

Research Article

Unequal Error Protection Techniques Based on Wyner-Ziv Coding

Liang Liang,¹ Paul Salama,² and Edward J. Delp (EURASIP Member)¹

¹ Video and Image Processing Laboratory (VIPER), School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

² Department of Electrical and Computer Engineering, Indiana University—Purdue University at Indianapolis, Indianapolis, IN 46202, USA

Correspondence should be addressed to Edward J. Delp, ace@ecn.purdue.edu

Received 31 May 2008; Revised 2 November 2008; Accepted 17 March 2009

Recommended by Frederic Dufaux

Compressed video is very sensitive to channel errors. A few bit losses can stop the entire decoding process. Therefore, protecting compressed video is always necessary for reliable visual communications. Utilizing unequal error protection schemes that assign different protection levels to the different elements in a compressed video stream is an efficient and effective way to combat channel errors. Three such schemes, based on Wyner-Ziv coding, are described herein. These schemes independently provide different protection levels to motion information and the transform coefficients produced by an H.264/AVC encoder. One method adapts the protection levels to the content of each frame, while another utilizes feedback regarding the latest channel packet loss rate to adjust the protection levels. All three methods demonstrate superior error resilience to using equal error protection in the face of packet losses.

Copyright © 2009 Liang Liang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Channel errors can result in serious loss of decoded video quality. Many error resilience and concealment schemes have been proposed [1]. However, when large errors occur, most of the proposed techniques are not sufficient enough to recover the loss. In recent years, error resilience approaches employing Wyner-Ziv lossy coding theory [2] have been developed and have resulted in improvement in the visual quality of the decoded frames [3–13]. Other works applied distributed source coding onto error resilience include [14–17].

In 1976, Wyner and Ziv proved that when the side information is only known to the decoder, the minimum required source coding rate will be greater or equal to the rate when the side information is available at both encoder and decoder (see Figure 1). Denoting the source data by X and the side information by Y , where X and Y are correlated, but the side information Y is only available at the decoder, the decoder manages to reconstruct a version of X , X' , subject to the constraint that at most a distortion D is incurred. It was shown that $R^{\text{WZ}}(D) \geq R_{X|Y}(D)$ [2], where $R^{\text{WZ}}(D)$ is the data rate used when the side information is only available to the

decoder and $R_{X|Y}(D)$ represents the data rate required when the side information is available at both the encoder and the decoder.

Wyner and Ziv also proved that equality can be achieved when X is Gaussian memoryless source and D is mean square error distortion $D(X, X') = (X - X')^2$, as well as when the source data is the sum of an arbitrarily distributed side information Y and independent Gaussian noise U . In addition, they derived the rate boundary $R^{\text{WZ}} = R_{X|Y}(D) = (1/2) \log(\sigma_U^2 \sigma_X^2 / (\sigma_U^2 + \sigma_X^2) d)$ that can be achieved when $0 < D < \sigma_U^2 \sigma_X^2 / (\sigma_U^2 + \sigma_X^2)$, and where σ_U^2 and σ_X^2 are the variances of the Gaussian noise U and the source data X [2].

One of the earliest work of applying Wyner-Ziv lossy coding theory for error resilient video transmission is proposed in [3], 2003. The general approach is to use an independent Wyner-Ziv codec (as shown in Figure 3) to protect a coarse-version of the input video sequence, which can be decoded together with the side information from the primary MPEG-x/H.26x decoder. The basic system structure is shown in Figure 2. The approach proposed in [3] is known as systematic lossy forward error protection (SLEP).

SLEP, in addition to an MPEG-2 encoder, uses a Wyner-Ziv encoder made up of a coarse quantizer and a lossless

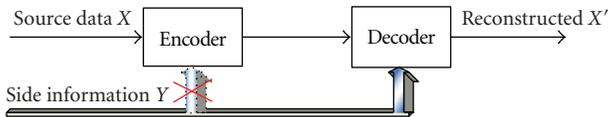


FIGURE 1: Side information available at decoder only.

Slepian-Wolf encoder that utilizes Turbo coding. The input to the Wyner-Ziv encoder consists of the reconstructed frames obtained from the MPEG-2 encoder. These are initially coarsely quantized and then passed onto a Turbo encoder [18, 19], which outputs selected parity bits. At the receiving end, a Turbo decoder uses the output of the MPEG-2 decoder, as side information, and the received parity bits to recover the lost video data. In the absence of any channel errors, the output of the SLEP decoder will be the same as that of the MPEG-2 decoder. If however, channel errors corrupt the MPEG-2 stream, then SLEP attempts to reconstruct a coarse version of the MPEG-2 stream via the received parity bits, which may have also been corrupted. The quality of the reconstructed version depends on the quantization step used by the coarse quantizer as well as the strength of the Turbo code.

Improvements to SLEP have been proposed in [9, 12], and have resulted in a lower data rate for Wyner-Ziv coding as well as improved decoded video quality. It is noted that the SLEP method has been applied to H.264 in [12].

Another approach of using Wyner-Ziv coding for robust video transmission was proposed in [20], in which the Wyner-Ziv encoder consisted of a discrete cosine transform, a scalar quantizer and an irregular repeat accumulate code as the Slepian-Wolf coder.

Our approach to unequal error protection is also based on Wyner-Ziv coding and is motivated by the SLEP approach. The overall goal of our schemes is to correct errors in each frame by protecting motion information and the transform coefficients. The primary codec is an H.264/AVC codec and the Wyner-Ziv codec utilizes coarse quantization and a Turbo codec. Instead of protecting everything associated with the coarsely reconstructed frames, we separately protect motion information, and transform coefficients produced by the primary H.264 encoder. The idea being that since the loss of motion information impacts the quality of decoded video differently from the loss of transform coefficients, both should receive unequal levels of protection that are commensurate with their respective contributions to the quality of the video reconstructed by the decoder [21]. The motion information is protected via Turbo coding whereas the transform coefficients are protected via Wyner-Ziv coding. This approach is referred to as unequal error protection using Pseudo Wyner-Ziv (UEPWZ) coding.

We improve the performance of our unequal error protection technique by adapting the parity data rates for protecting the video information to the content of each frame. This is referred to as content adaptive unequal error protection (CAUEP) [22]. In this scheme, a content adaptive function was used to evaluate the normalized sum of the absolute difference (SAD) between the reconstructed

frames and the predicted frames. Depending on pre-selected thresholds, the parity data rates assigned to the motion information and the transform coefficients were varied for each frame. This resulted in a more effective and flexible error resilience technique that had an improved performance compared to the original UEPWZ.

Another approach to improve the proposed unequal error protection is to send feedback regarding the current channel packet loss rates to the Pseudo Wyner-Ziv encoder, in order to correspondingly adjust the amount of parity bits needed for correcting the corrupted slices at the decoder [23]. This approach is referred to as feedback aided unequal error protection (FBUEP). At the decoder, the current packet loss rate is estimated based on the received data and sent back to the Pseudo Wyner-Ziv encoder via the real-time transport control protocol (RTCP) feedback mechanism. This information is utilized by the Turbo encoders to update the parity data rates of the motion information and the transform coefficients, which are still protected independently. At the Wyner-Ziv decoder, the received parity bits together with the side information from the primary decoder are used to decode and restore corrupted slices. These in turn are sent back to the primary decoder to replace their corrupted counterparts. It is to be noted that simply increasing the parity bits when the packet loss rate increases is not applicable, since it will exacerbate network congestion [24]. Instead, the total transmission data rate should be kept constant, which means that when the packet loss rate increases, the primary data transmission rate should be lowered in order to spare more bits for parity bits transmission.

Our proposed error resilience schemes aim to improve both the rate distortion performance as well as the visual quality of the decoded video frames when video has been streamed over data networks such as wireless networks that experience high packet losses. In our experiments, we only consider packet erasures whether due to network congestion or uncorrected bit errors. The main focus of our scheme is for applications such as video conferencing, especially in a wireless network scenario, where serious packet losses will result in unpleasant distortion during real time video streaming.

In this paper, UEPWZ is described in Section 2, and the details of CAUEP and FBUEP as well as the improvement in performance achieved are presented in Section 3. The experimental results of the three techniques are compared and analyzed in Section 4, showing the significant improvement the CAUEP and the FBUEP achieved in rate distortion performance and the visual quality of the decoded frames. Finally, the conclusion is provided in Section 5.

2. Unequal Error Protection Based on Wyner-Ziv Coding

As mentioned previously, the approach to unequal error protection undertaken here is based on Wyner-Ziv coding and is motivated by the SLEP approach. The primary codec is an H.264/AVC codec and the Wyner-Ziv codec utilizes coarse quantization and two pairs of Turbo codecs. Instead

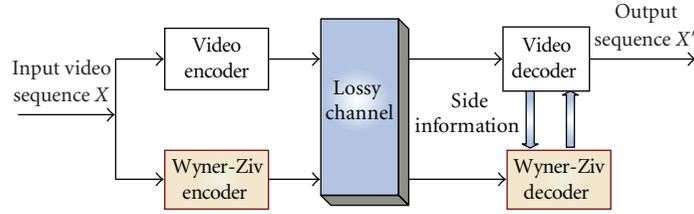


FIGURE 2: Error resilient video streaming using Wyner-Ziv coding.

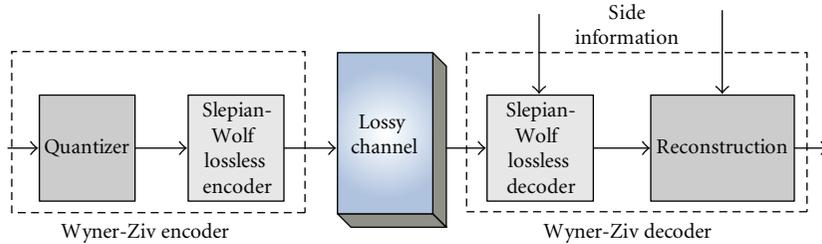


FIGURE 3: Wyner-Ziv codec.

of protecting everything associated with the coarsely reconstructed frames, we separately protect motion information and transform coefficients produced by the primary H.264 encoder. The idea being that since the loss of motion information impacts the quality of decoded video differently from the loss of transform coefficients, both should receive unequal levels of protection that are commensurate with their respective contributions to the quality of the video reconstructed by the decoder [21]. The block diagram depicting the unequal error protection system is shown in Figure 4.

In H.264/AVC, there are 9 modes used for predicting a 4×4 block in an I frame and 4 modes for predicting a 16×16 block from its neighbors [25, 26]. The mode index and the transform coefficients are critical for proper frame reconstruction at the decoder. In the case of P and B frames, the H.264/AVC standard allows the encoder the flexibility to choose among different reference frames and block sizes for motion prediction. In particular, the standard permits block sizes of 4×4 , 4×8 , 8×4 , 8×8 , 8×16 , 16×8 , and 16×16 . Since motion vectors belonging to neighboring blocks are highly correlated, motion vector differences (MVD) are encoded and transmitted to the decoder side, together with the reference frame index, mode information and the residual transform coefficients.

In the unequal error protection scheme, the important video information are protected through the Pseudo Wyner-Ziv coder. In the case of I frames, mode information (MI) as well as the transform coefficients are protected whereas motion vector differences, mode information and reference frame index (RI) are protected for P and B frames. These are scanned and used to create long symbol blocks that are sent to the Turbo encoder.

In order to mitigate the mismatch between the transform coefficients input to the Wyner-Ziv encoder and the corresponding side information at the Wyner-Ziv decoder, an inverse quantizer, identical to the one used in the H.264/AVC

decoder, is initially used to de-quantize the coefficients. These are then coarsely quantized by a uniform scalar quantizer with 2^N levels ($N \leq 8$), and used to form a block of symbols that is passed onto the Turbo encoder. The quantization step size for processing the transform coefficients is therefore $2^{(8-N)}$. In all cases, the output of the Turbo encoder is punctured to reduce the overall data rate.

Due to the importance of maintaining its accuracy the motion information is not quantized. Instead, the Turbo encoder takes in the motion information directly and outputs the selected parity bits. It can be noticed that without using quantization, the processing of Turbo coding motion information itself is not strictly speaking Wyner-Ziv coding. Therefore, we name the whole secondary encoder as Pseudo Wyner-Ziv encoder instead of Wyner-Ziv encoder, and we refer to this scheme as unequal error protection using Pseudo Wyner-Ziv coding (UEPWZ). However, the application of Turbo coding in our schemes is different from straight forward error control coding. In our application, only the parity bits p produced by the Turbo encoder are transmitted to the decoder. The output data stream u from the first branch is not transmitted to the decoder side. This is illustrated in Figure 5. The corresponding decoded error prone primary video data from the H.264 decoder will be used as to decode the parity bits received by the Turbo decoders.

Because of the independent processing of the motion data and the transform coefficients in the Pseudo Wyner-Ziv encoder, the parity data rates in the corresponding Turbo encoder can be assigned separately.

The Turbo encoder we used consists of two identical recursive systematic encoders (see Figure 5) [27], each having the generator function: $H(D) = (1 + D^2 + D^3 + D^4)/(1 + D + D^4)$. The input symbols sent to the second recursive encoder are interleaved first in a permuter before being passed to it. The puncture mechanism is used to delete some of the parity bits output from the two recursive encoders, in order to meet

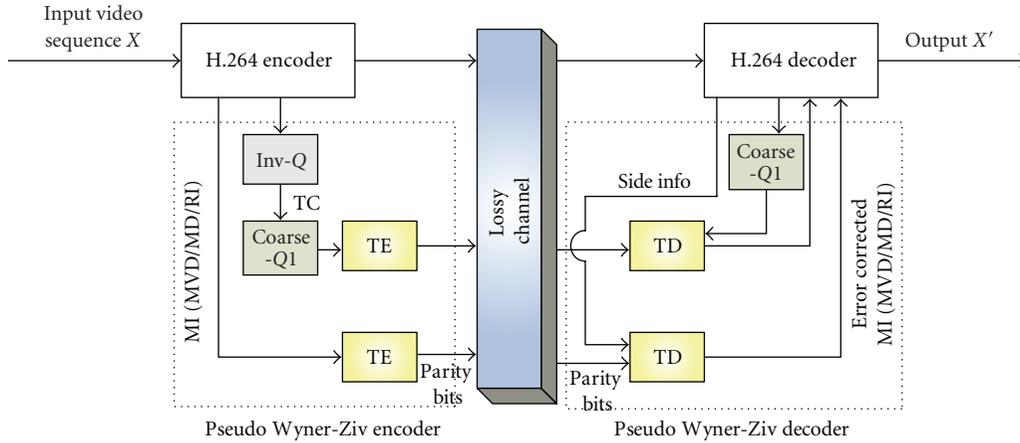


FIGURE 4: Unequal error protection based on Wyner-Ziv coding.

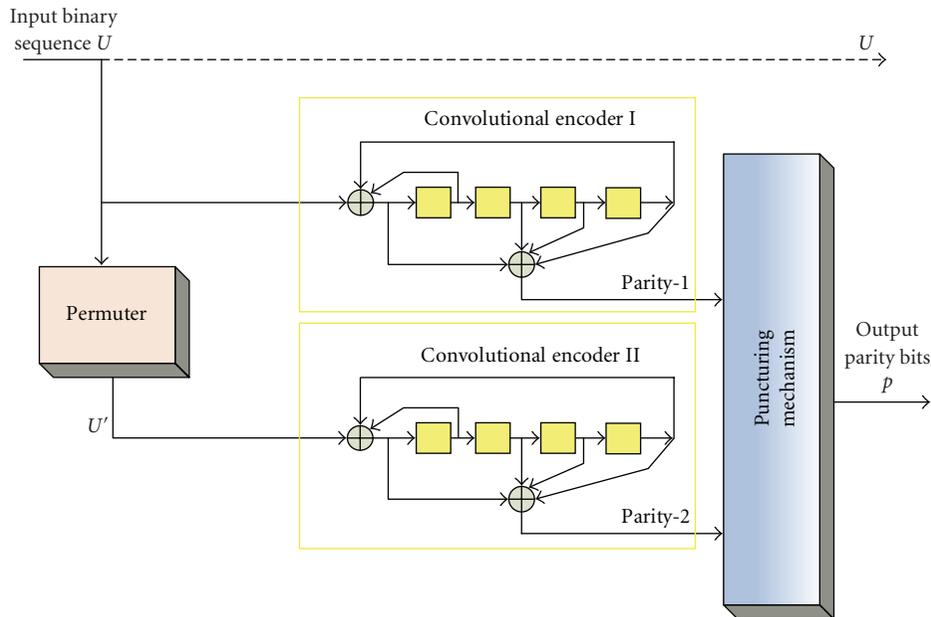


FIGURE 5: Parallel turbo encoder.

a target parity data rate. Only parity bits are transmitted to the decoder side. The first branch of data, symbolized by the dashed line in Figure 5, is not transmitted. The error correction capability of the Turbo coder also depends on the length of the symbol blocks. In our scheme, the symbol block length is in the unit of a frame instead of a slice. For the transform coefficients, the symbol block length is 25344 for a QCIF sequence. In the proposed scheme the motion vectors are obtained for each 4×4 blocks, which makes the symbol block length of 3168. The experiment results also show that the Turbo encoder still maintains strong error correction ability for such a symbol block length.

The Turbo decoder utilizes the received parity bits and the side information from the H.264/AVC decoder, to perform the iterative decoding using two BCJR-MAP decoders [27]. The error corrected information is then sent back to the

H.264/AVC decoder to replace the error corrupted data. In this process, the decoded error-prone transform coefficients are first sent to a coarse quantizer, which is the same as the one used at the Pseudo Wyner-Ziv encoder side. The reason is that at the encoder side, in order to save data rate usage by the Wyner-Ziv coding, a coarse version of the transform coefficients is Turbo encoded. However, Only the output parity bits are transmitted to the decoder side. The video data u output from the Turbo encoder is not transmitted. Instead, the H.264 decoded transform coefficients are used as it, together with the received parity bits of the Turbo encoded coarse-version transform coefficients, to decode the error corrected coarse version of the transform coefficients. When using the real-time transport protocol (RTP), packet loss can be inferred at the decoder easily by checking the sequence number field in the RTP headers. Wyner-Ziv decoding only

performs when the decoder detects packet losses. When no packet loss happens, the H.264 decoded transform coefficients are used for decoding the residual frames. However, when packet loss happens, the coarser version of the transform coefficients decoded by the Turbo decoder is used to limit the maximum degradation that can occur. In the parallel process, the error corrupted motion information received by the H.264/AVC decoder was sent directly to the corresponding Turbo decoder, together with the received corresponding parity bits, to decode the error corrected motion information. It is then sent back to the H.264/AVC decoder to replace the error-corrupted motion information. The reconstructed frames can be further used as the reference frames in the following decoding process. Therefore, the final version of the decoded video sequence are obtained based on the error corrected motion information and the transform coefficients, which resulted in good quality decoded frames as shown in Section 4. However, in the case of serious channel loss and/or limited available data rate for error protection, the Pseudo Wyner-Ziv coder might not have enough strength to recover all the lost video information. Also there is no fall back mechanism in use to ensure the correct turbo decoding. On this point, the UEPWZ takes the advantage of allocating different protection level on different protected video data elements depending on their overall impact on the decoded video sequence. The experiments showed that by assigning unequal data rate for protecting motion information and the transform coefficients, the rate distortion performance can be improved compared to the equal parity data rate allocation case.

3. Improved Unequal Error Protection Techniques

In this section, the two approaches developed to improve UEPWZ technique are introduced in detail. Content adaptive unequal error protection (CAUEP) improves UEPWZ from the encoder side by analyzing the content of each frame while feedback aided unequal error protection (FBUEP) utilizes channel loss information conveyed from the H.264 decoder side. Both approaches improved the original UEPWZ in a different aspect, which results in further efficiency on data rate allocation and the significant improvement on the visual quality of the decoded frames.

3.1. Content-Adaptive Unequal Error Protection. In UEPWZ, the parity data rates for Turbo coding the motion information and the transform coefficients are always set in advance and fixed throughout. However, in a video sequence, different video content in each part of the sequence may require different amounts of protection for the corresponding video data elements. The amount of the motion contained in each frame may change over time, which means part of the video sequence may contain a large amount of motion while some other parts may only contain slow motion content. For this type of video sequences, fixed parity data rate assignment may result in inefficient error protection. When motion content increases in the video sequence, the pre-assigned parity data rate may become insufficient to correct the errors

TABLE 1: Setting of parity data rate (PDR).

SAD range	PDR assignment
$SAD_n \leq T_1$	$PDR_{MI} = \frac{1}{4}, PDR_{TC} = 0$
$T_1 < SAD_n \leq T_2$	$PDR_{MI} = \frac{1}{2}, PDR_{TC} = 0$
$T_2 < SAD_n \leq T_3$	$PDR_{MI} = \frac{1}{2}, PDR_{TC} = \frac{1}{8}$
$SAD_n > T_3$	$PDR_{MI} = \frac{1}{2}, PDR_{TC} = \frac{1}{4}$

while it may result in sending redundant parity bits when the motion content decreases in the same video sequence.

The goal of developing an efficient error resilience technique is to make the algorithm applicable to all types of video sequences. Therefore, a function needs to be embedded in the Wyner-Ziv coder to analyze the video content, such as the amount of the motion, in each frame. CAUEP improves UEPWZ by adapting the protection levels of different video data element, to the content of each frame.

In order to achieve this goal a content adaptive function (CAF) that utilizes the normalized sum of absolute difference (SAD) between each reconstructed frame and its predicted counterpart is used. This is given by $SAD_n = \sum_{i=1, j=1}^{i=N, j=M} |X_{i,j} - X_{p(i,j)}| / N \times M$, where $X_{i,j}$ denotes the reconstructed pixel value at position (i, j) , $X_{p(i,j)}$ is the value of the predicted pixel at position (i, j) , and SAD_n represents the normalized total value of SAD of the n th frame in the sequence.

The SAD of each frame is compared to three pre-defined thresholds T_1 , T_2 and T_3 , in order to decide the importance level between the motion information and the transform coefficients. The thresholds and the corresponding sets of parity data rates assignments were chosen experimentally (see Table 1). In these experiments, the normalized average SADs of different type of video sequences were analyzed at the same encoding condition. Different thresholds are chosen for different types of video sequences which were all based on extensive test results. The parity data rates for each range of SADs are not designed to add up to the same number. When SAD is small ($SAD < T_1$), the least amount of the parity bits are transmitted to the decoder side. As SAD increases, higher amount of the parity bits are needed for correcting the lost packets. It also needs to mention that thresholds selection is dependent on the encoding data rate. A suggested range for T_1 , T_2 , and T_3 at encoding data rate of 512 kbps is: [23, 25], [11, 13] and [5, 7]. The parity data rates given in the Table 1 is the puncturing rate of each code word. For example, 1/8 is the total output Turbo encoding parity data rate, which means 1 out of every 16 parity bits is output from each convolutional encoder (refer to Figure 5). The experimental results given in section 4 showed that by using the parity data rate allocation and the thresholds decision in Table 1, the content adaptive unequal error protection can provide a better rate distortion performance and the visual quality of the decoded video sequences, comparing to our previously proposed unequal error protection. Both

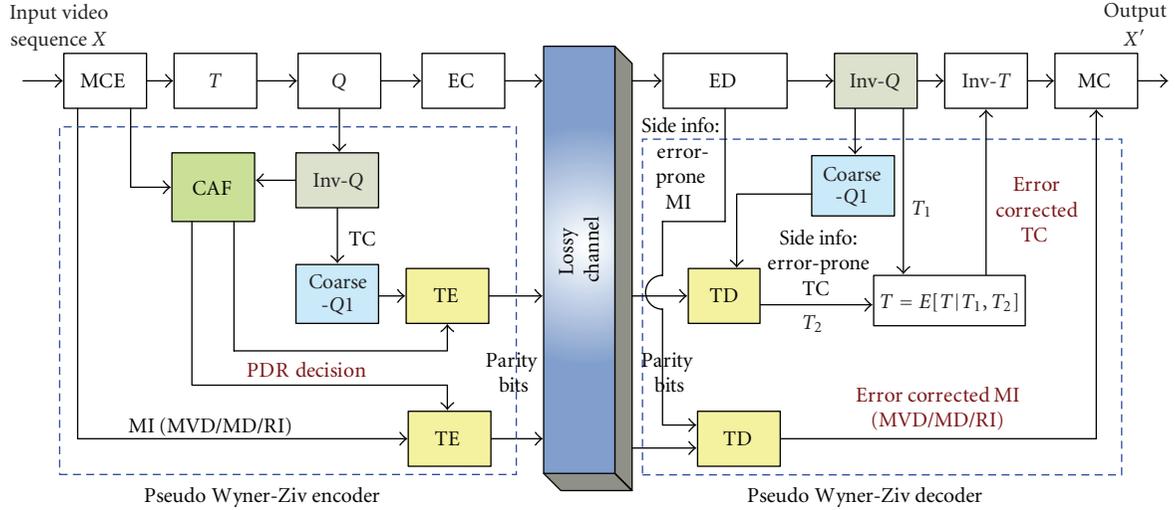


FIGURE 6: Content adaptive unequal error protection using Wyner-Ziv coding.

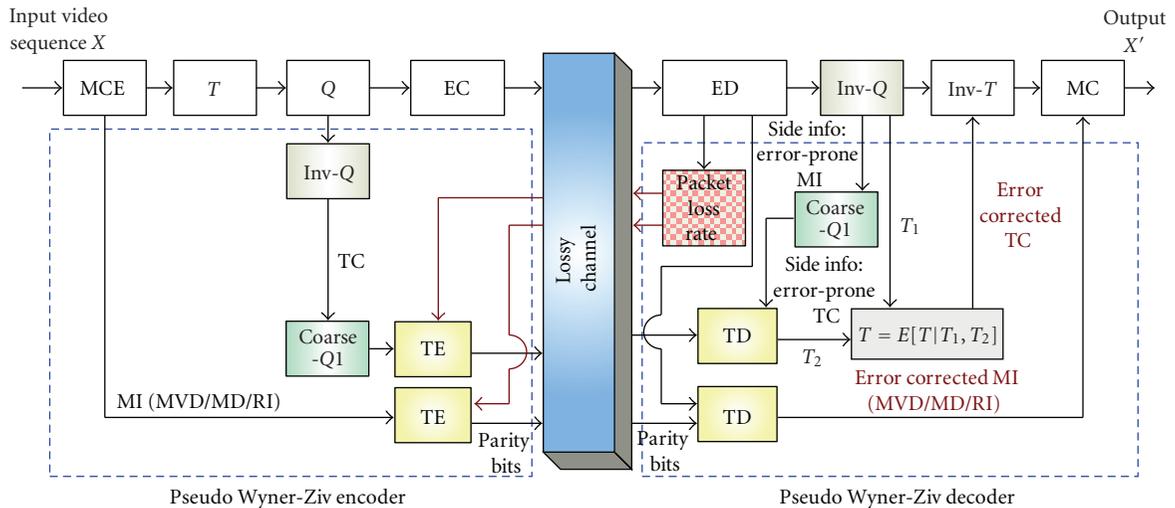


FIGURE 7: Feedback aided unequal error protection based on Wyner-Ziv coding.

techniques outperform the equal error protection case and the H.264 with error concealment case as shown in Section 4. However, depending on the channel condition and the sequence characters, it may not guarantee perfect recovery of the lost data in all cases. The calculation of the SAD and the comparison to the thresholds are straight forward, therefore it does not add much complexity to the system. The block diagram of the system is shown in Figure 6.

3.2. Feedback Aided Unequal Error Protection. Another approach to improve the unequal error protection is to exploit the feedback information of the channel loss rate from the decoder side. The parity data rates assigned for Turbo encoding the protected video information can accordingly be adjusted.

It is to be noted that data networks suffer from two types of transmission errors, namely random bit errors due to noise in the channels and packet losses due to network

congestion. When transmitting a data packet, a single uncorrected bit error in the packet header or body may result in the whole packet being discarded [28–33]. In the current work, we only consider packet losses, whether due to network congestion or uncorrected bit errors. When using the real-time transport protocol (RTP), determining which packets have been lost can be easily achieved by monitoring the sequence number field in the RTP headers [24, 34]. Therefore, the packet loss rate of each frame can be easily obtained at the decoder.

Figure 7 depicts a block diagram of the FBUEP. At the H.264/AVC encoder, each frame is divided into several slices. Both the motion information and the transform coefficients of each slice are sent to the Pseudo Wyner-Ziv encoder to be encoded independently by the two Turbo encoders. As for UEPWZ, the parity data rates allocated to protecting the different elements of the video sequence are assigned independently.

At the decoder, the packet loss rate of each frame is evaluated based on the received video information. It is then sent back to the two Turbo encoders via the RTCP feedback packets. Depending on the channel packet loss rates conveyed, the two Turbo encoders adjust the parity data rates for encoding the motion information and the transform coefficients of the current frame.

3.2.1. RTCP Feedback. In the decoder, the channel packet loss rate is obtained based on the received data and sent back to the Pseudo Wyner-Ziv encoder. If the available bandwidth for transmitting the feedback packets is above a certain threshold then an immediate mode RTCP feedback message is sent, otherwise the early feedback RTCP mode is used [35]. The two Turbo encoders update the parity data rates for encoding the motion information and the transform coefficients based upon the received RTCP feedback conveying the packet loss rates. This way the Pseudo Wyner-Ziv encoder attempts to adapt to the decoder's needs, while avoiding blindly sending a large number of parity bits that may not be needed when the packet loss is low or zero. In the case of high channel packet loss rate, the Pseudo Wyner-Ziv encoder enhances the protection by allocating more data rates to the Turbo encoded data, especially the motion information, while decreasing relatively the data rate used for encoding the main data stream by the H.264/AVC encoder. In this way, the total data rate is kept as a constant so that it will not exacerbate the possible congestion over the network transmission.

According to the RTCP feedback profile that is detailed in [35], when there is sufficient bandwidth, each loss event can be reported by means of a virtually immediate RTCP feedback packet. In the RTCP immediate mode, feedback message can be sent for each frame to the encoder. In our scheme an initial parity data rate value is set at the beginning of transmitting a video. When the channel loss condition changes, the immediate mode RTCP feedback packet sends the latest channel packet loss rate to the Turbo encoders to adjust the parity data rate assignment for the next frame. If we let N_L denote the average number of loss events to be reported every interval T by a decoder, B the RTCP bandwidth fraction for our decoder, and R the average RTCP packet size, then feedback can be sent via the immediate feedback mode when

$$N_L \leq \frac{B * T}{R}. \quad (1)$$

In the RTCP protocol profile [35], it was assumed that 2.5 percent of the the RTP session bandwidth is available for RTCP feedback from the decoder to the encoder. For example, for a 512 kbits/s stream, 12.8 kbits are available for transmitting the RTCP feedback. If we assume an average of 96 bytes (768 bits) per RTCP packet and a frame rate of 15 frames/second, then by (1), we can conclude that $N_L \leq 12800 * (1/15)/768 = 1.11$. In this case, the RTCP immediate mode can be used to send one feedback message per frame to the encoder.

When $N_L > B * T/R$, the available bandwidth is not sufficient for transmitting a feedback message via the immediate mode. In this case, the early RTCP mode is turned

on. In this mode, the feedback message is scheduled for transmission to the encoder at the earliest possible time, although it can not necessarily react to each packet loss event. In this case, a received feedback message at the encoder side may not reflect the latest channel loss rate. We therefore propose to send an estimate average channel packet loss rate based on packet loss rates of the previous k frames. It gives a better estimate of the recent channel packet loss rate. This scheme is detailed in Section 3.2.2.

When the Pseudo Wyner-Ziv encoder does not receive feedback regarding the current packet loss rates (the feedback packet got lost during transmitting back to the Turbo encoders or the available bandwidth is not sufficient for immediate mode feedback), the Turbo encoders keep using the last received channel packet loss rate to decide the parity data rates for encoding the motion information and the transform coefficients of the current encoded frame.

3.2.2. Delay Analysis. Delay must be considered when feedback is used. In our system, a RTCP feedback message is transmitted via the immediate mode, if the available RTCP transmission data rate is above the threshold as defined in (1). Through this mechanism the decoder reports the packet loss rate associated with each received frame to the encoder. The Pseudo Wyner-Ziv encoder then utilizes this information to select the parity data rates for encoding the motion information and the transform coefficients of the current encoded frame.

In early feedback mode, rather than sending feedback on a frame by frame basis, we propose to send the feedback packets to the Pseudo Wyner-Ziv encoder every k frames ($k = 1, 2, \dots, q$). The feedback in this case is the average channel packet loss rate (L_k^m) evaluated based on the history of the received video information of the past k frames, as given in (2). m represents the m th set of the k frames received at the decoder. In this equation $S_{i,j}$ is a counter counting the number of the error corrupted slices in the i th received frame. i is counted in terms of every k frames ($i = 0, \dots, k$). j is the index of the received slice and each frame is assumed to be partitioned into n slices. The parity data rates assignment, for Turbo encoding the motion information and the transform coefficients of the next k frames, is then updated once every k frames and therefore has higher resilience to the delay problem:

$$L_k^m = \frac{1}{K} \sum_{i=1}^k \sum_{j=1}^n S_{(i,j)}, \quad (2)$$

$$S_{(i,j)} = \begin{cases} 0, & \text{the error free packet is received,} \\ 1, & \text{the error corrupted packet is received.} \end{cases} \quad (3)$$

Furthermore, in the frame by frame based feedback strategy, if the packet loss rate of the current decoded frame is the same as the previous frame's, no feedback message needs to be sent back to the encoder. In the same way, if the average channel packet loss rate of the current received k frames (L_k^m) is equal to the average packet loss rate of the past k frames ($L_k^{(m-1)}$), no feedback is needed to be sent back to

TABLE 2: Parity data rate (PDR) assignment for FBUEP method.

Packet loss rate	Parity data rate assignment
$0 < N_{PL} \leq 11\%$	$PDR_{MI} = \frac{4}{16}, PDR_{TC} = \frac{2}{16}$
$11\% < N_{PL} \leq 22\%$	$PDR_{MI} = \frac{6}{16}, PDR_{TC} = \frac{2}{16}$
$22\% < N_{PL} \leq 33\%$	$PDR_{MI} = \frac{7}{16}, PDR_{TC} = \frac{3}{16}$
$33\% < N_{PL} \leq 44\%$	$PDR_{MI} = \frac{8}{16}, PDR_{TC} = \frac{4}{16}$
$44\% < N_{PL} \leq 55\%$	$PDR_{MI} = \frac{10}{16}, PDR_{TC} = \frac{4}{16}$
$55\% < N_{PL}$	$PDR_{MI} = \frac{12}{16}, PDR_{TC} = \frac{8}{16}$

the Pseudo Wyner-Ziv encoder. In other words, the feedback message is only sent back to the encoder when the packet loss rate is changed. Therefore, there are three scenarios when no feedback is received by the Turbo encoders. One is that the channel packet loss rate is kept as a constant at the moment. Another case is that the feedback packet got lost during transmitting back to the Turbo encoders. The third case is that the available bandwidth is not sufficient for immediate mode feedback. Accordingly, the Turbo encoders only update the parity data rates for encoding the motion information and the transform coefficients when they received the updated feedback regarding the latest packet loss rate.

3.2.3. Data Rate Assignment between Primary Encoding and the Pseudo Wyner-Ziv Encoding. When packet loss rates increase, simply increasing the parity data rates for Turbo encoding the motion information and the transform coefficients while keeping the same data rate for the primary video data coding would only exacerbate channel congestion [24]. A better way would be to reduce the data rate allocated to the primary video data transmission slightly and correspondingly increase the data rate allocated to the transmission of parity bits, so that the total transmission data rate at any packet loss rate is kept constant. Furthermore, more efficient use of the data rate can be achieved by assigning different protection levels to the motion data and the transform coefficients in the Pseudo Wyner-Ziv encoder at different channel packet loss rate.

In our scheme, the parity data rates assigned to the motion information and the transform coefficients were evaluated experimentally. The parity data rates settings at different range of channel packet loss rate were tested by extensive experiments on different video sequences. The experiment results showed that the enough lost information can be corrected for reconstructing a visually good quality decoded frames (See Table 2).

4. Experiments

To evaluate the proposed techniques, experiments were carried out using the JM10.2 H.264/AVC reference software.

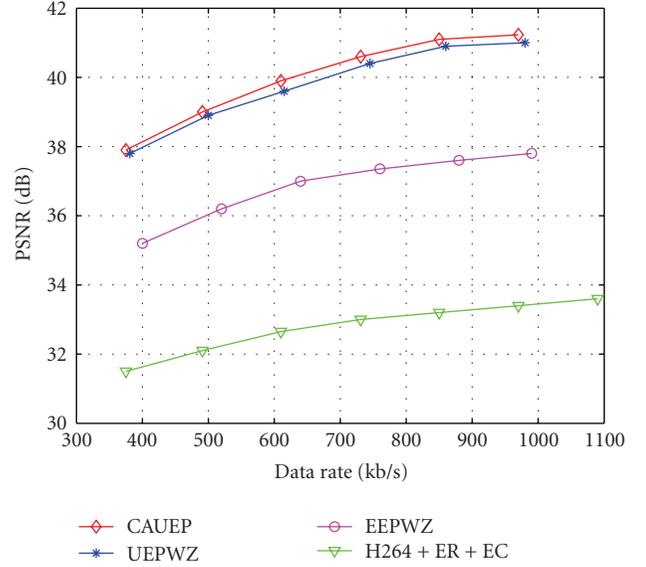


FIGURE 8: Rate-distortion performance of *foreman.qcif* at fixed packet loss rate. The results of CAUEP, UEPWZ, EEPWZ and H.264 + ER + EC are compared. CAUEP achieved the best performance but close to that of the UEPWZ due to the content of the video sequence.

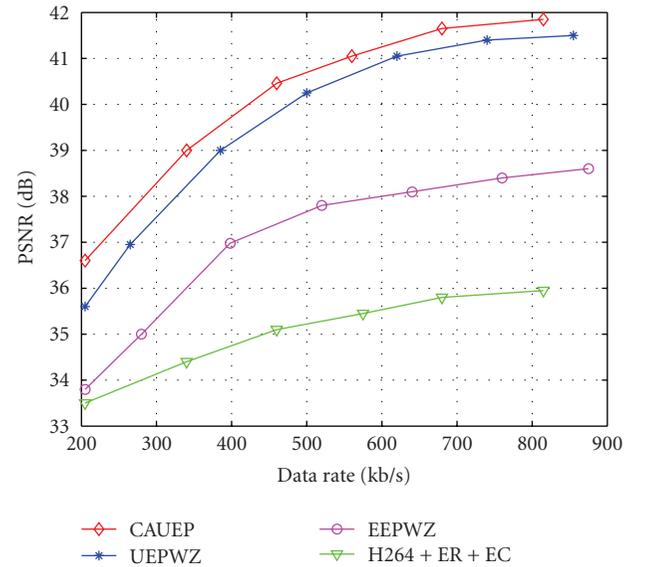


FIGURE 9: Rate-distortion performance of *carphone.qcif* at fixed packet loss rate. For this sequence CAUEP outperform UEPWZ by 0.3 to 1 dB.)

The frame rate for each sequence was set at 15 frames per second with an *I-P-P-P...GOP* structure. In our experiment, each QCIF frame is divided into 9 slices. The primary encoded video data output from the H.264 encoder are packetized into 9 packets per frame, each containing the video information of one slice. The Turbo encoded parity bits of the motion information and the transform coefficients corresponding to each slice are also sent in separate packets. All three types of the packets are subjected to random losses

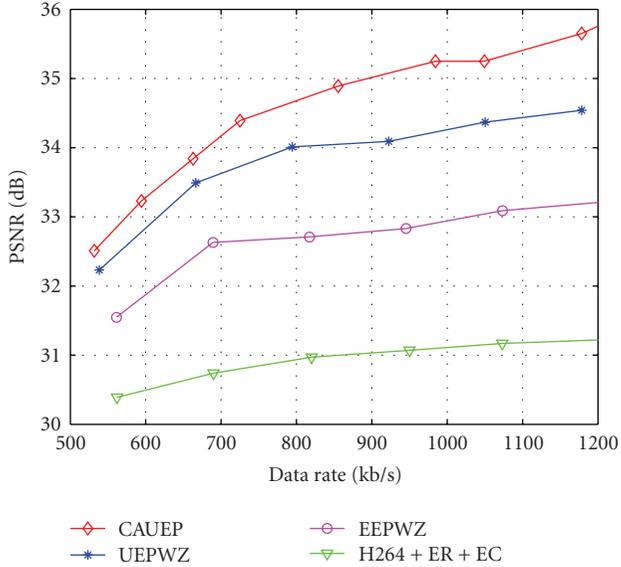


FIGURE 10: Rate-distortion performance of *stefan.qcif* at fixed loss rate. For this sequence and a packet loss rate of 22%, the CAUEP outperform the UEPWZ by 0.3 to 1.12 dB.

over the transmission channel. We did not attempt to make all packets the same size. Since the packets containing the parity bits of the motion information or the transform coefficients are much smaller in size comparing to the H.264 packets, the possibility of getting lost over a wireless network transmission is therefore much smaller. All the experiments results were averaged over 30 lossy channel transmission realizations.

As has been mentioned in Section 3.2, data networks suffer from two types of transmission errors: random bit error and packet drop. In our experiments, we only consider the case of packet erasures, whether due to network congestion or uncorrected bit errors. Lower the total data rate to reduce the network congestion is a realistic solution when packet loss is very high. However, since our main application is for video streaming over wireless networks in which case the packet loss situation is more complicated, we did not consider it in our current experiments. It is to be noted that simply increasing the parity bits when the packet loss rate increases is not applicable, since it will exacerbate network congestion (see [20]). Instead, the total transmission data rate should be kept constant, which means that when the packet loss rate increases, the primary data transmission rate should be lowered in order to spare more bits for parity bits transmission.

In our experiments, channel packet loss is simulated by using uniform random number generators. Our algorithm focuses on wireless network application in which case severe packet loss could happen. In the case of wireless network transmission, the probability that the packet arrives in error is approximately proportional to its length [12]. Assume the length of the H.264 data packet is l_h , and the lengths of the parity bits packets containing the motion information and the transform coefficients are l_{wm} and l_{wt} , respectively. If the probability of the packet loss of H.264 data is r_h , then the

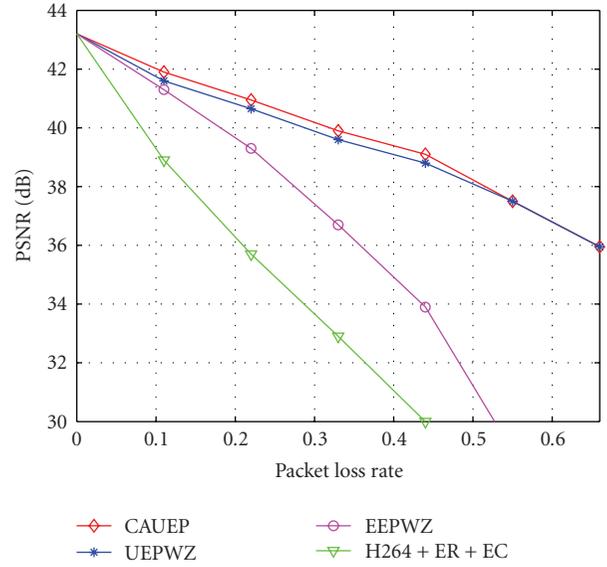


FIGURE 11: Packet loss rate performance of *foreman.qcif*.

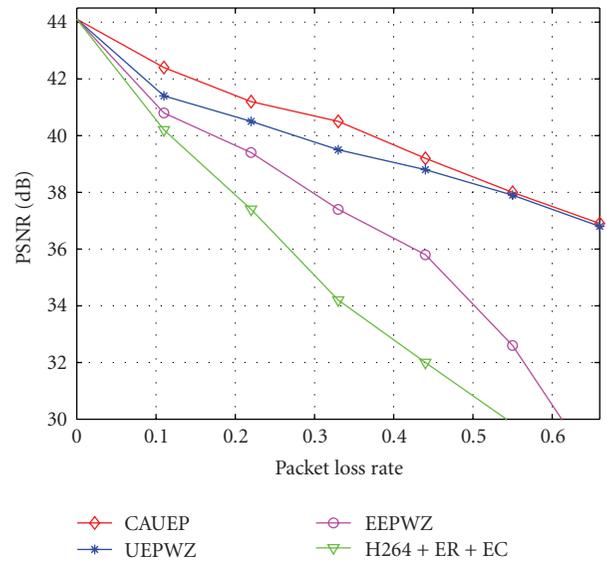


FIGURE 12: Packet loss rate performance of *carphone.qcif*.

probabilities of the packet loss of the motion information and the transform coefficients packets are $r_{wm} = r_h l_{wm} / l_h$ and $r_{wt} = r_h l_{wt} / l_h$, respectively. This is implemented in our packet loss simulation.

In our Wyner-Ziv based schemes, different parity data rate settings have been tested extensively for different types of video sequences. For the tested sequences, the final decision on the parity data rate assignments that are given in the paper can achieve a better rate distortion performance, the visual quality of the decoded frames and the overall data rate usage comparing to other values of the parity data rates.

Figures 8 and 9 show the results for fixed packet losses in which the channel packet loss rate is always fixed at 33% for the two sequences *foreman* and *carphone*, respectively. To see the performance comparison at a different fixed packet

loss rate, the *stefan.qcif* sequence is used to generate the results at 22% packet loss case as shown in Figure 10. It is noted that for fixed losses FBUEP offers no advantage over UEPWZ. In fact, when both use the same parity data rates, their performance will be identical. For this reason, we do not include the results of the FBUEP in Figures 8, 9 and 10. For EEPWZ and UEPWZ methods, the PDR are fixed through transmitting a video sequence. For primary video encoding at a data rate of 512 kbps, the corresponding parity data rate assigned to Turbo encoding the motion information and the transform coefficients are 1/4 and 1/8. For EEPWZ method, the parity data rates allocation for motion information and the transform coefficients in this case are both 3/16. For UEPWZ and EEPWZ methods, the parity data rate assignment is always fixed. The data rate allocation between the primary video layer and the parity layer is kept at 1 : 5. For FBUEP and CAUEP methods, the parity data rate assignments are always adaptive to the content of the frame or the channel packet loss rate. The overall average data rate used for parity bits and the primary video data transmission should also be kept equal to or less than 1 : 5.

As can be observed from the figures, CAUEP has the best performance, outperforming UEPWZ by around 0.2 dB in the case of *foreman* sequence and by around 0.3 to 1 dB in the case of *carphone* and *stefan*. When using EEPWZ the motion information and the transform coefficients were provided the same protection level. The EEPWZ is a similar case of SLEP since the motion information and the transform coefficients are protected at the same parity data rate. The difference is that in the EEPWZ case, the parity bits of the motion information and the transform coefficients are sent in individual packets. This makes the experiment results comparable with our unequal error protection based methods. The curve of H264 + ER + EC shows the result of the H.264 using slice group feature for error resilience in the encoding process and the previous colocated slice replacement for the error concealment strategy in the decoding process. All four schemes use the same amount of total data rate. Wyner-Ziv based methods allocated part of the total data rate budget to transmit the information protected via the Pseudo Wyner-Ziv codec. In the H.264 with error concealment, the total data rate is all allocated for transmitting the H.264 encoded video information. We think this is a fair comparison since the total data rate is the same for all the tested schemes. It can be seen from the experiment results that the rate distortion performance and the visual quality can both be improved by sparing certain amount of total data rate for protecting the important video information by Pseudo Wyner-Ziv coding.

Figures 11 and 12 exhibit the average performance of the four strategies when the packet losses range from 0 to 66% for *foreman* and *carphone qcif* sequences. The total data rate was kept around 512 kbps and the packet loss rates at 11%, 22%, 33%, 44%, 55% and 66% have been tested. Again, CAUEP outperforms the other three techniques. Compared to UEPWZ, CAUEP gains about 0.2-0.3 dB for *foreman* and 0.5–1 dB for *carphone*, and its performance converges to that of UEPWZ as the packet loss rate becomes

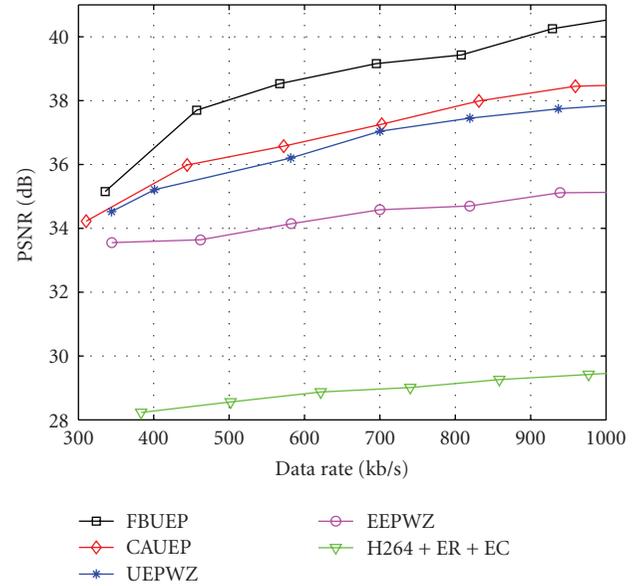


FIGURE 13: Rate distortion performance (*foreman-qcif*) (dynamic packet loss case).

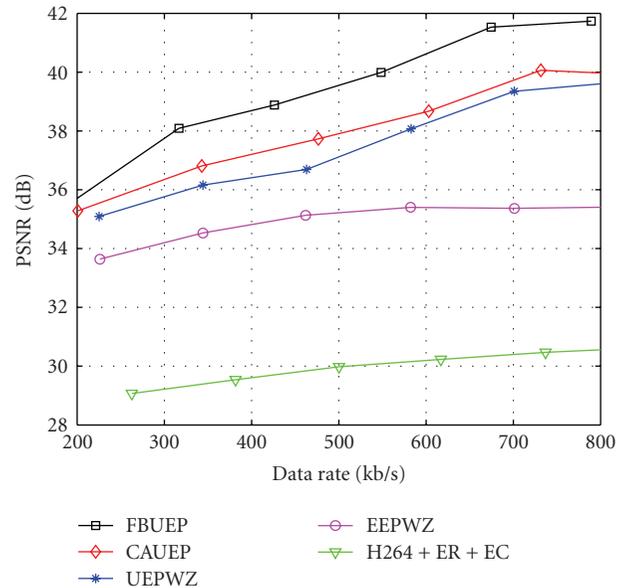


FIGURE 14: Rate distortion performance (*carphone-qcif*) (dynamic packet loss case).

severe. This is because both techniques breaks down due to too serious packet loss and insufficient data rate available for error correction.

In general, channel conditions change over time, resulting in variable packet loss rates. In the following experiments, the channel packet loss rates were varied during the transmission time of the video sequences. In our simulation, the lowest packet loss rate is 0 while the highest possible packet loss rate is 55%. The mean of the overall channel packet loss is at 23.2%. The parity data rates allocated to the motion information and the transform coefficients, in the case of FBUEP, are shown in Table 2.

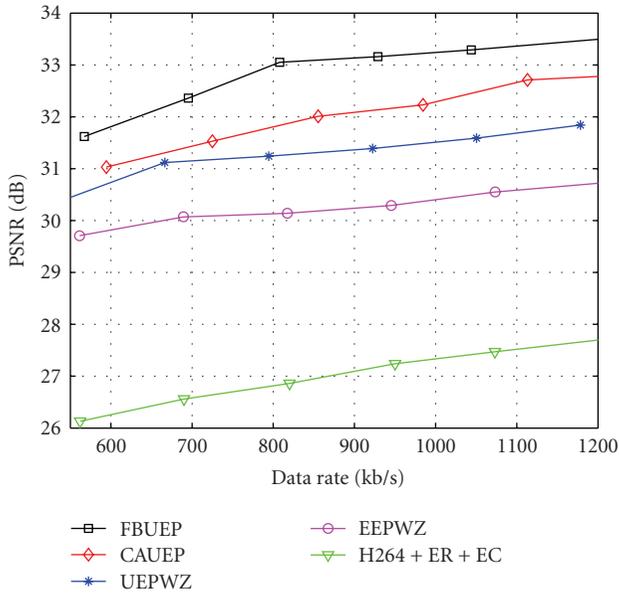


FIGURE 15: Rate distortion performance (stefan-qcif) (dynamic packet loss case).



FIGURE 16: Visual comparison between the original 85th frame (a) and that produced by CAUEP (b) (PSNR = 38.42 dB).

Figures 13, 14, and 15 depict the results for the dynamic packet loss case. As can be seen in the dynamic packet loss case, the CAUEP and UEPWZ schemes achieved lower PSNRs at the same data rates compared to those in Figures 8, 9, and 10. One of the reasons is that the CAUEP and the UEPWZ schemes were not able to allocate enough parity bits for protecting the important video information when the channel packet loss rates became higher. Furthermore, distortion is accumulated over a sequence of successive P frames due to motion compensation, until a new I frame is inserted. Unlike the other schemes, FBUEP attempts to be aware of the varying packet loss rates and is therefore able to adjust the parity data rates accordingly.

For visual comparison, the 85th frame from *foreman*, which was protected via the various schemes described above, has been decoded and depicted along with the original frame in Figures 16, 17, and 18. The results presented are for dynamic packet losses. It can be seen that both UEPWZ and CAUEP produce block artifacts on the left and right cheeks of the person in the figure, with CAUEP generating less artifacts than UEPWZ. It is also observed that the use of feedback, as



FIGURE 17: Visual comparison between the 85th frame produced by FBUEP (a) ($k = 5$, PSNR = 39.75 dB) and UEP (b) (PSNR = 37.15 dB).



FIGURE 18: Visual comparison between the 85th frame produced by EEP (a) (PSNR = 34.64 dB) and H264 + EC (b) (PSNR = 29.12 dB).

in FBUEP, produced the most visually pleasing image. It also has higher PSNR values than the others.

Figure 18 compares the results from using EEP and the H.264/AVC with error concealment applied to the decoded frames. Both the visual quality and the PSNRs are much worse than those of UEPWZ, CAUEP and FBUEP.

5. Conclusion

This paper described and compared three error resilience techniques each utilizing a Pseudo Wyner-Ziv codec to protect important video information produced by an H.264/AVC codec. In each scheme the motion information and the transform coefficients are protected independently. In the first scheme, unequal error protection using Pseudo Wyner-Ziv coding (UEPWZ), motion information and the transform coefficients are provided fixed albeit different protection levels for the entire video sequence. In the second method, content adaptive unequal error protection (CAUEP), the protection afforded motion information and the transform coefficient were updated each frame according to frame content. The third technique, feedback aided unequal error protection (FBUEP), utilized packet loss rates sent from the decoder to the encoder to choose the parity data rates allocated to encode the motion information and the transform coefficients. It was demonstrated that UEPWZ, CAUEP, and FBUEP are more error resilient to packet losses than equal error protection techniques and provide more visually pleasing images. It was also shown that FBUEP is better suited for handling time varying losses while CAUEP

has better performance in the presence of fixed losses. This paper aims to show the different amount of contribution that could be obtained from each algorithm. Future work will focus on combining both CAUEP and the FBUEP to develop a more efficient error resilient technique. In addition, we will carry out a study on the system's complexity.

Acknowledgment

This work was supported by a grant from the Indiana Twenty-First Century Research and Technology Fund.

References

- [1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.
- [2] D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- [3] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video-applications to compression and error resilience," in *Proceedings of the IEEE Data Compression Conference (DCC '03)*, pp. 93–102, Snowbird, Utah, USA, March 2003.
- [4] A. Aaron, S. Rane, D. Rebollo-Monedero, and B. Girod, "Systematic lossy forward error protection for video waveforms," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 609–612, Barcelona, Spain, September 2003.
- [5] S. Rane, A. Aaron, and B. Girod, "Systematic lossy forward error protection for error resilient digital video broadcasting," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 588–595, San Jose, Calif, USA, January 2004.
- [6] S. Rane and B. Girod, "Systematic lossy error protection versus layered coding with unequal error protection," in *Wyner-Ziv Video Coding*, vol. 5685 of *Proceedings of SPIE*, pp. 663–671, San Jose, Calif, USA, January 2005.
- [7] S. Rane and B. Girod, "Analysis of error-resilient video transmission based on systematic source-channel coding," in *Proceedings of Picture Coding Symposium (PCS '04)*, pp. 453–458, San Francisco, Calif, USA, December 2004.
- [8] S. Rane, A. Aaron, and B. Girod, "Systematic lossy forward error protection for error resilience digital video broadcasting—a Wyner-Ziv coding approach," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3101–3104, Singapore, October 2004.
- [9] S. Rane, A. Aaron, and B. Girod, "Error resilient video transmission using multiple embedded Wyner-Ziv descriptions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '05)*, vol. 2, pp. 666–669, Genoa, Italy, September 2005.
- [10] S. Rane and B. Girod, "Systematic lossy error protection of video based on H.264/AVC redundant slices," in *Visual Communications and Image Processing*, vol. 6077 of *Proceedings of SPIE*, pp. 1–9, San Jose, Calif, USA, January 2006.
- [11] S. Rane, P. Baccichet, and B. Girod, "Modeling and optimization of a systematic lossy error protection system based on H.264/AVC redundant slices," in *Proceedings of the 25th Picture Coding Symposium (PCS '06)*, Beijing, China, April 2006.
- [12] P. Baccichet, S. Rane, and B. Girod, "Systematic lossy error protection based on H.264/AVC redundant slices and flexible macroblock ordering," *Journal of Zhejiang University*, vol. 7, no. 5, pp. 900–909, 2006.
- [13] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.
- [14] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: a video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2436–2448, 2007.
- [15] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249–258, 2004.
- [16] J. Wang, A. Majumdar, K. Ramchandran, and H. Garudadri, "Robust video transmission over a lossy network using a distributed source coded auxiliary channel," in *Proceedings of Picture Coding Symposium (PCS '04)*, pp. 41–46, San Francisco, Calif, USA, December 2004.
- [17] J. Wang, V. Prabhakaran, and K. Ramchandran, "Syndrome-based robust video transmission over networks with bursty losses," in *IEEE International Conference on Image Processing*, pp. 741–744, Atlanta, Ga, USA, October 2006.
- [18] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: turbo-codes," in *Proceedings of International Conference on Communications*, pp. 1064–1070, Geneva, Switzerland, May 1993.
- [19] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.
- [20] Q. Xu, V. Stankovic, and Z. Xiong, "Layered Wyner-Ziv video coding for transmission over unreliable channels," *Signal Processing*, vol. 86, no. 11, pp. 3212–3225, 2006.
- [21] L. Liang, P. Salama, and E. J. Delp, "Unequal error protection using Wyner-Ziv coding for error resilience," in *Visual Communications and Image Processing*, vol. 6508 of *Proceedings of SPIE*, pp. 1–9, San Jose, Calif, USA, January 2007.
- [22] L. Liang, P. Salama, and E. J. Delp, "Content-adaptive unequal error protection based on Wyner-Ziv coding," in *Proceedings of Picture Coding Symposium (PCS '07)*, Lisbon, Portugal, November 2007.
- [23] L. Liang, P. Salama, and E. J. Delp, "Feedback-aided error resilience technique based on Wyner-Ziv coding," in *Visual Communications and Image Processing*, vol. 6822 of *Proceedings of SPIE*, pp. 1–9, San Jose, CA, USA, January 2008.
- [24] M. Johanson, "Adaptive forward error correction for real-time internet video," in *Proceedings of the 13th Packet Video Workshop (PV '03)*, Nantes, France, April 2003.
- [25] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, Wiley, Chichester, UK, 2003.
- [26] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [27] W. E. Ryan, *Concatenated Codes and Iterative Decoding*, John Wiley & Sons, New York, NY, USA, 2003.
- [28] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [29] X. Zhu and B. Girod, "Video streaming over wireless networks," in *Proceedings of European Signal Processing Conference (EUSIPCO '07)*, Poznan, Poland, September 2007.
- [30] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: does burst-length matter?"

- in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03)*, vol. 5, pp. 684–687, Hong Kong, April 2003.
- [31] K. Stuhlmüller, N. Farber, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.
- [32] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and H. J. Chao, “MPEG-4 compressed video over the Internet,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '99)*, vol. 4, pp. 327–331, Orlando, Fla, USA, May-June 1999.
- [33] Y. T. Hou, D. Wu, W. Zhu, H.-J. Lee, T. Chiang, and Y.-Q. Zhang, “End-to-end architecture for MPEG-4 video streaming over the Internet,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP '99)*, vol. 1, pp. 254–257, Kobe, Japan, October 1999.
- [34] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, “H.264/AVC in wireless environments,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.
- [35] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Ray, “Extended RTP profile for real-time transport control protocol (RTCP)-based feedback (RTP/AVPF),” July 2006.