Abstract

We extend the work of Sherwood and Zeger [1, 2] to progressive video coding for noisy channels. By utilizing a three-dimensional (3-D) extension of the set partitioning in hierarchical trees (SPIHT) algorithm [3], we cascade the resulting 3-D SPIHT video coder [4, 5] with the rate-compatible punctured convolutional (RCPC) channel coder [6] for transmission of video over a binary symmetric channel (BSC). Progressive coding is achieved by increasing the target rate of the 3-D embedded SPIHT video coder as the channel condition improves. The performance of our proposed coding system is acceptable at low transmission rate and bad channel conditions.

1 Introduction

Transmission of compressed images/video over a noisy channel may suffer from disturbance or channel noise. The result is usually uncontrolled degradation in reproduction quality, especially when variable-length coding (VLC) is used for high performance compression. Therefore, a major concern of the designer is the control of errors so that reliable transmission can be obtained.

In recent years, we have witnessed a great deal of effort expended on the problem of devising efficient error control methods in a noisy environment, since Shannon's demonstration [7] that, by proper encoding of the information, errors can be reduced to any desired level without sacrificing the rate of information transmission. In general, we can categorize the error control schemes into two classes, *forward error correction* (FEC) and *automatic repeat request* (ARQ). Error control of a one-way system must be accomplished by FEC, that is, by employing error-correcting codes that automatically correct errors detected at the receiver. The principle is that FEC adds a certain amount of redundancy to the data stream to protect it from channel errors. A typical one-way information transmission system is depicted in Fig. (1).

In some cases, a transmission system can be two-way; that is, information can be sent in both directions and the transmitter also acts as a receiver. In a two-way communication system, error control can be accomplished using the ARQ strategy. In an ARQ system, when errors are detected at the receiver, a request is sent for the transmitter to repeat the message, and this continues until the message is received correctly. Examples of ARQ strategies include *stop-and-wait, go-back-N*, and *selective-repeat* [8].

The major advantage of ARQ over FEC is that error detection requires much simpler decoding equipment than does error correction. Also, ARQ is adaptive in the sense that information is retransmitted only when errors occur. However, when the channel error rate is high, retransmission requests must be sent frequently, and the system throughput, the rate at which newly generated messages are correctly received, is lowered. In that case, a hybrid FEC/ARQ strategy, a combination of ARQ for the less likely error patterns and FEC for the most frequent error patterns, will be more efficient than ARQ alone.

In the past, various methods have been proposed for image transmission over noisy channels using both FEC [1, 9, 10] and hybrid FEC/ARQ schemes [11]. Among them, Sherwood and Zeger's SPIHT-RCPC scheme [1, 2], which cascades the SPIHT image source coder with a RCPC channel coder, gives very promising results. They demonstrate that by choosing the best available source and channel coders independently, and wisely selecting the corresponding source and channel code rates, one can obtain most of the gains of joint source channel coding.

Although progressive video transmission over noisy channels and heterogeneous networks has widespread applications, to our best knowledge, there are only a few published results in this area (e.g., [12]). In this paper, we extend Sherwood and Zeger's work from images to video with a hybrid FEC/ARQ scheme using a RCPC code [6] combined with an ARQ strategy to protect the 3-D SPIHT bit-stream from being corrupted by channel errors [13]. We assume that a noiseless return channel with narrow bandwidth is available. Experiments show that our proposed hybrid system is acceptable at low transmission rate and bad channel conditions.

2 Set partitioning in hierarchical trees (SPIHT)

Using similar ideas of embedded zerotree coding (EZW) [14], the two-dimensional (2-D) SPIHT algorithm in [3] is based on viewing wavelet coefficients as a collection of spatial orientation trees, with each tree consisting of coefficients from all subbands that correspond to the same spatial location in an image (see Fig. 2 (a)). A set of coefficients in the tree is significant if the largest one (in magnitude) in the set is greater than or equal to certain threshold (e.g., a power of two), otherwise, it is insignificant. Similarly, a coefficient (or pixel) is significant if its magnitude is greater than or equal to the threshold, otherwise, it is insignificant. SPIHT is a continuous refinement algorithm with the significance of a larger set in the spatial orientation tree being tested at first, followed by its subsets (or child sets). The threshold is halved whenever all the coefficients are tested. SPIHT can thus be thought of as a bit-plane coding scheme. With the sign bits and refinement bits (for coefficients that become significant earlier) being coded on the fly, SPIHT achieves embedded coding in the wavelet domain using three lists: the list of significant pixels (LSP); the list of insignificant pixels (LIP); and the list of insignificant sets (LIS). The 2-D SPIHT coder performs competitively with most other coders published in the literature [15], while having nice features such as relatively low complexity and rate embeddedness. It represents the current state-of-the-art of wavelet image coding.

In [4, 5], we utilized a 3-D extension of the SPIHT algorithm and described a very low bitrate embedded 3-D SPIHT video coding algorithm. Besides motion estimation, the 3-D SPIHT algorithm is in principle the same as 2-D SPIHT, except that 3-D wavelet coefficients are treated as a collection of 3-D spatio-temporal orientation trees (see Fig. 2 (b)) and that context modeling in arithmetic coding is more involved. A block-based motion estimation scheme is currently implemented in the 3-D SPIHT coder, and an option for not using motion estimation is also allowed to reduce the encoding complexity. Every 16 frames form a group of frames for 3-D wavelet transformation. Extension to color is accomplished without explicit bit allocation, and can be used for any color space representation. Spatio-temporal orientation trees coupled with powerful SPIHT sorting and refinement turns out to be very efficient. Even without motion compensation, the new video coder provides comparable performance to H.263 objectively and subjectively when operating at bitrates of 30 to 60 Kb/s with minimal system complexity. More importantly, it outperforms MPEG-2 at the same bitrate. In addition to being rate scalable, the 3-D SPIHT video coder allows multiresolutional scalability in encoding and decoding in both time and space. This added functionality along with many desirable features, such as full embeddedness for progressive transmission, precise rate control for constant bitrate traffic, and low complexity for possible software only video applications, makes the new video coder an attractive candidate for multimedia applications like Internet video.

3 Channel Error Effect on 3-D SPIHT Bit-Stream

3-D SPIHT generates an embedded bit-stream so that the decoder can choose any desired video quality and can receive more bit-planes, whenever more channel bandwidth is available, for better reconstructed video quality in the absence of channel error. The 3-D SPIHT bit-stream consists of three parts: header information, significance information, and refinement information. Header information corresponds to the first couple of bytes of the bit-stream, which contains general video information such as video dimension and the maximum magnitude bit with which to start decoding.

Since 3-D SPIHT employs 3-D spatio-temporal trees for the significance test of a large group of pixels to achieve high compression, each significance information bit thus concerns a group of pixels. In addition, significance information bits are further compressed using arithmetic coding. While 3-D SPIHT achieves high coding efficiency by these means, the algorithm is also extremely sensitive to channel error. Even a single channel bit-error occurring in significance information bits can cause the decoder to lose synchronization of set partitioning with the encoder, the result of which is a total collapse of the decoding subsequent process. For example, when a channel error alters the significance information bit of a coefficient, the coefficient moved to the LSP by the encoder at some threshold may still be in the LIP in the decoder. As a result, the decoder is expecting more significance information bits while the encoder only sends refinement bits for the coefficient, which will totally change the decoder's execution path.

Although channel error occurring in a refinement bit only affects a single coefficient, the embedded nature of the 3-D SPIHT bit-stream necessitates that significance information bits and refinement bits are interleaved, making unequal error protection ineffective in this case.

4 3-D SPIHT-RCPC/ARQ Framework

In this section, we cascade 3-D SPIHT with RCPC/ARQ as depicted in Fig. (3) assuming an ideal return channel. A similar concatenated coding scheme was used in [16] to transmit an MPEG video bit-stream over a noisy channel modeled by *additive white Gaussian noise* (AWGN).

In our work, we adopt the similar assumption that the channel in our consideration is binary symmetric with error probability ϵ as high as 10^{-1}. The procedure of the SPIHT-RCPC video coder is illustrated in Fig. (4), where the gray block represents a segment of the channel encoded bit-stream.

We first partition the 3-D SPIHT bit-stream into equal length segments, with each segment being of length N (bits). Each segment is then passed through a cyclic redundancy code (CRC) [17, 18] parity checker to generate $c = 16$ parity bits. In a CRC, binary sequences are associated with polynomials and codewords are selected such that the associated codeword polynomials $v(x)$ of $N+c$ bits segments are the multiples of a certain polynomial $g(x)$ called the *generator polynomial*. For this work, we pick the CRC generator polynomial $g(x) = x^{16}+x^{14}+x^{12}+x^{11}+x^{8}+x^{5}+x^{4}+x^{2}+1$ from the list in [18], which corresponds to the binary sequence of (10101100100110101).

Next, m bits, where m is the memory size of the convolutional coder, are padded at the end of each $N + c$ bits segment to flush the memory of the RCPC coder. Hence, each N bits of the 3-D SPIHT bit-stream is transformed into $N + c + m$ bits before being encoded by the RCPC channel encoder. Finally, $N + c + m$ bits of the segment are passed through the rate r RCPC channel encoder, which is a type of punctured convolutional coder with the added feature of rate compatibility. More details on the RCPC coder will be discussed in the following subsection.

Finally, the channel encoded bit-stream is then transmitted over the computer simulated binary symmetric channel (BSC).

Since the encoder inserts some redundancy bits into the 3-D SPIHT bit-stream according to the rate r of RCPC, the effective source coding rate R_{eff} should be less than the total transmission rate R_{total}, and is given by

$$R_{eff} = \frac{Nr}{N+c+m} R_{total}, \tag{1}$$

where a unit of R_{eff} and R_{total} can be either bits/pixel, bits/sec, or just the length of bit-stream in bits.

At the destination, the channel decoder converts the received bit-stream (packets) into a 3-D SPIHT bit-stream using the list Viterbi decoding algorithm [19, 21] for the convolutional code, in which the "best path" chosen is the one with the lowest path metric that also satisfies the checksum equations. Failure to find such a path is termed a decoding failure. In practice, only a certain number of the best paths are held in contention until a decoding decision is made at a certain depth in the trellis. If none of these paths satisfy the checksum equation a decoding failure is declared.

In the event of a decoding failure, the decoder sends a negative acknowledgment to the transmitter, requesting the same packet. This procedure is limited to only one request, considering the transmission delay and narrow bandwidth of the return channel. This return channel is assumed, however, to be ideal, so that the request is always perfectly received. Therefore, this ARQ scheme only lowers the system throughput marginally (one extra bit per packet).

4.1 Probability of Incomplete Decoding

We assume that probability of a packet decoding failure with the Viterbi decoder is given as p with $p \ll 1$. The decoding error of the Viterbi algorithm occurs independently for each packet received. We express the probability of decoding failure (incomplete decoding) occurring in the first M packets, $P(M)$, of the bit-stream in terms of p. First, we focus on the simple RCPC strategy without ARQ. Hence, we have

$$\begin{aligned} P(M) &= 1 - Prob\{\text{No Failure in M packets}\} \\ &= 1 - (1-p)^M \\ &\approx Mp, \end{aligned} \tag{2}$$

where $(1-p)$ is the probability of successful Viterbi decoding for each packet. ¿From this, it can be observed that the longer the bit-stream, the more likely it is to be decoded incorrectly.

When a return channel is available, a better idea is to use an ARQ strategy, combined with Viterbi decoding to reduce incomplete decoding probability. The idea is that it is very unlikely that the retransmitted packet will be decoded incorrectly again. By requesting the packet only one more time, we have found that the probability of incomplete decoding of the entire video bit-stream can be very small. However, we assume that the retransmission can take place only once. If the retransmitted packet is decoded incorrectly again, we just discard the following packets. The probability of decoding failure occurring in one of the first M packets in this case is given by

$$\begin{aligned} P(M) &= 1 - \text{Prob}\{\text{No Failure in M packets}\} \\ &= 1 - [(1-p) + p(1-p)]^M \\ &= 1 - (1-p^2)^M \\ &\approx Mp^2. \end{aligned} \tag{3}$$

The first term in the bracket, $(1-p)$, is the probability of successful Viterbi decoding for each packet when no request is needed, and the second term, $p(1-p)$, is the probability of successful Viterbi decoding when one request is successful after one decoding failure. We assume for this analysis and the simulations that follow the same probability p of packet decoding failure in the forward and repeat channel. Hence, Equation (3) implies that by employing an ARQ scheme, we can reduce the probability of incomplete decoding by a factor of p.

4.2 The choice of RCPC rate and system complexity

The RCPC rate r is defined as $k/n < 1$, where k is the number of input bits entering the RCPC coder and n is the number of corresponding output bits. Hence, the rate can be interpreted as the number of information bits entering the encoder per transmitted symbol. Embeddedness in 3-D SPIHT and rate compatibility of the RCPC coder mean that high rate codes are embedded in lower rate codes. Consequently, none of the already transmitted code bits are wasted, but are used to improve FEC coding. If a return channel is available, this scheme can be used for a hybrid FEC/ARQ scheme, not by repeating the whole packet if the transmission is unsuccessful, but only by transmitting additional code bits of a low rate RCPC code until the code is powerful enough to successfully decode.

The effect of incomplete decoding is often acceptable since the 3-D SPIHT bit-stream is embedded. Hence, the result of incomplete RCPC decoding will be the video decoded at a lower source coding rate (the decoder just discards whatever remains when it finds incomplete decoding). If a higher RCPC rate is chosen for a given BER, one can achieve higher typical PSNR of the decoded video, while we have higher probability of a packet decoding failure p. On the contrary, a lower RCPC rate gives less typical PSNR, keeping the incomplete decoding probability small. Therefore, there is a trade-off between source coding rate and channel coding rate for a given total transmission rate. Theoretical results on optimal tradeoff between fixed-delay source coding and block channel coding are derived in [20] for the BSC, and they are the bases for choosing the RCPC rates in a practical image transmission system [1, 2]. We follow the same choices in this work as in [1, 2] when the channel conditions are the same.

The complexity of the channel decoder is very low as we limit the list decoding search depth to 10 paths in the trellis. A detailed discussion can be found in [2].

5 Simulation Results

In our simulation, we used the same set of parameters for the CRC parity checker and RCPC channel coder as in [1]. That is: $N = 200$, $c = 16$, and $m = 6$. The test BSC channel has three

BER	0.1	0.01	0.001
RCPC rate (r)	2/7	2/3	8/9
R_{eff} (bps)	651223	1519519	2026026
Packet decoding failure p	3.0×10^{-5}	1.3×10^{-5}	1.0×10^{-5}
Number of packets M per GOF	1740	4060	5412
$P(M)$ without ARQ	0.05	0.05	0.05
$P(M)$ with ARQ	1.5×10^{-6}	6.6×10^{-7}	5.0×10^{-7}

Table 1: 3-D SPIHT-RCPC coding parameters, packet decoding probability, and incomplete decoding probability at total transmission rate of 2.53 Mbps

different BER's of $\epsilon = \{0.1, 0.01, 0.001\}$, and their corresponding RCPC rates and R_{eff} are given in Table (1), where R_{eff} is calculated from Equation (1) by keeping the total transmission bit-rate R_{total} constant to 2.53 Mbps.

The RCPC rate r in [2] was chosen such that probability of incomplete decoding of RCPC for a bit-stream of 262144 bits (equivalently total transmission rate of 1.0 bpp with 512×512 image) is less than 0.01 for three different BER's. The number of packets depends on the rate r. For example, if $r = 2/3$ and the size of bit-stream is $R_{total} = 262144$ bits, we have $M = R_{eff}/N = \frac{Nr}{(N+c+m)N} R_{total} = (2/3)/222 \times 262144 \approx 788$ packets in this case. In other words, for $r = 2/3$, $P(788) = 0.01$. Consequently, probability of a packet decoding failure of the Viterbi decoder depends on rate r. For example, we can obtain the probability of a packet decoding failure, $p = 1 - (1 - P(788))^{1/788} = 1.3 \times 10^{-5}$, for $r = 2/3$. The probability of a packet decoding failure for given RCPC code rates and incomplete decoding of a 3-D SPIHT-RCPC bit-stream of a group of frames (GOF) at total transmission rate of 2.53 Mbps are also given in Table (1), where we can see that a hybrid FEC/ARQ method gives much less probability of incomplete decoding.

We tested "Football" and "Table Tennis" sequences at 30 frames per second of SIF (352×240) format. Coding results of the two sequences for the fixed transmission rate R_{total}, depicted as graphs of PSNR versus frame number, using RCPC with ARQ at various BER's, are shown in Figure (5) and the same without ARQ are shown in Figure (6). Table (2) shows the average PSNRs of the two sequences in these simulations. As expected with higher BER's, the penalty for not using ARQ becomes quite severe. In fact, at BER = 0.1, the results were too poor to be reported, as incomplete decoding always occurred early in the bit stream, leaving too few uncorrupted bits for meaningful decoding and reconstruction.

In none of these experiments with ARQ is there severe degradation of reconstructed video due to the channel errors. In fact, for a given image sequence, total bit-rate and channel error probability, the PSNR performance of 3-D SPIHT-RCPC remains constant with probability nearly one over all possible channel error patterns. The cost paid in this case is the reduced average (or frame-by-frame)

BER	0.1		0.01		0.001		0.0001		0.0
	ARQ	No ARQ	ARQ	No ARQ	ARQ	No ARQ	ARQ	No ARQ	ARQ/No ARQ
Football	27.4	-	32.1	24.5	32.8	28.2	-	32.8	34.2
Tennis	30.3	-	34.2	28.7	35.8	31.4	-	34.3	37.2

Table 2: Average PSNR in dB of 3-D SPIHT-RCPC coding at different BER's, assuming the total transmission rate $R_{total} = 2.53$ Mbps

PSNR due to the lowered source coding rate. Hence, the distortion is now controllable. In other words, since 3-D SPIHT is embedded, the decoder can request for more information (additional 3-D SPIHT-RCPC bit-stream) to improve the video quality from the transmitter whenever more channel bandwidth is available, while without channel coding the distortion in the decoded video due to the channel noise would have been totally uncontrollable. Typical reconstructions of "Football" sequence at channel bit error rates of 0.1 and 0.001 are shown in Fig. (7).

6 Summary and Conclusions

We considered a hybrid FEC/ARQ error control scheme to protect 3-D SPIHT bit-stream from being corrupted by the channel noise, since the 3-D SPIHT bit-stream is very sensitive to channel noise. 3-D SPIHT source coding algorithm was cascaded with the RCPC channel encoder and the channel encoded bit-stream was passed through the BSC channel. At the destination, the channel decoder (Viterbi decoder and ARQ) detected and corrected every channel bit-error pattern resulting in no uncontrollable degradation of the decoded video. Hence, transmission of 3-D SPIHT video over a noisy channel amounts to the transmission of the bit-stream over a noiseless channel with reduced bit-rate. Due to the embeddedness of 3-D SPIHT bit-stream, incomplete channel decoding often gives the possibility of reasonable quality of decoded video, which can be further improved only by transmitting additional information.

References

[1] P. G. Sherwood and K. Zeger, "Progressive Image Coding for Noisy Channels", *IEEE Signal Processing Letters*, vol. 4, pp. 189-191, July 1997.

[2] P. G. Sherwood and K. Zeger, "Progressive Image Coding on Noisy Channels", *Proc. DCC'97*, pp. 72–81, April 1997.

[3] 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 Tech.*, Vol. 6, pp. 243-250, June, 1996.

[4] W. A. Pearlman, B.-J. Kim, and Z. Xiong, "Embedded video subband coding with 3-D SPIHT," in *Wavelet Image and Video Compression*, P. Topiwala, eds, Kluwer, 1998.

[5] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Very low bit-rate embedded video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," submitted to *IEEE Trans. Circuits and Systems for Video Tech.*, October 1997. Postscript version of the manuscript available at http://cipr.rpi.edu/publications/publications.html.

[6] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications", *IEEE Trans. on Comm.*, vol. 36, pp. 389-400, April 1988.

[7] C. E. Shannon, "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 (Part I) and 623–656 (Part II), July 1948.

[8] Shu Lin and Daniel J. Costello, Jr., "Error Control Coding", Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1983.

[9] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 926-943, June 1992.

[10] Q. Chen and T. R. Fischer, "Image coding using robust quantization for noisy digital transmission," *IEEE Trans. on Image Processing*, vol. 7, pp. 496-505, April 1998.

[11] V. Chande, H. Jafarkhani, and N. Farvardin, "Joint source-channel coding of images for channels with feedback," *Proc. IEEE Information Theory Workshop*, San Diego, CA, February 1998.

[12] G. Cheung and A. Zakhor, "Joint source/channel coding of scalable video over noisy channels," *Proc. ICIP'96*, Lausanne, Switzerland, September 1996.

[13] Z. Xiong, B.-J. Kim, and W. A. Pearlman, "Progressive Video Coding for Noisy Channels", *Proc. ICIP'98*, Chicago, IL, October 1998.

[14] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, 12, pp. 3445-3463, December 1993.

[15] http://www.icsl.ucla.edu/~ipl/psnr_results.html.

[16] J. W. Modestino, W. A. Pearlman, and Ligang Lu, "Reliable Transmission of MPEG Video over Direct Broadcast Satellite", *GE Corporate Research and Development*, August 1994.

[17] T. V. Ramabadran and S. S. Gaitonde, "A Tutorial on CRC Computations", *IEEE Micro*, vol. 8, pp. 62-75, August 1988.

[18] G. Castagnoli, J. Ganz, and P. Graber, "Optimum Cyclic Redundancy-Check Codes with 16-Bit", *IEEE Transactions on Communications*, vol. 38, pp. 111–114, January 1990.

[19] G. D. Forney, Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 169-176, February 1984.

[20] B. Hochwald and K. Zeger, "Tradeoff between source and channel coding," *IEEE Trans. on Inform. Theory*, vol. 43, pp. 1412-1424, September 1997.

[21] N. Seshadri and C. Sundberg, "List Viterbi decoding algorithm with applications," *IEEE Transactions on Communications*, vol. 42, pp. 313-323, 1994.

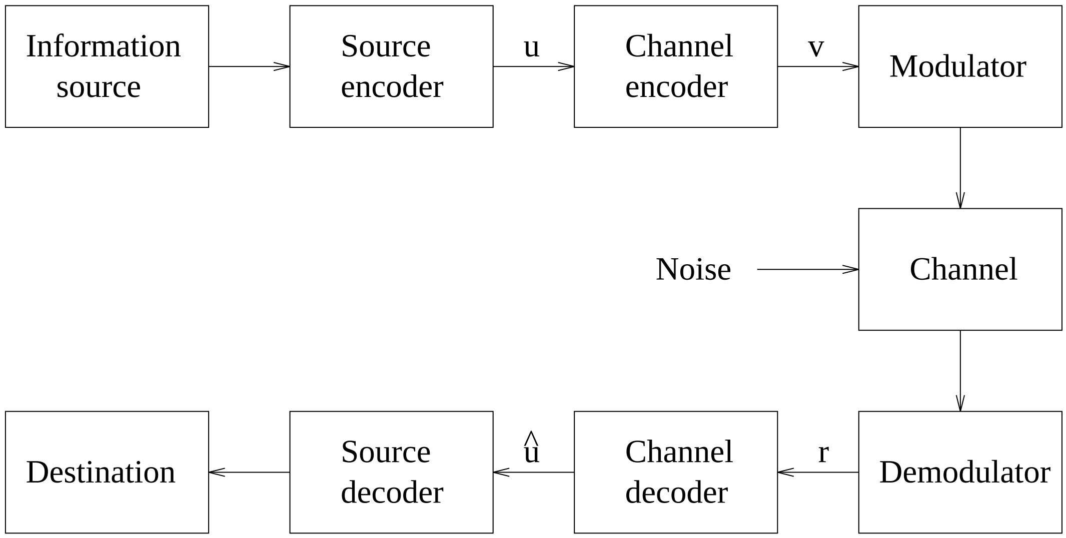
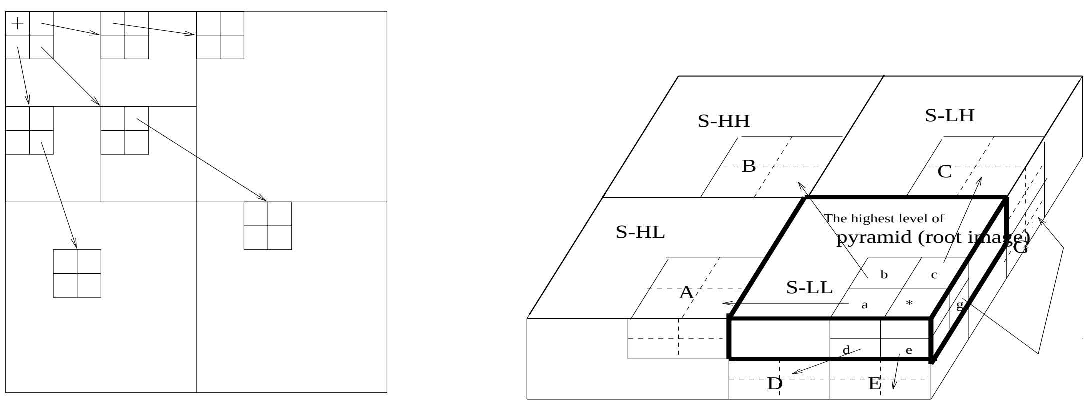


Figure 1: Block diagram of a typical transmission system.



(a) (b)

Figure 2: 2-D spatial orientation tree and 3-D spatio-temporal orientation tree.

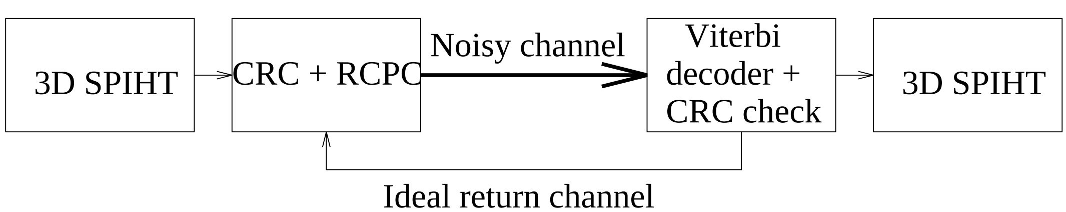


Figure 3: 3-D SPIHT-RCPC/ARQ system framework assuming an ideal return channel.

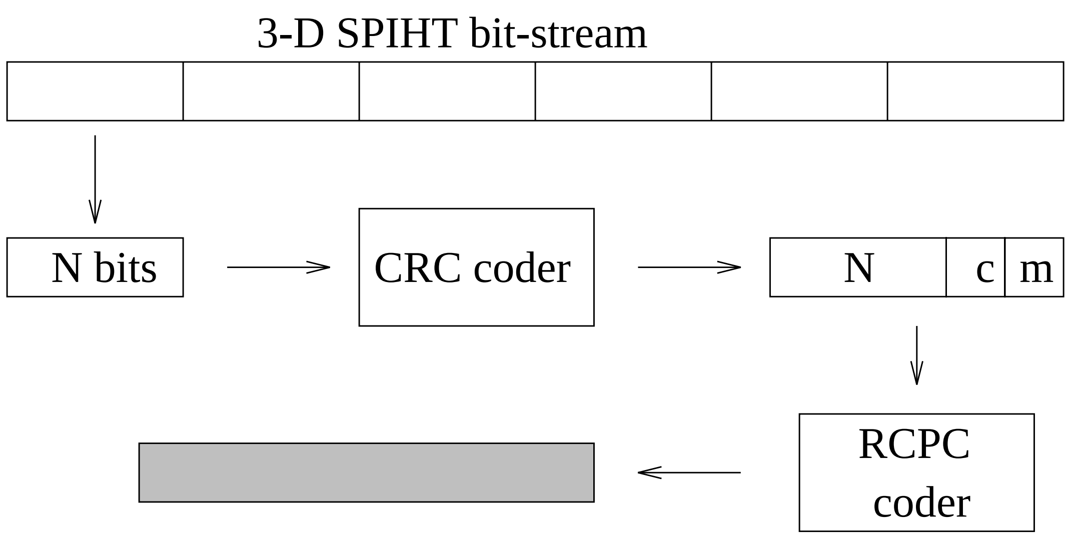


Figure 4: System description of 3-D SPIHT-RCPC coder.

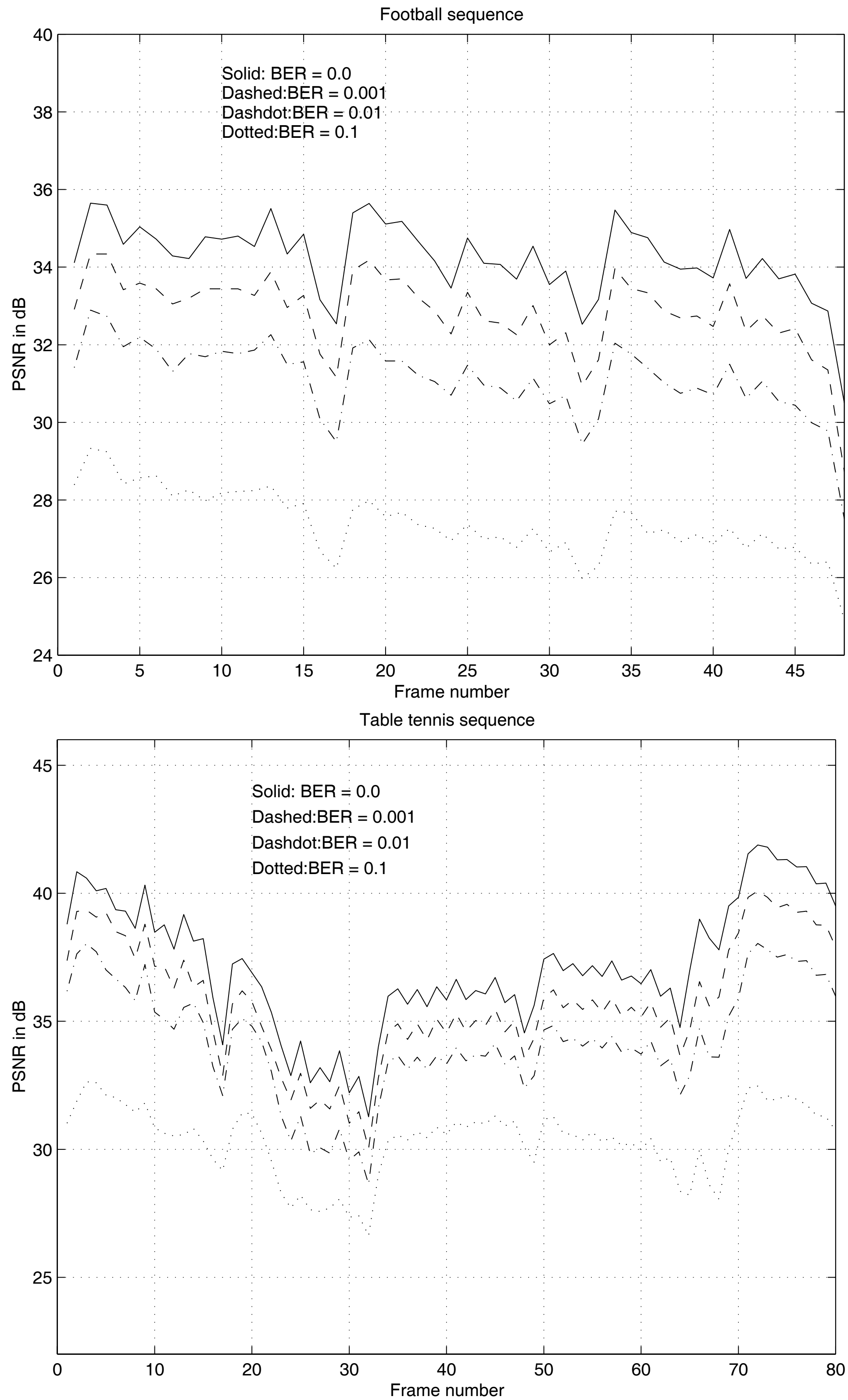


Figure 5: 3-D SPIHT-RCPC with ARQ frame-by-frame PSNR of "Football" and "Table tennis" sequences at different BER's. Total transmission rate is fixed to 2.53 Mbps for both sequences.

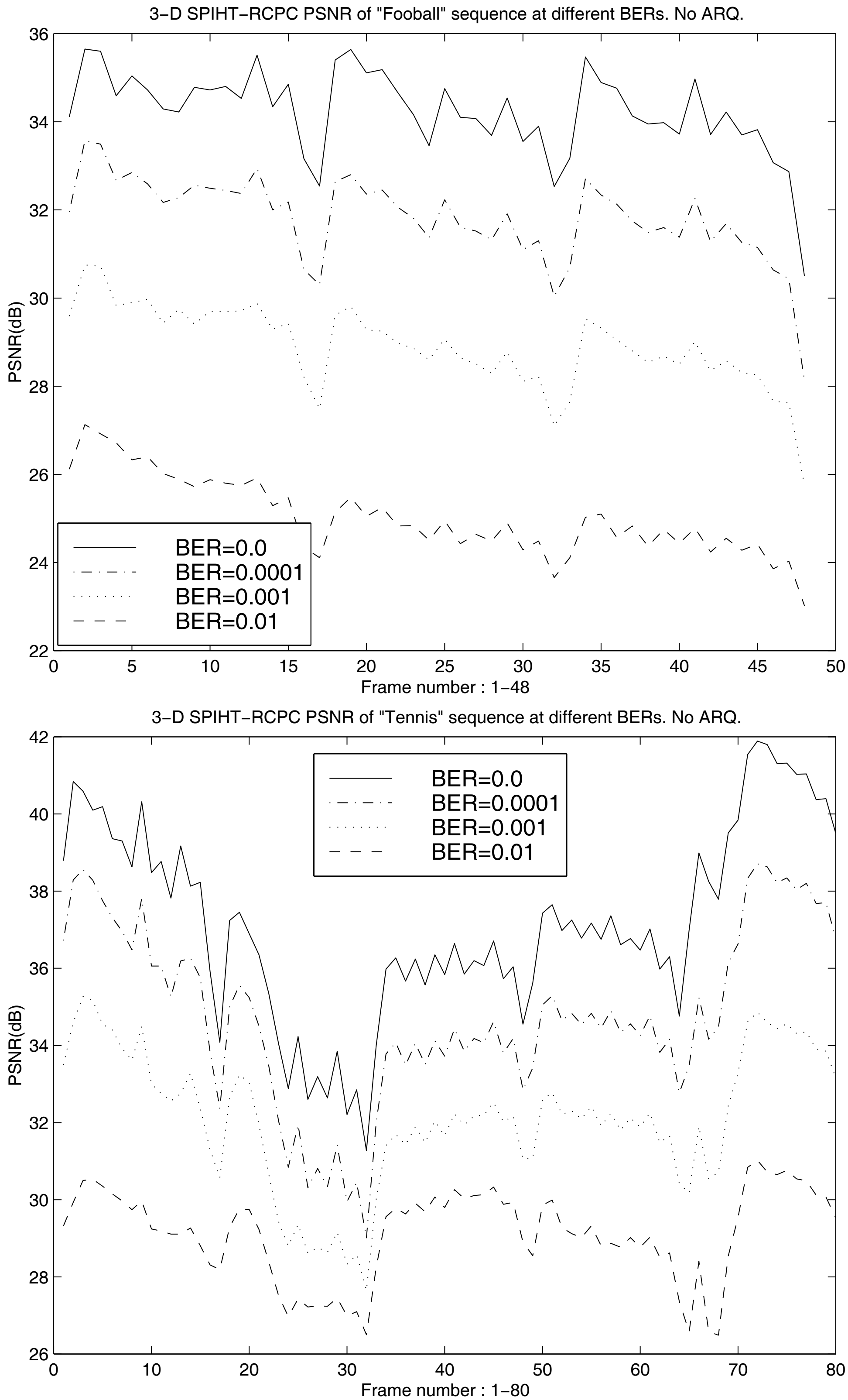


Figure 6: 3-D SPIHT-RCPC without ARQ frame-by-frame PSNR of “Football” and “Table Tennis” sequences at different BER’s. Total transmission rate is fixed to 2.53 Mbps for both sequences.

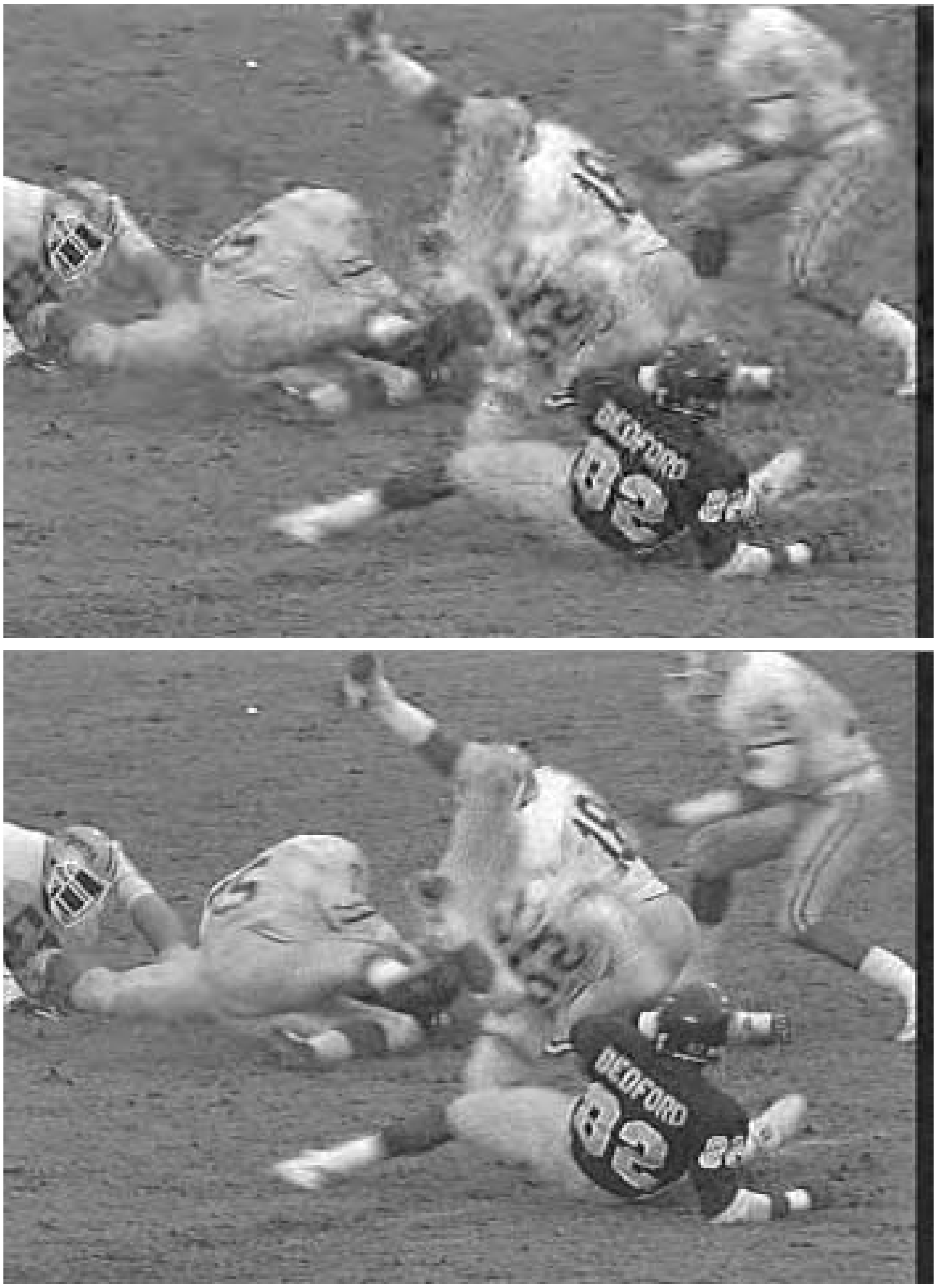


Figure 7: The same "Football" reconstructions (frame 16) at BER = 0.1 (top: PSNR = 26.68 dB) and BER = 0.001 (bottom:PSNR = 31.76 dB) with 3-D SPIHT-RCPC/ARQ. Total transmission rate is set to 2.53 Mbps.

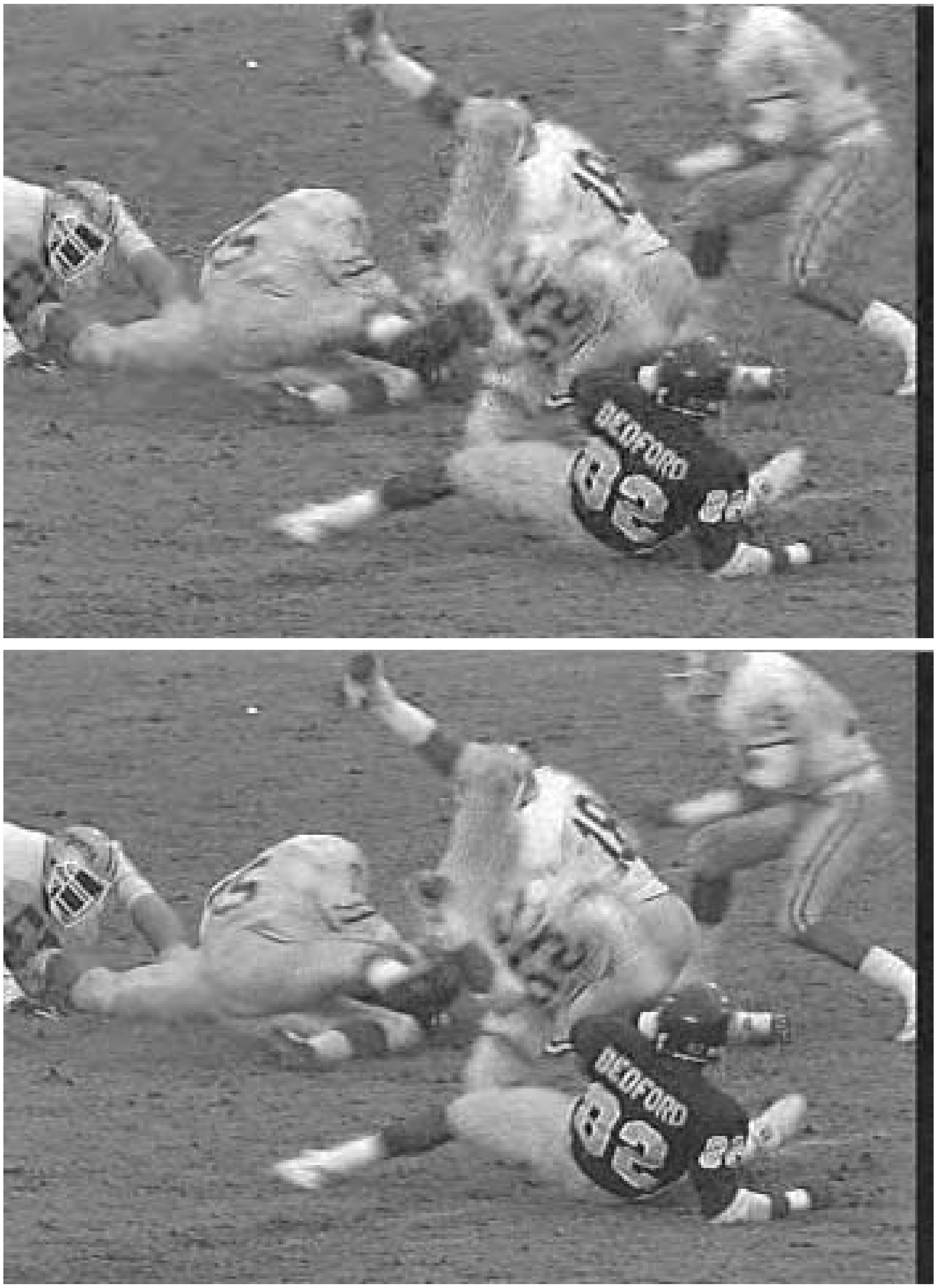


Figure 8: The same "Football" reconstructions (frame 16) without ARQ (top:PSNR = 28.21 dB) and with ARQ (bottom:PSNR = 31.76 dB) at BER = 0.001. Total transmission rate is set to 2.53 Mbps.