partitioning the wavelet coefficients into many independent sub-bitstreams while maintaining spatio-temporal (s-t) tree structures. For higher protection against channel noise, we use a product code. In each packet, the concatenation of a rate compatible punctured convolutional (RCPC) code and an error detecting parity check (CRC) code is used. Across the packets, Reed-Solomon codes are used. These steps provide the robust source coder with additional layers of protection against channel noise. Finally, in the decoder side, an error concealment function is performed for the lost blocks. Simulations show that the multilayered protection of 3-D SPIHT outperforms the methods that use single layer protection in terms of average PSNRs and the PSNR ranges, and provides higher average PSNR's and lower PSNR variances.

Key words: video compression, video transmission, robust source coding, error resilient transmission, joint source and channel coding, 3-D wavelet transform, embedded wavelet coding.

1 Introduction

Wavelet-based encoders have been shown to perform better than almost any other existing compression scheme. In addition, they have the nice features of being progressive and computationally simple. However, wavelet zerotree coding algorithms are, like all algorithms producing variable length codewords, sensitive to bit errors. Even a single-bit transmission error may lead to loss of synchronization between encoder and decoder execution paths, which may lead to a total collapse of decoded video quality.

There have been some attempts [12,13] to achieve error resilience with wavelet-based embedded video bitstreams over noisy channels. The main idea was to partition the wavelet coefficients into groups and encode/decode them separately with the 3-D SPIHT algorithm. The useful property of this procedure is that any decoding failure affects only a certain bitstream, and does not propagate into any other. This idea was first reported by Creusere [17] for image transmission. However, the error resilient methods in [12,13] stopped decoding for the substream whenever a decoding failure occurred, even though the probability of decoding failure was very low.

In wavelet-based encoded images for error concealment, Rogers and Cosman were able to group a variable number of blocks in one packet using a packetizable zerotree wavelet (PZW) compression scheme [5]. In the event of a single packet loss, the 8 neighbors of the missing trees are still present, because each packet is composed of trees from widely dispersed locations using a recursive tessellation technique [4]. Then, an interpolation method is used to conceal the lost wavelet coefficients. Rane *et al.* [8] developed an algorithm claimed to be

compatible with the wavelet-based JPEG 2000 image compression standard. Instead of using common retransmission query protocols, the researchers reconstruct the lost blocks in the wavelet-domain using the correlation between the lost block and its neighbors. The algorithm first uses a simple method to determine the presence or absence of edges in the lost block. This is followed by an interpolation scheme, designed to minimize the blockiness effect, while preserving the edges or texture in the interior of the block.

In previous work [13], we used the average values of surrounding coefficients for the missing coefficients only in the spatio-temporal root subband. This method conceals the lost coefficients in background very successfully. However, the decoded sequence still suffers from loss of edge information, because this method conceals only lost coefficients in the root subband and does not conceal other high frequency information in higher level subbands, where the wavelet coefficients contain most of the edge information.

In this paper, we use ERC-SPIHT [13] with a rate-compatible punctured convolutional (RCPC) code [9] and a Reed-Solomon code [6] to provide additional layers of protection against bit errors and packet erasures. Then, we present a fast error concealment scheme, borrowed from Rane *et al.*'s work with images, for the embedded video bitstream of ERC-SPIHT. In addition to using a simple averaging method in the root subband, we detect the presence or absence of edges in the lost block of every image. Then we use an interpolation scheme to recover the lost edge information. We measure the performance of error resilience over binary symmetric channel (BSC) and packet erasure channel. In addition, we introduce the ERC-SPIHT which is encoded with an asymmetric tree structure. This ERC-SPIHT gives a little higher compression ratio than that of ERC-SPIHT with normal (symmetric) tree structure.

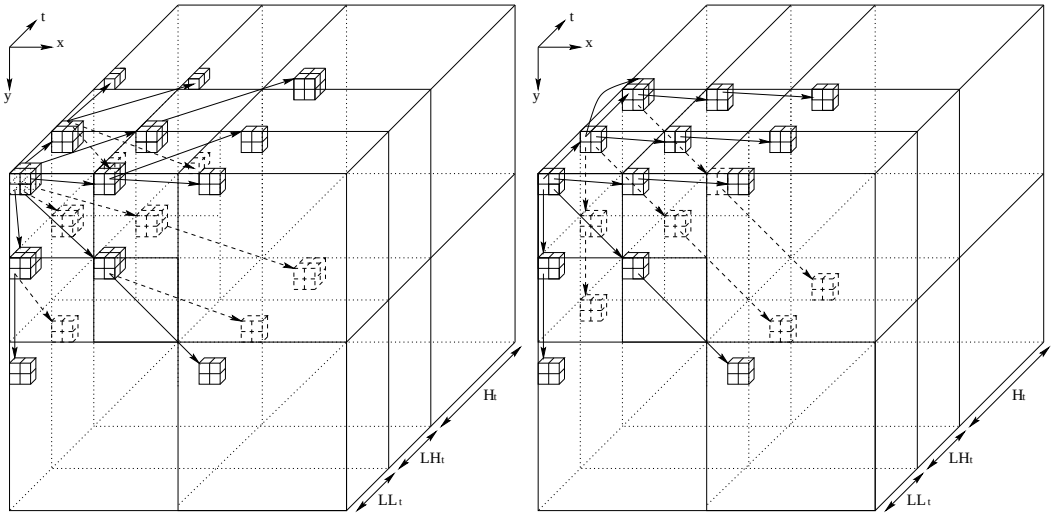

Fig. 1. Example of spatio-temporal orientation trees : 2 level wavelet packet decomposition case (21 subbands). (a) Left : symmetric tree structure, (b) : Right : asymmetric tree structure

The organization of this paper is as follows: Section 2 shows background of ERC-SPIHT. In section 3 we analyze the performance of the error resilience of the ERC-SPIHT. Section 4 shows error resilient embedded video transmission with a product channel code. Section 5 describes an error concealment method. Section 6 provides simulation results, and Section 7 concludes this paper.

2 Background of ERC-SPIHT

2.1 3-D SPIHT Symmetric Tree Structure

On the 3-D subband structure, there are 3-D spatio-temporal trees, and their parent-offspring relationships. The 3-D SPIHT algorithm efficiently exploits

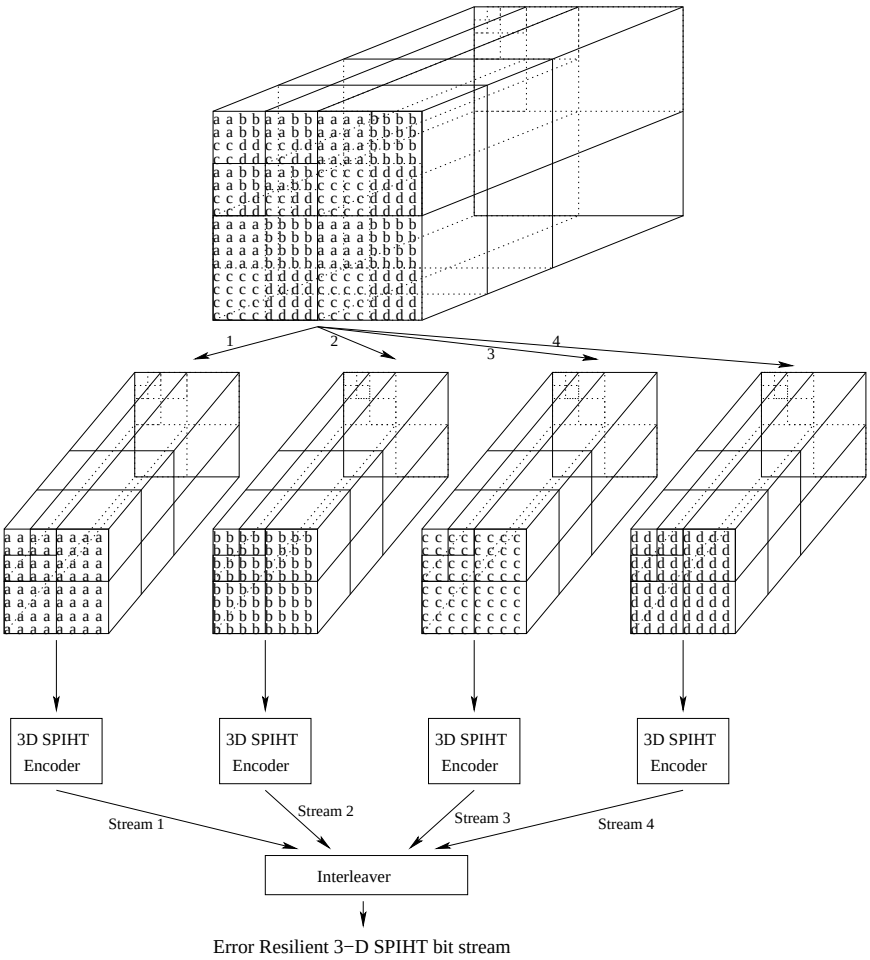

Fig. 2. Structure of the error resilient 3-D SPIHT

these spatio-temporal tree relationships. Figure 1 illustrates these relationships in the case of a two-level wavelet packet decomposition case with 21 subbands created by two levels of symmetric spatial decomposition, followed by two levels of temporal wavelet decomposition. With these 3-D spatio-temporal tree structures, the 3-D SPIHT algorithm generates completely embedded bitstreams, so that a single file for a video sequence can provide progressive video quality, that is, the algorithm can be stopped at any compressed file size. In previous works [12,13], we have used the 3-D SPIHT compression kernel with the tree structure showed in Figure 1 (a). This tree structure is just a

simple extension of the symmetric 2-D SPIHT tree structure.

2.2 3-D SPIHT Asymmetric Tree Structure

Other researchers have proposed a more efficient tree structure [2,3] as shown in Figure 1 (b). The main idea of this tree structure is to make the trees longer. The 3-D SPIHT algorithm is one of the tree based coders, and the tree-based coders tend to give better performance when tree depth is long and the distribution is lopsided, since that increases the probability of a coefficient value being zero as we move from root to leaves. To make that kind of tree, we can simply decompose into more levels. We can not always do so, since there is a limitation according to the image size and GOF size. On the other hand, the asymmetric tree structure always gives a longer tree than that of the normal 3-D SPIHT.

The major difference between the 3-D SPIHT in [19] and the asymmetric tree 3-D SPIHT, depicted in Figure 1, is that the asymmetric tree structure has 2×2 wavelet coefficients element in root subband (LL) rather than $2 \times 2 \times 2$ wavelet coefficients element, and branches differently. The top left coefficient of each 2×2 group in the lowest transaxial subband in the asymmetric tree structure branches to the 2×2 offspring group in the same relative transaxial location of the following axial subband. The branching in the directions of the spatial coordinates is the same as that for a two-dimensional wavelet transform, a single point in the lower frequency band to a 2×2 group of same orientation in the next higher frequency subband.

One potential advantage over the original symmetric tree structure is that it

can be more easily applied to a different number of decompositions between the spatial and temporal dimensions, because this tree structure is naturally unbalanced. This function is very useful when the frame size is big and temporal decomposition levels are limited. In that case, we can decompose to more levels in the spatial domain than the temporal domain. More spatial decompositions usually produce noticeable coding gain. The tree structure of this different number of decompositions can be easily extended. For higher number of temporal decompositions, the tree structure can be also extended. Kim *et al.* also used an unbalanced tree structure in [19], but it was more difficult to apply.

2.3 Grouping Method of 3-D SPIHT

In zerotree algorithms such as SPIHT, when a single bit error occurs in a bit conveying significance of a coefficient or set of coefficients, the result is loss of synchronization between the encoder and decoder, giving erroneously decoded data beyond the point of the error. In other words, all the bits received after the first error bit are unusable. To get error resilience, we partition a three-dimensional wavelet transform into independent coding units, so that an error in any one unit does not affect the others. As shown in Figure 2, we can divide 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. In this figure, we show an example of separating the 3-D wavelet transform coefficients after two levels of spatio-temporal decomposition into four independent groups, denoted by a,

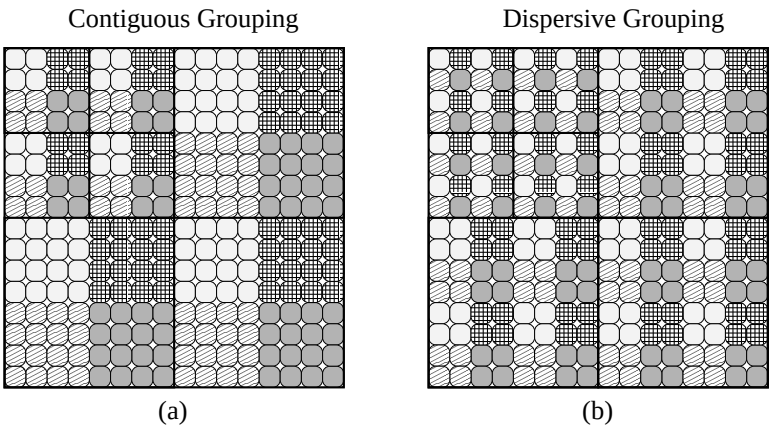

Fig. 3. Graphical illustration of the two grouping methods (a) contiguous grouping, (b) dispersive grouping

b, c, d, each one of which retains the spatio-temporal tree structure of normal 3-D SPIHT [19], and the normal 3-D SPIHT algorithm is just a case of $S = 1$, and we can flexibly choose S.

2.4 ERC-SPIHT

Figure 3 illustrates two methods of grouping ; contiguous (a) and dispersive (b) of the transformed frame. Following tree links of the asymmetric tree in Figure 1 (b) generates the corresponding 3D groups. ERC-SPIHT generates some number S of different groups with the dispersive grouping shown in Figure 3 (b), and then encodes each group independently using the 3-D SPIHT algorithm, so that S independent embedded 3-D SPIHT substreams are created. The S sub-bitstreams are then interleaved in appropriate fixed size units (e.g. bits, bytes, packets, etc.) prior to transmission so that the embedded nature of the composite bitstream is maintained. Therefore we can stop decoding at any compressed file-size or let run until nearly lossless reconstruction is obtained.

The main advantage of the ERC-SPIHT dispersive grouping is the gaining

of error resilience while maintaining coding efficiency. 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 connected local region, but dispersed over the global region. Therefore, assignment of the same bitrates to each substream is nearly optimal in this 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 defective coefficients can be concealed by the other surrounding coefficients which are decoded at a higher rate. Therefore, ERC-SPIHT does not suffer from small local areas which are decoded with very low resolution or accuracy.

3 Error Resilience of ERC-SPIHT

The most important characteristic of the SPIHT coder is embeddedness in fidelity, which means that we can get higher image quality with more uncorrupted bits. Therefore, we measure the error resilience in this section by the number of clean packets received. Our method holds the similar characteristic of error resilience as in [14,17].

3.1 Probability of Decoding Error

As explained later, we use the Viterbi algorithm to decode each packet, and the decoding error of the Viterbi algorithm occurs independently for each packet received. We assume the probability of a packet decoding failure with the Viterbi decoder is given as ξ with $\xi \ll 1$. If M is the total number of

packets, then, both the normal 3-D SPIHT and ERC-SPIHT have the same total number of packets, M. However, the ERC-SPIHT has S substreams, each consisting of $\frac{M}{S}$ packets, which totals M packets. We can express the probability of decoding error occurring in one of the first $\frac{M}{S}$ packets, $p_1(\frac{M}{S})$, of the bitstream in terms of, ξ. Hence, we have

$$\begin{aligned} p_1(\tfrac{M}{S}) &= 1 - Prob\left\{No\ Failure\ in\ (\tfrac{M}{S})\ packets\right\} \\ &= 1 - (1-\xi)^{(\frac{M}{S})} \\ &\approx (\tfrac{M}{S})\xi \end{aligned} \tag{1}$$

where $\xi \ll 1$. From this, we can see that the longer the bit-stream, the more likely it is to be decoded incorrectly, in other words, if $\frac{M}{S}$ is small, then the $p_1(\frac{M}{S})$ will be also small. Hence, we have

$$p_1(\tfrac{M}{S}) \leq p_1(M) \tag{2}$$

for $S > 1$.

3.2 Expected Number of Uncorrupted Packets

We can also find the expected number of uncorrupted packets. Let $p_2(k)$ be the probability of successful decoding of first k packets of a substream before the first packet decoding failure. Then

$$p_2(k) = \begin{cases} \xi(1-\xi)^k, & 0 \leq k < \frac{M}{S} \\ (1-\xi)^k, & k = \frac{M}{S} \end{cases} \tag{3}$$

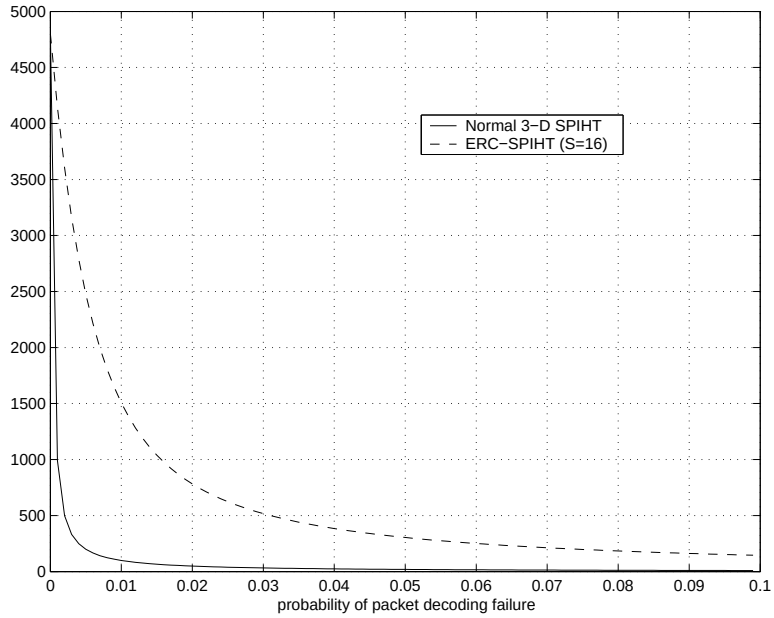

Fig. 4. Average number of uncorrupted packets vs. probability of packet decoding failure ξ

Then, the average number m of uncorrupted packets before the first packet decoding failure in S substreams is

$$m = S\sum_{n=0}^{\frac{M}{S}} np_2(n) = \begin{cases} S\sum_{n=0}^{k} n\xi(1-\xi)^n, & 0 \le k < \frac{M}{S} \\ \\ S\left(\sum_{n=0}^{k-1} n\xi(1-\xi)^n + k(1-\xi)^k\right), & k = \frac{M}{S} \end{cases} \tag{4}$$

$$= \begin{cases} \frac{S}{\xi}(1-\xi)\left[1-(1-\xi)^k(1+\xi k)\right], & 0 \le k < \frac{M}{S} \\ \\ \frac{S}{\xi}(1-\xi)\left[1-(1-\xi)^k\right], & k = \frac{M}{S} \end{cases}$$

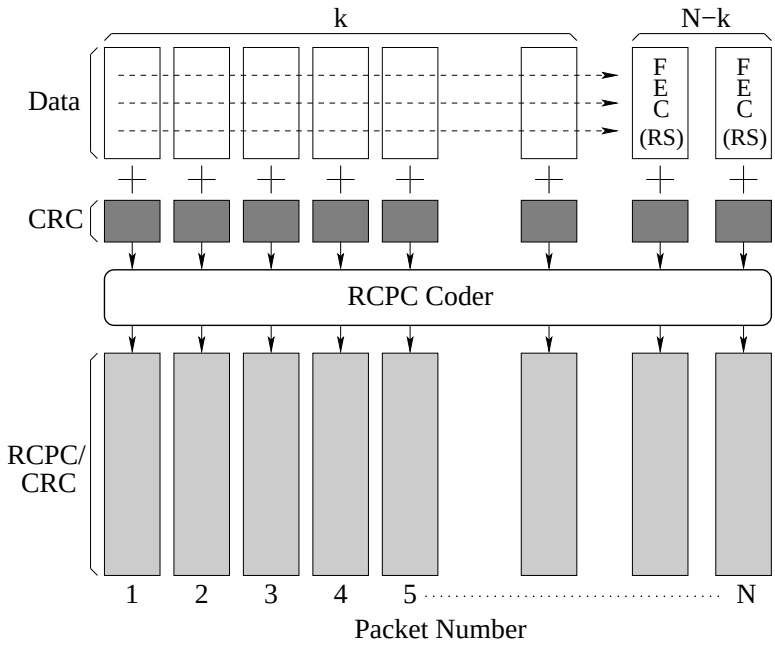

Fig. 5. RS and RCPC/CRC product code

For the normal 3-D SPIHT case, we can just set $S = 1$. Therefore, for the ERC-SPIHT, when the channel condition is favorable, in other words $\xi \approx 0$, we could get similar number of total packets as with the 3-D SPIHT, but when the channel condition becomes unfavorable, in other words $0 < \xi < 1$, then, there are many more clean packets which are correctly decoded. From the equations above, we know that partitioning wavelet coefficients to independent substreams gives more clean packets than that of the original bitstream.

Figure 4 shows the comparison of average number of uncorrupted packets m vs. probability of packet decoding failure ξ between normal 3-D SPIHT and ERC-SPIHT ($S = 16$) when total number of packets is 4800. We assume that the probability of packet decoding failure ξ ranges from 0-0.1. As we can see, we can expect many more clean packets with ERC-SPIHT as the channel becomes unfavorable.

Using ERC-SPIHT, we separate the sub-bitstreams into fixed length packets, then interleave them to obtain an embedded composite bitstream. Our goal is to minimize the decoding failures in a BSC, which occur in relatively few packets in the bitstream. To do this, we use a systematic Reed-Solomon (RS) code together with RCPC/CRC code. This method had been used by Sherwood and Zeger [15] with 2-D SPIHT compressed bitstreams to form a product code. RS codes are effective at recovering erased symbols when the locations of erased symbols are known. When packets are completely discarded, we can utilize RS codes that are optimized for erasure [1]. We can consider the decoding failures as packet erasures, and the RS code can be used to correct the erasures.

An RS code is usually labeled with (N, k), where N is the block length and k is the number of source symbols. When the code is systematic, the first k of the N encoded symbols are the source symbols and the remaining $N - k$ symbols are the parity checks. They have the property that an (N, k) code can exactly recover the k source symbols from any size k subset of the N total symbols.

Figure 5 illustrates the structure of RS and RCPC/CRC product code. First, we packetize the source bitstream, then we encode every k packets with an RS code. This step adds $N - k$ redundant packets, which makes the total number of packets N. The second step is to put a CRC code to each of N packets. The CRC provides a high probability indication of the decoding success or failure. The last step is to encode each packet with RCPC coder. We can see these

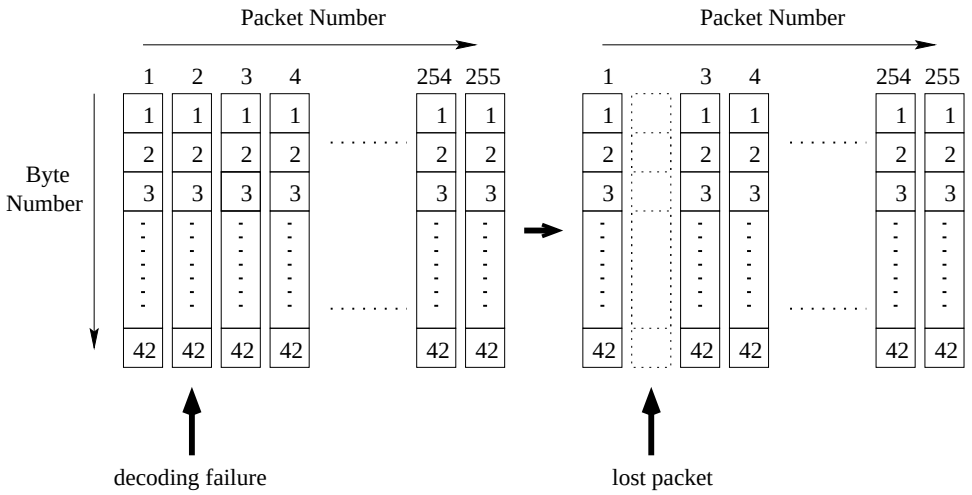

Fig. 6. example of decoding failure at packet number 2

steps in Figure 5. As we can see, the RS code acts on the rows, and the CRC and RCPC codes on the columns.

In the the decoder, RS decoding is necessary only when a decoding failure is detected in a column. From this idea, the decoded data bits in the packets across a row can be used immediately if no decoding failure is detected, because data packets are sent first, and then the error-protected packets of the RS coder are sent later. Therefore we can eliminate the delay cost when the channel condition is good.

Figure 6 shows an example of a decoding failure occurring at packet number 2 encoded by RCPC/CRC code and a systematic RS code (255,254). In this figure, the numbers (1-42) in a column are the byte numbers and the number (1-255) in a row are the packet numbers, and the packet number 255 contains parity check information. The reason to use the packet size of 42 bytes is that we use 200 data bits in every packet, and add 24 bits of CRC check bits and memory bits. When we apply the channel code with rate 2/3, the channel coded packet size becomes 42 bytes.

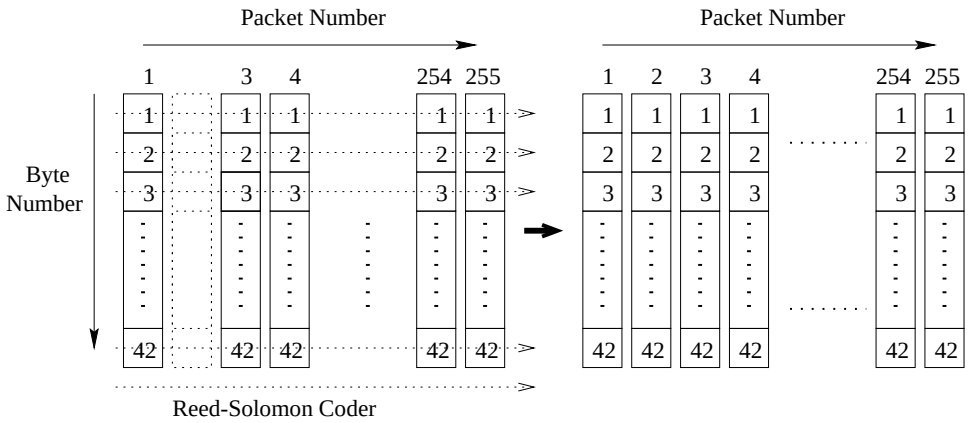

Fig. 7. example of recovering the erased data using Reed-Solomon code

Figure 7 illustrates the example of recovering erased data using the Reed-Solomon code. In this case, the decoder knows the packet number in which decoding failure occurs. The RS coder considers this corrupted packet as a packet erasure, and can recover the erased packet, since the number of packet erasures, in this case one, is no greater than the number of parity-check symbols. We know that if decoding failure occurs in 255th packet, we do not have to use the RS decoder, because the code is systematic. Therefore, the RS decoder is needed only when decoding failure occurs between packet number 1 and 254.

5 Error Concealment Method

In previous work [13], we concealed the missing blocks only in the spatio-temporal root subband using the average value of coefficients surrounding the lost coefficients before the inverse wavelet transformation. The concealment is done just before the inverse spatio-temporal wavelet transformation. In sixteen frames with three levels of spatio-temporal decomposition, the root subband comprises the first two wavelet coefficient frames. As we saw in previous work

[13], root subband averaging successfully recovered most of the background image. However, we could not recover the detail information of the image when the missing block contains an edge.

When we apply wavelet transformation in this paper we follow the steps as in Figure 8, so that spatial transformation precedes temporal transformation. Any early decoding failure or missing substream affects the related spatio-temporal tree block in wavelet domain, and the block affects a full GOF (Group Of Frames). The similar idea was used in the coding of Region of Interest (ROI) associated with the STTP-SPIHT algorithm in Ref. [7] that used the contiguous grouping. As we can see, the corresponding coefficients of all the spatio-temporal orientation trees in any substream are located in the same spatial location in each coefficient frames. Therefore, inverse temporal transformation will not affect other cleanly recovered coefficients, because the transformation uses one-dimensional filtering only in the temporal direction. In the decoder, therefore, we apply the concealment step right after the inverse temporal transformations. The next step is the inverse spatial transformation, and this transformation is applied to each individual coefficient frame. In this way, we can conceal the error effects in the root subband in each frame rather than conceal only those in the spatio-temporal root subband as in Ref. [13].

However, there are still some blocks which are not successfully recovered by this lowest spatial subband concealment in each coefficient frame. Most un-recovered blocks are parts of edges, and the edge information is in the high frequency subbands. Therefore, it is necessary to conceal in the high frequency subbands to recover detailed information of the image.

As Rane *et al.* did in Ref. [8], we utilize concealment in the high frequency

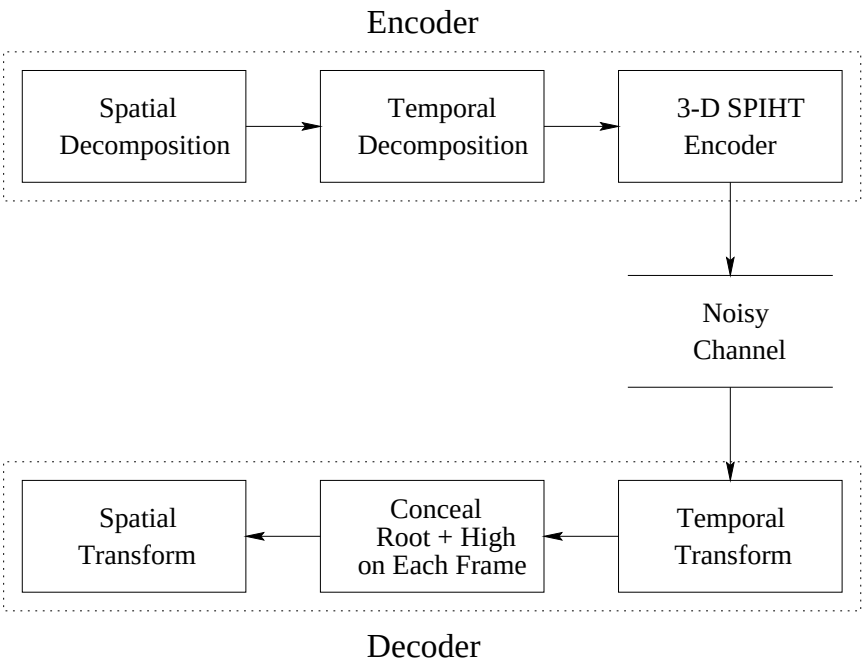

Fig. 8. Block diagram of 3-D SPIHT encoding/ decoding structure with different order of wavelet transformation and error concealment

subband in addition to the root subband concealment. The magnitude of a wavelet coefficient in a specific location tells both the amount of change and the spatial location at which the change occurs. Each level of the wavelet transform domain has three different subbands except at the lowest level, and each subband has different characteristics, such as vertical (V), horizontal (H), or diagonal (D) detail. Figure 9 (a) shows this idea for a three level decomposed image.

To recover the lost edge information, we test the missing block to see whether the block contains part of an edge or not. In Figure 9 (b), we show an example of a test of vertical edges in V1 level. A missing substream affects the 2×2 coefficient group with a fixed interval, I_1 in V1 level. In that case, we just need the coefficients right above and right below the missing one, and test whether the values are greater than certain threshold value. If any of four tests meets the condition, then we decide the missing block is part of a vertical edge, and

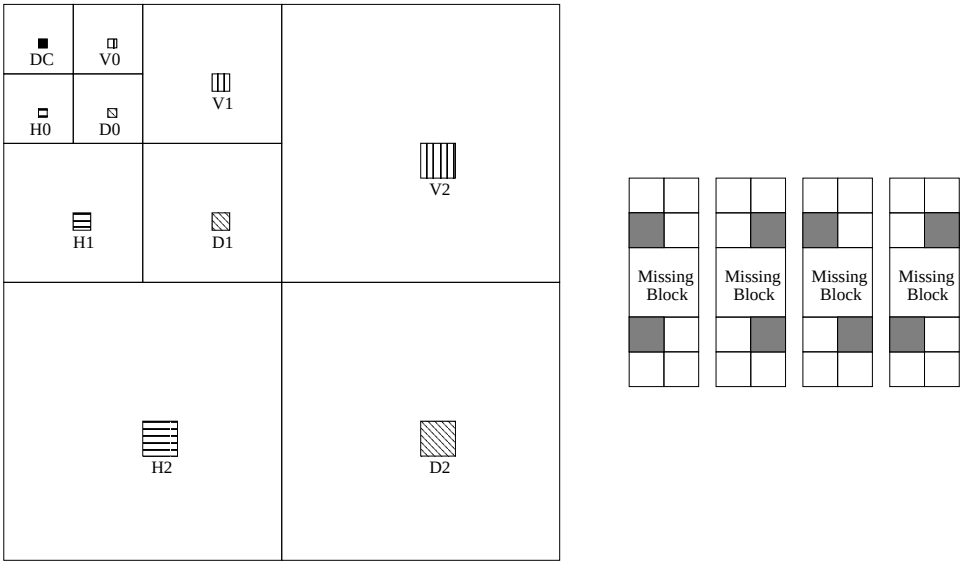

Fig. 9. Graphical illustration of error concealment in the case of missing block is in part of vertical edge. (a) Left : Characteristics of vertical (V0, V1, V2), horizontal (H0, H1, H2) or diagonal (D0, D1, D2) detail in each subband, (b) Example : Test of vertical edge

apply linear least squares interpolation. Analogously, the coefficients left and right of the missing block are needed to recover horizontal detail, and diagonal neighboring coefficients are needed to recover the diagonal detail.

For level 2, we can still expand the equation in Ref. [8] to estimate the lost coefficient in that level. However, in this paper, we conceal errors only in levels 1 and 2 in addition to those in the root subband. This additional high frequency subband concealment almost always gives successful reproduction of video over noisy channels even when some of substreams are totally missing.

6 Results

In this section, we provide the results of comparison among the 3-D SPIHT, ERC-SPIHT, and MPEG-2 in a noiseless channel. We shall hereafter call ERC-

SPIHT(ST) for ERC-SPIHT with symmetric tree structure. We also provide the PSNR results of different number of decomposition levels between the spatial and temporal dimensions. Then, we provide the simulation results in a noisy channel and compare the ERC-SPIHT with product code with other error resilient methods [13,18]. For the ERC-SPIHT, we use the described concealment method for the missing substream and the substreams that have early decoding failure. For the edge test, we determine the threshold value after testing with a number of trials. This threshold value is used to see whether the missing block contains part of edge or not. The product code consists of an outer Reed-Solomon code over GF(256) followed by an inner RCPC/CRC code, as illustrated in Figure 5. Therefore, packets are bytes. We can recover any one packet erasure from 255 packets if we know the erased packet number. In previous works [12,13] without the RS code, we transmitted 4060 packets for BER of 0.01 and 5413 packets for BER of 0.001 over 2.53 Mbps channel with frame rate of 30 Hz. When we use RS (255,254) coder, 16 additional parity-check packets and 21 additional parity-check packets are needed for BER of 0.01 and 0.001 respectively. Therefore, there will be 4044 and 5392 data packets for BER of 0.01 and 0.001 respectively. That means the product code lowers the source data rate in a noiseless channel only about 0.002 bpp and 0.003 bpp compared with the ERC-SPIHT with RCPC coder for BER of 0.01 and 0.001 respectively. We have used the same channel conditions as in our previous works [12,13].

For video transmission over channels with bit errors and packet erasures, we assume the packets suffer from both bit errors and packet erasures. We use the packet erasure model as used in Cosman *et al*'s work [16]. The packet losses are typically bursty. Therefore, in addition to the probability of packet erasure

$P_{erasure}$, the packet erasure model includes a packet burst length parameter (i.e. the number of consecutive erased packets), N. Within the model, the source output is divided into groups of N packets and each group is erased with probability P where

$$P = \frac{P_{erasure}}{N}$$

so that the overall erasure rate is $P_{erasure}$, regardless of the burst length. To test the effect of burst packet erasures, burst lengths $N = 1$, $N = 5$, and $N = 10$ were simulated with packet erasure rates of 1 %, 10 %, and 20 %.

	Football	Susie
MPEG-2	33.27	42.90
3-D SPIHT [1]	34.20	44.66
3-D SPIHT [2]	34.33	44.72
ERC-SPIHT(ST) (S=16)	33.64	44.13
ERC-SPIHT [1] (S=16)	33.96	44.31
ERC-SPIHT [2] (S=16)	34.01	44.36

Table 1

Comparison of PSNRs (dB) for $352 \times 240 \times 48$ "Football" and "Susie" sequence (gray) using MPEG-2, 3-D SPIHT with symmetric tree structure (3-D SPIHT [1]), 3-D SPIHT with asymmetric tree structure (3-D SPIHT [2]), ERC-SPIHT with symmetric tree structure (ERC-SPIHT(ST)), with asymmetric tree structure (ERC-SPIHT [1]), and with different number of decompositions (ERC-SPIHT [2]), at total transmission rate of 2.53 Mbps in noiseless channel

6.1 Compression Performance in Noiseless Channel Condition at Total Transmission Rate of 2.53 Mbps

Table 1 shows the comparison of PSNRs (dB) in a noiseless channel among the MPEG-2, 3-D SPIHT with symmetric tree structure (3-D SPIHT [1]), 3-D SPIHT with asymmetric tree structure (3-D SPIHT [2]), ERC-SPIHT with symmetric tree structure (ERC-SPIHT(ST)), with asymmetric tree structure (ERC-SPIHT [1]), and with different number of decomposition (ERC-SPIHT [2]). For the case of different number of decomposition, we kept the same number of temporal decomposition (three level), and used one more level (four) of spatial decomposition. As we can see, the PSNRs of ERC-SPIHT are about 0.1-0.3 dB higher than those of the ERC-SPIHT(ST), and the PSNRs of ERC-SPIHT and different number of decomposition are about 0.2-0.4 dB higher than those of the ERC-SPIHT(ST), and comparable to those of the original 3-D SPIHT. ERC-SPIHT with the same number of decompositions and the same tree structure as 3-D SPIHT will have slightly lower performance due to the extra overhead needed for sub-bitstream headers. When we compare with MPEG-2, the average PSNRs of ERC-SPIHT (S=16) are about 0.7 - 1.5 dB higher than those of the MPEG-2.

6.2 Video Transmission over Channel with Bit Errors

Table 2 shows the comparison of average PSNRs (dB) of $352 \times 240 \times 48$ monochrome "Football" and "Susie" sequence over fifty (50) independent runs among 3-D SPIHT/RCPC, ERC-SPIHT/RCPC ($S = 16$), ERC-SPIHT/RCPC+RS ($S = 16$), and MPEG-2/RCPC at total transmission rate of 2.53 Mbps with

	Football		Susie	
	0.01	0.001	0.01	0.001
3-D SPIHT/RCPC	24.50	28.20	34.46	37.64
ERC-SPIHT ($S = 16$)/RCPC	30.23	31.81	40.31	41.76
ERC-SPIHT ($S = 16$)/RCPC+RS	30.74	32.38	41.42	42.52
3-D SPIHT/RCPC+ARQ	32.10	32.80	41.71	43.23
MPEG-2/RCPC	26.35	28.09	36.66	38.87

Table 2

Comparison of average PSNRs (dB) of $352 \times 240 \times 48$ gray "Football" and "Susie" sequence over 50 independent trials among 3-D SPIHT/RCPC, ERC-SPIHT/RCPC ($S = 16$) with error concealment, ERC-SPIHT/RCPC+RS ($S = 16$) with error concealment, and MPEG-2/RCPC at total transmission rate of 2.53 Mbps with bit error rates (BER) of 0.01 and 0.001

bit error rates (BER) of 0.01 and 0.001. As we can see, the ERC-SPIHT with product code gives very high PSNRs, and results comparable to the 3-D SPIHT with RCPC and ARQ method, which assumes that there exists an ideal return channel. On the other hand, our ERC-SPIHT with product code does not need any return channel, and needs only a small delay on the decoder side for RS decoding. However, this delay happens only in the event of a decoding failure. When we compare with MPEG-2, the average PSNRs of ERC-SPIHT with product code are about 3.5 - 4.5 dB higher than those of the MPEG-2. In addition to the PSNR differences from the MPEG-2, the ERC-SPIHT gives visually better results than MPEG-2, because ERC-SPIHT no longer suffers from small areas which are decoded with a very low resolu-

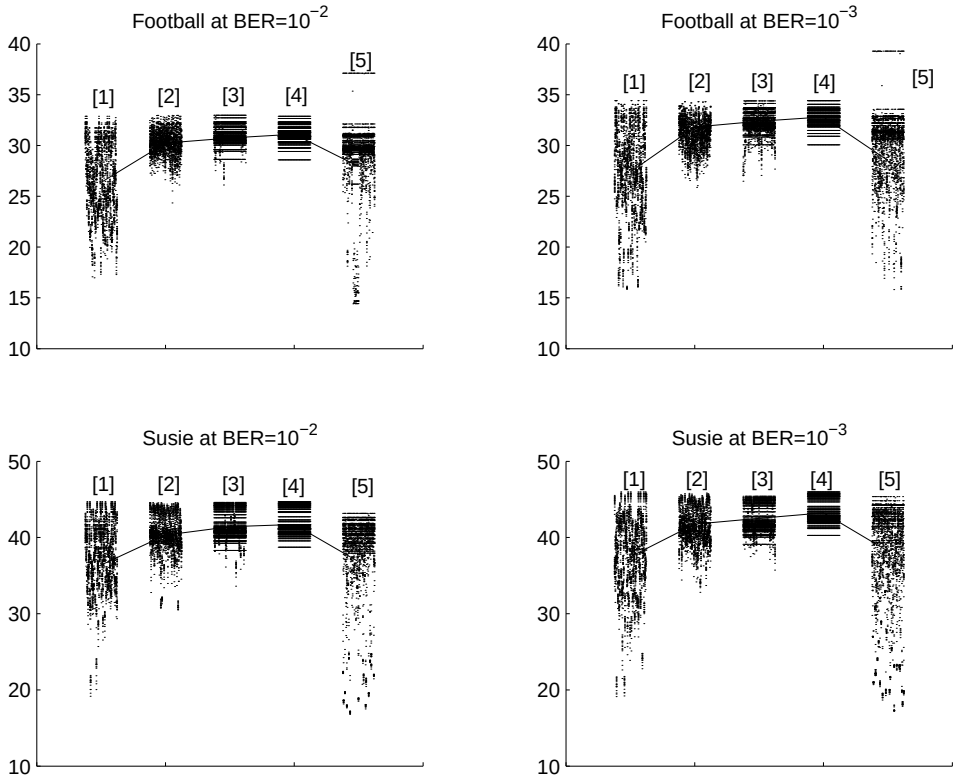

Fig. 10. Comparison of range of PSNR values among 3-D SPIHT/RCPC [1], ERC-SPIHT/RCPC [2], ERC-SPIHT/RCPC+RS [3], 3-D SPIHT/RCPC+ARQ [4], and MPEG-2/RCPC [5].

tion associated with the contiguous grouping. However, the decoding failure in MPEG-2 affects some block, and the block is filled with some other picture's block or black pixels.

Figure 10 shows the PSNR ranges of fifty trials. As we can see, MPEG-2 has much wider range of PSNRs than the ERC-SPIHT. In addition to the higher average PSNRs of ERC-SPIHT over MPEG-2, therefore, ERC-SPIHT gives reliable PSNRs in every frame, so that we can always expect a certain quality of video.

Table 3 and Table 4 show the comparison of average PSNRs (dB) of $352 \times 240 \times 16$ monochrome "Football" sequence over fifty (50) independent runs between 3-D SPIHT/RCPC+RS and ERC-SPIHT/RCPC+RS ($S = 16$) over bit errors and packet erasure channels. We used the RCPC code rate 2/3 for bit error 0.01 with total transmission rate of 2.68 Mbps, and 8/9 for bit error 0.001 with total transmission rate of 2.01 Mbps. For the RS code, we used (255,247) for packet erasure rate of 1 %, (255,219) for packet erasure rate of 10 %, and (255,191) for packet erasure rate of 20 %. Table 3 shows the average PSNRs when BER is 0.01, and Table 4 shows the average PSNRs when BER is 0.001. When the burst length is one, the 3-D SPIHT and ERC-SPIHT give similar average PSNRs over different packet erasure rates. However, as the burst length gets longer, we can find that the average PSNRs of ERC-SPIHT are 3 to 12 dB higher than those of normal 3-D SPIHT. As we can see, ERC-SPIHT gives better performance than normal 3-D SPIHT as burst length length parameter N gets larger, because the packets are interleaved when we transmit, so that the burst packet erasures are distributed among many substreams.

6.4 Performance of Error Concealment

To show the performance of error concealment, we assumed the decoding error occurred in the beginning of the substream number 7 (seventh packet) for the 352×240 "Football" sequence as we showed in previous work [13]. Figures 12 shows the sample sequence to compare the results after the error concealment.

Probability of Packet Erasure			1%	10%	20%
Burst Length	1	3-D SPIHT	31.63	29.58	30.18
		ERC-SPIHT	31.42	30.86	29.88
	5	3-D SPIHT	27.52	27.13	20.96
		ERC-SPIHT	31.42	30.86	30.32
	10	3-D SPIHT	25.76	25.43	18.38
		ERC-SPIHT	31.56	30.79	30.35

Table 3

Comparison of average PSNRs (dB) of $352 \times 240 \times 16$ gray "Football" sequence over 50 independent trials among 3-D SPIHT/RCPC+RS and ERC-SPIHT/RCPC+RS ($S = 16$) with error concealment at total transmission rate of 2.68 Mbps with bit error rates (BER) of 0.01

In all figures, the left column shows the results from the ERC-SPIHT without error concealment, the middle column the results with root subband error concealment, and the right column the results with root and high subband error concealment. As we can see, in the case of no concealment, there are many black spots in the images. When we use only root subband concealment for the missing coefficients, we can recover the missing areas very well, but there are still some spots where recovery is not successful, and these spots are marked by a rectangle. In Figure 12 (b), the player's right shoulder, in (e), the player's head and another players left shoulder, and in (h), the player's same left shoulder and another player's line on his uniform are not recovered successfully. In (c), (f), and (i) of the figure, we can see the edges are nearly perfectly recovered by root and high subband concealment.

Probability of Packet Erasure			1%	10%	20%
Burst Length	1	3-D SPIHT	31.52	30.12	29.08
		ERC-SPIHT	31.47	30.54	29.80
	5	3-D SPIHT	27.79	27.37	26.99
		ERC-SPIHT	31.56	30.68	30.36
	10	3-D SPIHT	26.21	25.24	25.84
		ERC-SPIHT	31.56	30.92	30.41

Table 4

Comparison of average PSNRs (dB) of $352 \times 240 \times 16$ gray "Football" sequence over 50 independent trials among 3-D SPIHT/RCPC+RS and ERC-SPIHT/RCPC+RS ($S = 16$) wit error concealment at total transmission rate of 2.01 Mbps with bit error rates (BER) of 0.001

7 Conclusion

In this paper, we presented a multilayered protection of embedded video bitstreams over bit errors and packet erasure channels. Figure 11 shows this idea. In Figure 11, top layer in encoding process represents the robust source encoding part using the ERC-SPIHT algorithm. We achieve the robustness by processing independent s-t blocks. In addition to that, we showed that the ERC-SPIHT could achieve higher compression performance with asymmetric tree structure in noiseless channel condition. The following two layers show channel encoding by the product code. The bottom layer in the decoding process is RCPC/CRC decoder. In this layer, we can correct channel bit errors using RCPC code, and also detect the correction failure, called decoding fail-

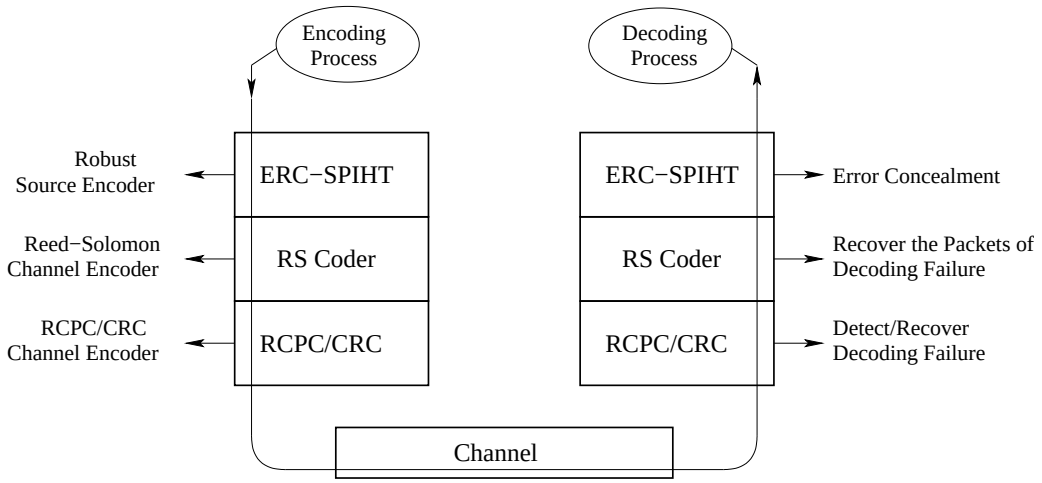

Fig. 11. Multilayered protection of embedded video bitstream

ure using CRC code. The failed packets could be recovered in the upper layer using Reed-Solomon code. Furthermore, the ERC-SPIHT gives steadily good performance over a packet erasure channel that is nearly constant with degree of packet loss severity. If there are still some blocks that are not recovered from bottom two layers, we could apply the top layer of the error concealment step. This step conceals most lost edge information as well as background. The concealment is obtained at a very low computational complexity. Therefore, it can be implemented in real time applications. Using this layered protection, we could get a similar result as 3-D SPIHT with RCPC and ARQ, which needs an ideal return channel. We also showed that the multilayered protection of 3-D SPIHT bitstream gave better results than MPEG-2 in both noiseless and noisy channel conditions, even though ERC-SPIHT uses no motion estimation/compensation as does MPEG-2.

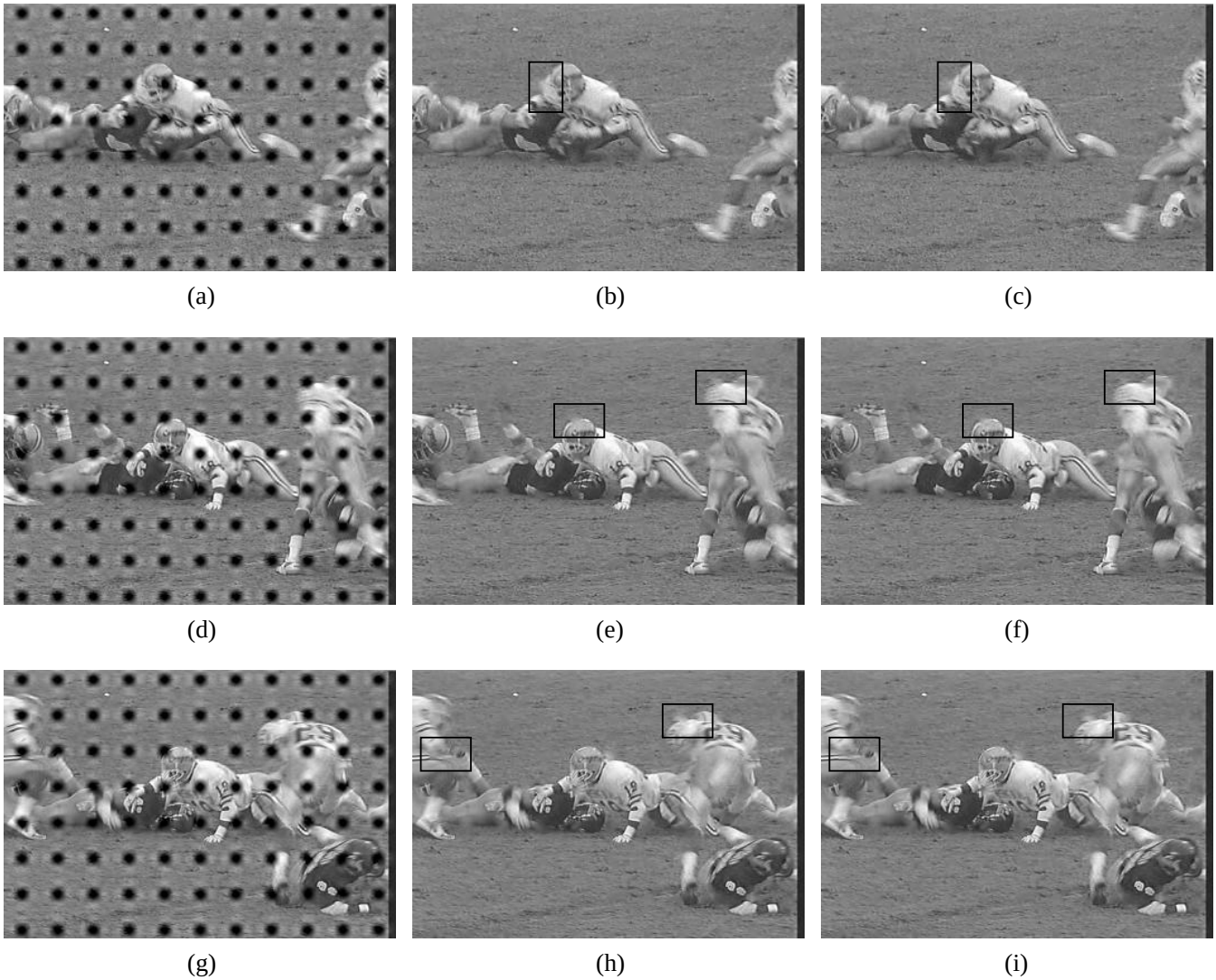

Fig. 12. 352×240 "Football" sequence coded to 1.0 bit/pixel, with ERC-SPIHT (S = 16). (a) frame 0 : without concealment after the seventh substream is missing, PSNR = 17.69 dB, (b) with root subband concealment, PSNR = 30.91 dB, (c) with root + high subband concealment, PSNR = 30.93 dB, (d) frame 4 : without concealment after the seventh substream is missing, PSNR = 17.63 dB, (e) with root subband concealment, 31.07 dB, (f) with root + high subband concealment, 31.17 dB, (g) frame 7 : without concealment after the seventh substream is missing, PSNR = 17.60 dB, (h) with root subband concealment, 29.63 dB, (i) with root + high subband concealment, 29.64 dB

References

[1] L. Rizzo, *Effective erasure codes for reliable computer communication protocols*, ACM Computer Communication, vol. 27, 1997, pp. 24-36.

[2] C. He, and J. Dong, Y. F. Zheng, and Z. Gao, *Optimal 3-D Coefficient Tree Structure for the 3-D Wavelet Video Coding*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 10, 2003, pp. 961-972.

[3] P. L. Dragotti, and G. Poggi, and A. R. P. Ragozini, *Compression of multispectral images by three-dimensional SPIHT algorithm*, IEEE Transactions on Geoscience and Remote Sensing, vol. 38, no. 1, 2000, pp. 416-428.

[4] R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, Mass., 1987

[5] J. K. Rogers and P. C. Cosman, *Wavelet zerotree image compression with packetization*, IEEE Signal Processing Letters, vol. 5, no. 5, 1998, pp. 105-107.

[6] S. Lin, and Jr. D. J. Costello, *Error Control Coding*, Prentice Hall Inc. Eaglewood Cliffs, New Jersey, 1983.

[7] S. Cho and W. A.Pearlman, *Region-Based SPIHT Coding and Multiresolution Decoding of Image Sequences*, Proceedings of Picture Coding Symposium, PCS-2001, 2001, pp. 283-286.

[8] S. Rane, J. Remus, and G. Sapiro, *Wavelet-Domain Reconstruction of Lost Blocks in Wireless Image Transmission and packet switched networks*, IEEE International Conference on Image Processing, vol. 1, 2002, pp. 309-312.

[9] J. Hagenauer, *Rate-compatible punctured convolutional codes (RCPC codes) and their applications*, IEEE Transactions on Communications, vol. 36, 1988, pp. 389-400.

[10] J. M. Shapiro, *Embedded image coding using zerotrees of wavelet coefficient*, IEEE Transactions on Signal Processing, vol. 41, 1993, pp. 3445-3462.

[11] A. Said and W. A. Pearlman *A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, 1996, pp. 243-250.

[12] 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*, IEEE Transactions on Circuits and Systems for Video Technology (CSVT), vol. 12, no 3, 2002, pp. 157-171.

[13] S. Cho and W. A. Pearlman *Error Resilience and Recovery in Streaming of Embedded Video*, Signal Processing, Special Issue on Image and Video Coding Beyond Standard, vol. 82, issue 11, 2002, pp. 1545-1558.

[14] P. G. Sherwood and K. Zeger *Progressive Image Coding on Noisy Channels*, Proc. DCC, 1997, pp. 72-81.

[15] P. G. Sherwood and K. Zeger *Error Protection for Progressive Image Transmission over Memoryless and Fading Channels*, Proceedings of International Conference on Image Precessing (ICIP), vol. 1, 1998, pp. 324-328.

[16] P. C. Cosman, J. K. 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 of Image Processing, vol. 9, issue 6, 2000, pp. 982-993.

[17] C. D. Creusere *A New Method of Robust Image Compression Based on the Embedded Zerotree Wavelet Algorithm*, IEEE Transactions on Image Processing, vol. 6, no. 10, 1997, pp. 1436-1442.

[18] Z. Xiong, B.-J Kim, and W. A. Pearlman *Progressive video coding for noisy channels*, In Proc. IEEE International Conference on Image Processing (ICIP '98), vol. 1, 1998, pp. 334-337.

[19] B.-J Kim, Z. Xiong, and W. A. Pearlman *Low bit-rate scalable video coding with 3-D Set partitioning in Hierarchical Trees (3-D SPIHT)*, IEEE Trans. Circuits and Systems for Video Technology, vol. 10, 2000, pp. 1374-1387.