Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 79769, Pages 1–13
DOI 10.1155/ASP/2006/79769

RD Optimized, Adaptive, Error-Resilient Transmission of MJPEG2000-Coded Video over Multiple Time-Varying Channels

Scott Bezan and Shahram Shirani

Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1

Received 25 February 2005; Revised 22 August 2005; Accepted 1 September 2005

To reliably transmit video over error-prone channels, the data should be both source and channel coded. When multiple channels are available for transmission, the problem extends to that of partitioning the data across these channels. The condition of transmission channels, however, varies with time. Therefore, the error protection added to the data at one instant of time may not be optimal at the next. In this paper, we propose a method for adaptively adding error correction code in a rate-distortion (RD) optimized manner using rate-compatible punctured convolutional codes to an MJPEG2000 constant rate-coded frame of video. We perform an analysis on the rate-distortion tradeoff of each of the coding units (tiles and packets) in each frame and adapt the error correction code assigned to the unit taking into account the bandwidth and error characteristics of the channels. This method is applied to both single and multiple time-varying channel environments. We compare our method with a basic protection method in which data is either not transmitted, transmitted with no protection, or transmitted with a fixed amount of protection. Simulation results show promising performance for our proposed method.

1. INTRODUCTION

Video data is very large in its raw form and as such requires some level of source coding (compression) in order to be effectively transmitted. Unfortunately, many compression methods (usually of the lossy variety) introduce distortion in the reconstructed frame. In addition to the quantization distortion, source coding can leave the coded data vulnerable to bit errors that frequently occur in transmission over error-prone channels. If a constant source rate is assumed, any further distortion seen in the reconstructed frame will be due to channel-introduced errors. To this end, error protection (channel coding) is introduced to the source-coded data in order to enable correction of some erroneous bits at the receiver. Ideally, the channel coder could add as much channel code as necessary to correct all bit errors in transmission. In reality, transmission channels are bandlimited, so this solution is infeasible. Therefore, a rate-distortion (RD) tradeoff exists between the amount of error correction added, while considering a channel bandwidth constraint and the reconstructed distortion influenced by channel errors. Finding the optimum balance between rate and distortion is challenging, but more complication arises when channels are time-varying. In this case, when the condition of a channel changes with time, past optimized channel codes may no longer be relevant, therefore some manner of adaptation to the channel condition is required.

In discussing RD optimization of a bitstream transmitted over an error-prone channel, it is advantageous to consider blocks of the original data stream. These blocks, or coding units, divide a normally large amount of data into more manageable units. Also, in situations involving multiple channels, if channel resources are not sufficient for transmitting the entire bitstream, one can divide the bitstream using these coding units over multiple channels. Unfortunately, the dependence of one coding unit on another creates a challenge that must be addressed as errors occurring in presently decoded coding units will affect the decoding of future ones.

Over the past few years, many researchers have examined the problem of RD optimization of multimedia data. In [1], joint source-channel coding (JSCC) was performed on a wavelet-based source-coded bitstream using the bit sensitivities of the wavelet coefficients. An operational RD function was then constructed based on rate-compatible punctured convolutional (RCPC) channel codes. Unequal error protection was employed in [2] on a layered source coding scheme using Reed-Solomon codes to achieve JSCC. An iterative approach was used to find the appropriate source and channel code rates across a binary symmetric channel. JSCC was also employed in [3] using a progressive source coder and RCPC channel codes to find an exact solution for optimal channel code allocation.

In [4], JSCC was employed using a progressive wavelet source coder and the concatenation of RCPC and a cyclic redundancy-check code for channel error protection. The bitstream was transmitted over a binary symmetric channel and it was stated that the only effect that channel noise had on the system was degradation due to lower-rate source coding. Punctured turbo codes were used in [5] in conjunction with the JPEG2000 source coding standard to employ JSCC over a single binary symmetric channel. JSCC was performed to yield packets of fixed size resulting in a rate-allocation problem that grows exponentially with the number of packets to be transmitted.

New algorithms were proposed in [6] to find optimal solutions to uneven error protection problems involving scalable source-coded bitstreams. It was shown in [6] that the complexity of their algorithm decreases dramatically if information of the convexity of the source coder is known. JSCC was employed using a concatenation of Reed-Solomon, RCPC, and cyclic redundancy-check codes. The SPIHT encoder and RCPC codes were used by [7] for source and channel coding, respectively, and transmitted the protected bitstream over a time-varying channel. A method of subsampling the SPIHT source encoder was proposed by [8] to achieve multiple descriptions of the original data.

In summary, [1–8] performed JSCC on data to be transmitted over a single channel. In this paper, channel code was added in a forward error correction manner to preencoded source data transmitted over *multiple time-varying channels*. A multicast scenario was discussed in [9], however, our paper considers multiple-channel transmission to a single user.

Forward error correction was performed by [10] under various channel conditions, however, the amount of channel code added was fixed to the particular channel condition. In [11], forward error correction was added to various channel conditions using a dynamic programming approach, however, it was performed on mean packet loss rates, not a time-varying channel situation.

In this paper, JPEG2000 is applied using frames of video data, appended with channel code and transmitted over multiple error-prone, time-varying channels. The channel coder chosen is a rate-compatible punctured convolutional (RCPC) coder which allows the generation of multiple code rates from a single encoder. Analysis is performed in order to determine the RD optimal amount of channel code to add to the source-coded frame based on the condition of the time-varying channel(s). As JPEG2000 compression is performed using frames of a video sequence, this is analogous to Motion JPEG2000 (MJPEG2000). In terms of compression efficiency, other video compression standards outperform MJPEG2000 due to the use of motion-compensated prediction between frames. However, MJPEG2000 encodes each frame independent of others and therefore offers superior performance when considering video editing and error robustness [12]. In addition, MJPEG2000 makes use of the wavelet transform which adds the advantage of a lower computational complexity than the calculation and manipulation of motion vectors used in MPEG video compression [13].

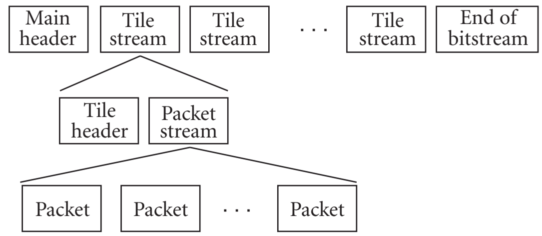


Figure 1: JPEG2000 bitstream [15].

Lastly, an argument could be made that a comparison between the approach proposed in this paper and multiple description coding with path diversity would be needed. Unfortunately, this was difficult due to the lack of research performed on multiple description coding over time-varying channels.

The paper is organized as follows. Section 2 discusses the JPEG2000 source and RCPC channel coder and channel model used herein. Section 3 discusses the proposed method of the paper by introducing the rate-distortion optimization techniques used. Next, experimental results for transmission across both single and multiple time-varying channels are outlined in Section 4. Finally, Section 5 concludes the paper.

2. BACKGROUND

2.1. Source coding

The JPEG2000 image compression standard was created to overcome the drawbacks of the existing widely used JPEG standard [10]. MJPEG2000 is the extension to video coding of the JPEG2000 standard where more attention is payed to error robustness and for the purpose of use in high-quality video systems [14]. Conversely, in MPEG video coding where compression efficiency is paramount, interframe coding (as well as intra-frame coding) is realized using motion-compensated prediction resulting in the propagation of errors throughout the decoded video sequence due to the interframe dependencies. In MJPEG2000, each frame is intra-coded without any dependency of surrounding frames. If errors occur in transmission, they will be confined to the frame in which they occurred [13].

Frames in MJPEG2000 are first divided into user-defined, nonoverlapping rectangular subframes called tiles which are coded independently using a combination of wavelets, quantization, and arithmetic encoding. The final JPEG2000 bitstream results in a sequence of headers detailing the encoding process, followed by the compressed data (see Figure 1). Following the main header are the independently coded tile streams. Each tile stream is comprised of a tile header and a stream of packets. The stream of packets is a packetized version of the compressed data [15]. Finally, the bitstream is concluded with a header indicating the end of the coded data.

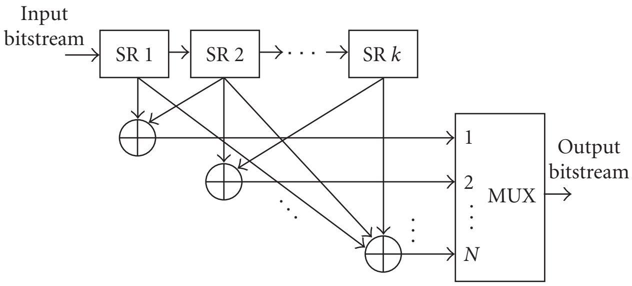


FIGURE 2: Basic convolutional encoder.

From the JPEG2000 encoder, two types of coding units are used here: tiles and packets. When tiles are the coding unit in question, the original frame will be encoded into a number of tiles and the packet stream in each tile is limited to one packet. Conversely, when packets are used, the entire frame is encoded as one large tile and multiple packets.

2.2. Channel coding

Channel coding is imperative when source-coded data is to be transmitted across error-prone channels. Convolutional codes are a type of channel code that interleaves redundant data into the bitstream so that erroneous bits can be corrected at the receiver. A convolutional encoder combines the present input bit with N combinations of up to $k-1$ past input bits (Figure 2). The N output bits are multiplexed to yield a continuous bitstream. A binary generator matrix $\mathbf{G}$ of size $k \times N$ details the connectivity between the input bits and the N output bits.

To achieve multiple rates from a single encoder, an $N \times P$ puncture matrix is inserted between the N output substreams and the multiplexer (not represented in Figure 2), where P is denoted as the "puncturing period." Blocks of P bits from the N output substreams are clocked into the puncture matrix. Wherever a "0" appears in the matrix, the corresponding bit in that position of the substream is discarded. Therefore, from an original code rate of $1/N$, multiple rates can be generated. In (1), r_c is the channel coding rate and $1 \leq z \leq (N-1)P$, where z is a parameter that is indicative of the amount of puncturing [16]:

$$r_c = \frac{P}{P+z}. \tag{1}$$

Introducing a rate-compatibility restriction on the puncture matrices ensures a mapping between different code rates. For a set of puncture matrices $p(z)$, the rate-compatibility restriction states that codes of a high code rate are embedded in those of a low code rate. This is detailed in [16] as follows:

$$\text{if } p_{ij}(z_o) = 1, \quad \text{then } p_{ij}(z) = 1 \quad \forall z \geq z_o,\ z_o \geq 1 \tag{2}$$

or, equivalently,

$$\text{if } p_{ij}(z_o)=0, \quad \text{then } p_{ij}(z)=0 \quad \forall z \leq z_o,\ z_o \leq (N-1)P-1, \tag{3}$$

where i and j are indices for the rows and columns of $p(z)$, respectively.

2.3. Channel model

Practical transmission channels are rarely static in terms of their characteristics. Since this paper examines the situation where the bit error rate of transmission channels varies with time, channels were simulated using additive white Gaussian noise (AWGN) to introduce random bit errors and an autoregressive model was used to simulate time-varying nature of the channel's characteristics. The autoregressive model used was as follows:

$$\text{BER}(n) = a\,\text{BER}(n-1) + w(n), \tag{4}$$

where BER denotes the bit error rate, n is the time index, a is a correlation factor, and $w(n)$ is a factor of white Gaussian noise. Therefore, the BER at the current time index is based on the most current past BER, weighted by the factor a and white Gaussian noise $w(n)$. A similar model was used in [17, 18] to characterize time-varying channels. As the BER changes by (4), the amount of error correction code should adapt accordingly.

3. PROPOSED ANALYSIS METHOD

Figure 3 shows the block diagram of the system used in this paper. The raw frame data (only the Y-component of the YUV sequence was utilized) is fed into the source coding block. Here, the raw frame is encoded at a constant rate into a number of coding units using MJPEG2000. After the completion of the source encoding, the coding units are then channel coded using an RCPC coder. To find the optimum amount of channel code to add at a particular instant of time, the RD optimization block polls the channels for their particular condition. With this information, the RD optimization block performs an analysis on the coding units of the frame. This analysis yields the appropriate amount of channel code to add to the coding units in order to minimize the reconstructed distortion while maintaining a rate budget dictated by the channel(s) and a transmission delay. On the receiver side, the received data is first channel decoded using the Viterbi maximum likelihood algorithm, then source decoded using the MJPEG2000 decoder. We show next how to adaptively optimize the amount of error correction code added to a video source-coded bitstream based on the conditions of time-varying channels. The details of the proposed analysis method are as follows.

As mentioned previously, an RCPC coder is used as the channel coder. Assume M puncture matrices $\mathbf{p}$ each yielding a particular amount of protection. Each coding unit i can then be encoded into M possible rates $\mathbf{r}_i$. The coding units encoded at a particular rate will have an associated distortion $\mathbf{d}_i$.

$$\begin{aligned}
\mathbf{p} &= \{p(1), \ldots, p(M)\}, \\
\mathbf{r}_i &= \{r_{i,p(1)}, \ldots, r_{i,p(M)}\}, \\
\mathbf{d}_i &= \{d_{i,p(1)}, \ldots, d_{i,p(M)}\}.
\end{aligned} \tag{5}$$

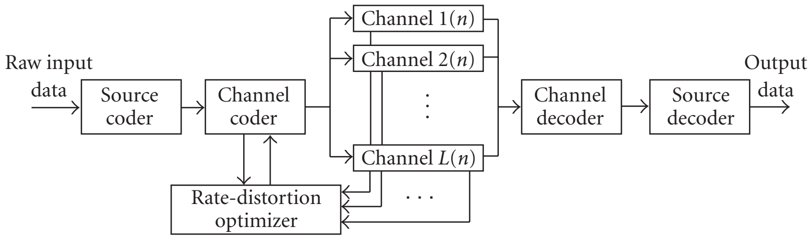


Figure 3: System diagram.

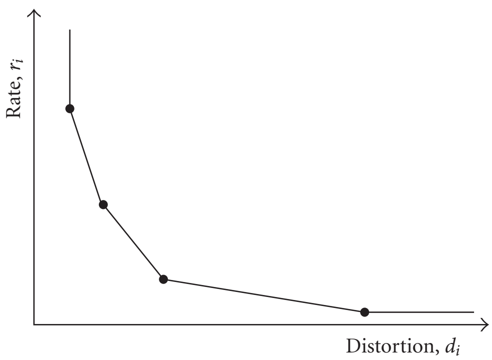


Figure 4: Rate-distortion map of operating points, where i represents the ith coding unit.

The pairs $(r_{i,p(m)}, d_{i,p(m)})$, $m = 1, \ldots, M$, create operating points on an RD map for the ith coding unit (Figure 4). Assuming that the source coder encodes the data into T coding units, an optimal bit allocation can be performed to yield a minimum reconstructed distortion while maintaining a bandwidth constraint. An optimal result can be found using dynamic programming as in [5, 6] as well as the Lagrangian method as in [2, 19]. The Lagrangian method was preferred for its ease of implementation and computational advantages [20].

3.1. Independent coding units

Assuming T independent coding units (that can be coded independently), the total distortion is calculated as

$$D = \sum_{i=1}^{T} d_{i,p(m_i)}, \tag{6}$$

where $p(m_i)$ is a selected puncture matrix from the set $\{p(1), \ldots, p(M)\}$ for coding unit i. We wish to minimize the total distortion subject to a channel defined rate constraint, that is,

$$\min D \quad \text{subject to } R = \sum_{i=1}^{T} \leq R_c. \tag{7}$$

In the above equation, $R_c = B \cdot \tau_m$ is the channel rate constraint, B is the bandwidth in (bps), and τ_m is the maximum acceptable delay in transmission of the frame. The transmission delay τ is defined as the amount of time it takes to transmit T coding units over a channel with a certain bandwidth of B bps and it is given by

$$\tau = \frac{\sum_{i=1}^{T} r_{i,p(m_i)}}{B}, \tag{8}$$

where

$$r_{i,p(m_i)} = \frac{r_{s,i}}{r_{c,i}}, \tag{9}$$

and $r_{s,i}$ is the rate in bits for the ith coding unit available from the source coder. $r_{c,i}$ is the channel code rate based on $p(m_i)$, where the subscript i is added to r_c to indicate the channel coding rate for coding unit i.

Using Lagrangian optimization, we can define a Lagrangian cost function J_i for each coding unit given by

$$J_i = d_{i,p(m_i)} + \lambda r_{i,p(m_i)}. \tag{10}$$

We can now change the optimization problem of (7) to

$$\min \sum_{i=1}^{T} J_i = \sum_{i=1}^{T} \min J_i, \tag{11}$$

where we have used the fact that the cost functions for each coding unit can be optimized independently [20]. Graphically, the above method translates to finding a line of slope λ that is tangent to the convex hull of the RD point $(r_{i,p(m)}, d_{i,p(m)})$ for each coding unit. Analysis is performed by calculating the sum of the rates $r_{i,p(m)}$ from the operating points for all coding units $i = 1, \ldots, T$. If the sum is greater than the rate budget, then λ must be altered to yield a smaller rate summation. Therefore, the reduction in distortion from using an extra bit of channel code for one coding unit is equal to the increase of distortion seen at another [20]. This analysis is performed recursively until the optimal balance is found.

3.2. Dependent coding units

If coding units are dependent on one another, errors from one decoded block of data will transfer to future decoded coding units. The total distortion is then given by

$$D = \sum_{i=1}^{T} d_{i,p(m_i)|p(m_{i-1}),\ldots,p(m_1)}, \tag{12}$$

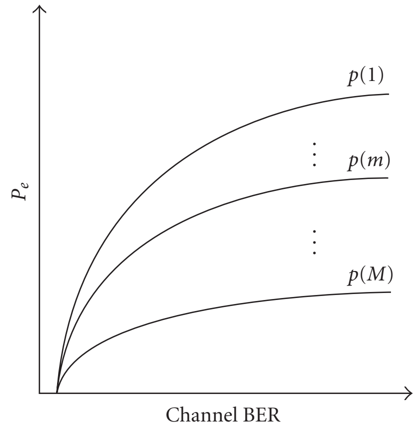


Figure 5: Channel characteristic plot.

where $d_{i,p(m_i)|p(m_{i-1}),\ldots,p(m_1)}$ is the distortion of the ith coding unit based on the protection used on coding units $1, 2, 3, \ldots, i-1$. The rate constraint of (7) remains the same, however, the Lagrange cost functions must reflect the conditional distortion:

$$\begin{gathered} J_1(\lambda) = d_{1,p(m_1)} + \lambda r_{1,p(m_1)}, \\ J_2(\lambda) = d_{2,p(m_2)|p(m_1)} + \lambda r_{2,p(m_2)} \\ \vdots \\ J_T(\lambda) = d_{T,p(m_T)|p(m_{T-1}),\ldots,p(m_1)} + \lambda r_{T,p(m_T)}. \end{gathered} \tag{13}$$

From (12), it is clear that the distortion at any coding unit i depends on the amount of protection used for previous coding units $i = 1, \ldots, T$. Obviously, finding all possible combinations of distortions for all coding units is too complex. To alleviate this problem, we used the universal rate-distortion characteristic (URDC) of [19, 21].

Given a multiple rate channel coder and information of the channel, we can calculate how the channel code will perform in a particular channel environment. Assume a set of puncture matrices $p = \{p(1), \ldots, p(M)\}$ which create a set of code rates $r_c = \{r_{c,p(1)}, \ldots, r_{c,P(M)}\}$. If we know the condition of a channel in terms of its BER, we can perturb a string of data protected with each aforementioned puncture matrix. We then channel decode the perturbed data and calculate the effective probability of bit error (P_e) for each amount of protection. This gives us an idea of how the channel codes will perform in this particular noisy environment. If the channel condition is known to be in a specific range, a set of P_e can be calculated for each BER and the results can be stored in a lookup table called a channel characteristic plot (see Figure 5). These calculations can be performed offline to reduce the amount of computational complexity at run time.

Since we are considering a constant source rate encoder, the distortion for any coding unit i is a function of P_e after channel decoding. A family of curves describing the distortion for a particular coding unit d_i versus the inverse P_e's is called the URDC curve [19]. It is extremely difficult to analytically obtain the URDCs due to application of variable length codes. Therefore, URDCs are obtained using simulations. These curves are generated by perturbing a coding unit i with bit error P_e (found using the current state of the channel and the channel characteristic plot) while coding units $1, 2, \ldots, i-1, i+1, \ldots, T$ are not corrupted. After the coding units are subjected to the appropriate bit errors, the resultant bitstreams are decoded and their reconstructed distortions are measured (this method is detailed in Figure 6). This process is ensemble averaged over a number of average iterations (e.g., 25 times). More details on URDC can be found in [19, 21] and a pictorial version is shown in Figure 6.

Considering the protections given to previous coding units and the individual contribution to distortion from coding unit i found using the URDC and channel characteristic plot, we now have a set of distortions and rates necessary to create the dependent RD map for coding unit i. Once this is done for each coding unit, the Lagrange cost functions are built and the optimization is performed as in Section 3.1.

3.3. Multiple time-varying channels

Assume now that L channels are available for transmission of the string of T coding units. It is further postulated that no single channel can handle all T coding units. The problem then becomes how to allocate the coding units across the channels of varying condition such that the reconstructed frames have minimum distortion and all rate limitations of the channels are respected. The order of the channels in terms of quality is assumed to be unimportant. This is a fair assumption since the decoder is uninterested in the avenue through which the data was sent.

The transmission delay in the multiple channel case is as follows. Let r_i denote the rate of the ith coding unit after error correction code. Then, in order to transmit T coding units, the set $\{r_1, r_2, \ldots, r_T\}$ is partitioned into L sets $C_1, C_2, \ldots, C_L$, where $C_j = \{r_i \mid \text{coding unit } i \text{ is transmitted over channel } j\}$. Then the transmission delay will be equal to

$$\tau = \max_j \left(\frac{\sum_{l \in C_j} r_l}{B_j} \right), \tag{14}$$

where B_j is the bandwidth of the jth channel (in bps).

The method for RD optimization across multiple-channels is much the same as optimizing one channel. In this case, however, an initial λ is selected so as to return a large rate summation. The coding units are now to be partitioned among the channels. For this, a greedy algorithm is employed. Coding units are allocated to the first available channel until the rate budget can no longer accommodate another coding unit. This channel is then considered closed and coding units are scheduled to be transmitted through the next available channel. This process continues until either the last coding unit is scheduled, or the rate constraints of all channels are exceeded and coding units remain. In the case of the latter, λ must be adjusted to allow a lower rate summation.

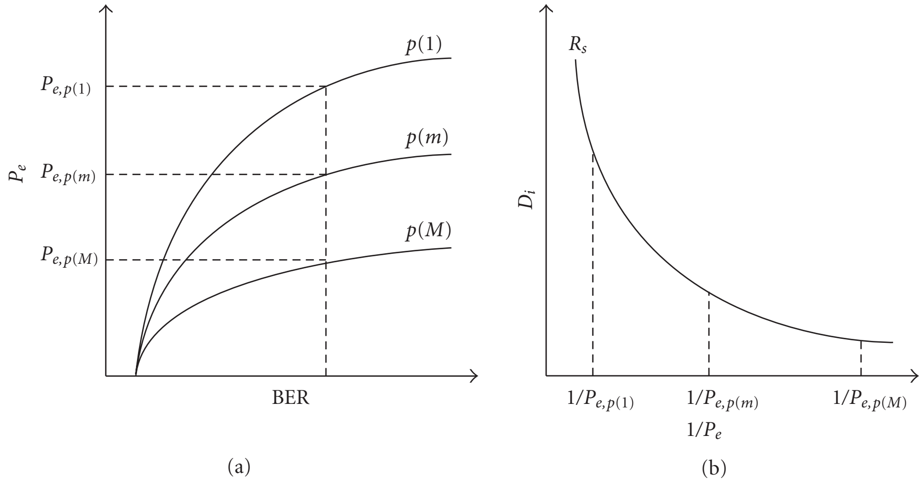


Figure 6: Method for finding distortion of dependent coding units. Figure (a) represents the channel characteristic plot and (b) represents the universal rate-distortion characteristic.

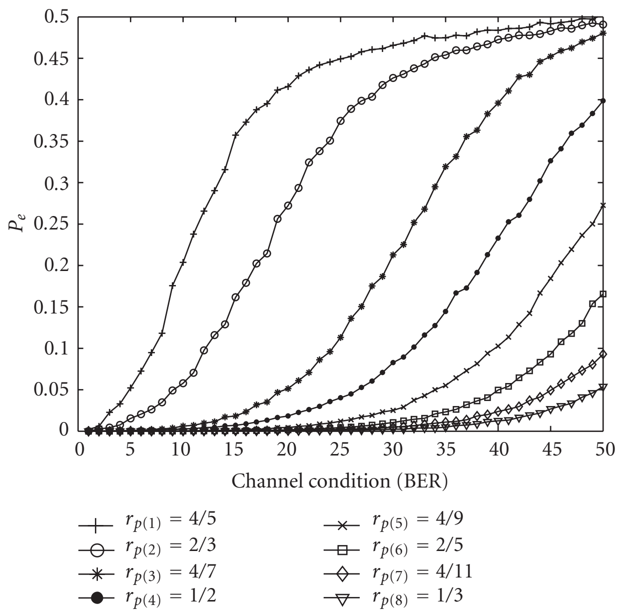


Figure 7: Channel characteristic plot for AR(1) channel with $a = 0.5$.

4. EXPERIMENTAL RESULTS

Frames from the sequences Mobile and Paris (each of size CIF or 352×288 pixels) were used to test the proposed method. Frames 100, 101, and 102 from each sequence were encoded using David Taubman's JPEG2000 encoder version 2.2 at 0.5 bit per pixel (bpp). Using tiles as coding units, the frames were encoded into 30 tiles. Conversely, when packets were used as coding units, 17 packets were generated.

4.1. Protection scheme

For the purpose of analysis, we compare our adaptive protection (AP) approach with a basic protection (BP) scheme. In the AP scheme, the amount of error correction code added to a coding unit is reoptimized at each new time interval based on the condition of the channel(s). The BP scheme, however, creates a set of RD curves with only three operating points.

(1) The coding unit is dropped.
(2) The coding unit is transmitted unprotected.
(3) The coding unit is protected with a predefined amount of error correction code.

This family of RD curves is then searched exhaustively to find the optimal solution for basic protection. This BP scheme was used as we were unable to find previous works with which to compare the time-varying channel model employed for frames.

The RCPC encoder had an unpunctured rate of 1/3, a constraint length of $k = 5$, and a generator matrix $\mathbf{G} = [23\ 35\ 57]^T$ represented in octal. Using puncture matrices found in [16], channel coding rates of $r_c = \{4/5, 2/3, 4/7, 1/2, 4/9, 2/5, 4/11, 1/3\}$ were available. The constant protection of the BP scheme had a rate equal to $r_{c,\text{BP}} = 4/9$.

As indicated in Section 2.3, the time-varying nature of the channels was modeled using a first-order AR process. The value of the parameter in (4) was $a = 0.5$. The nature of the channel by way of the channel characteristic plot is seen in Figure 7.

4.2. Results for tiles

The bitstreams were divided into 30 tiles. Based on the size of the frames (352×288 pixels), there are 4 sizes of tiles. The tile in the bottom-right corner of the frame will be 46×52 pixels. Secondly, the tiles along the right edge of the frame (excluding the aforementioned corner) are 60×52 pixels. The tiles along the bottom edge of the frame (excluding the lower-right corner) are of size 46×60 pixels. Finally, the remaining tiles are 60×60 pixels. Tables 1, 2, and 3 show the

Table 1: PSNR (dB) values for tiles over a single time-varying channel of indicated bandwidth.

Sequence	τ (s)	B (bps)				Frame number
		25 000		30 000		
		AP	BP	AP	BP	
Mobile	10	21.880	20.856	21.889	20.921	100
		21.890	21.209	21.892	21.240	101
		21.904	21.212	21.899	20.849	102
	6	21.839	20.771	21.887	20.774	100
		21.850	20.952	21.874	20.508	101
		21.897	21.099	21.889	20.669	102
Paris	10	25.790	23.970	25.789	23.793	100
		25.811	24.563	25.819	24.409	101
		25.804	24.470	25.801	24.151	102
	6	25.658	23.499	25.788	23.212	100
		25.766	23.981	25.770	23.831	101
		25.744	24.318	25.676	23.706	102

Table 2: PSNR (dB) values for tiles over two time-varying channels of indicated bandwidth.

Sequence	τ (s)	B (bps)				Frame number
		10 000, 20 000		20 000, 18 000		
		AP	BP	AP	BP	
Mobile	10	21.870	20.321	21.887	20.517	100
		21.880	20.190	21.886	20.989	101
		21.890	19.883	21.904	21.115	102
	6	21.855	20.028	21.832	20.084	100
		21.283	19.981	21.836	19.971	101
		21.559	19.501	21.856	20.880	102
Paris	10	25.735	22.555	25.778	23.356	100
		25.783	23.104	25.817	23.338	101
		25.781	23.032	25.764	23.416	102
	6	24.237	21.390	25.731	22.649	100
		24.759	22.660	25.749	21.260	101
		24.708	21.615	25.701	22.582	102

PSNR values for transmission of tiles over 1, 2, and 3 time-varying channels, respectively. In all cases, the PSNR of the AP scheme outperforms that of the BP scheme. In addition, when the amount of transmission delay is reduced, the PSNR values are less than or equal to those of longer delay. This is due to the fact that more coding units were transmitted in a set amount of time thereby spreading out the available resources across more data.

Figures 8, 9, and 10 show frames 100, 101, and 102 (vertical columns) visually detailing the performance of the AP scheme over the BP scheme. For example, frame 100 in Figures 8(a) and 8(d), the ball in the lower middle of the frame is distorted in the BP frame and not as much in the AP frame. It should be noted that since the simulations were run over 25 iterations (on average), the frames chosen are those for which the PSNR of the frame best matches (in a mean-square sense) the mean PSNR over the average number of iterations.

Table 3: PSNR (dB) values for tiles over three time-varying channels of indicated bandwidth.

Sequence	τ (s)	B (bps)				Frame number
		15 000, 9000, 1000		30 000, 25 000, 15 000		
		AP	BP	AP	BP	
Mobile	10	21.897	20.412	21.896	21.234	100
		21.893	20.590	21.893	21.324	101
		21.891	20.805	21.905	21.193	102
	6	21.748	20.260	21.878	20.059	100
		21.683	20.470	21.875	21.011	101
		21.308	20.322	21.901	20.962	102
Paris	10	25.800	23.804	25.800	24.128	100
		25.819	23.690	25.819	24.643	101
		25.738	22.239	25.804	24.496	102
	6	24.952	22.869	25.800	24.035	100
		25.480	23.572	25.783	23.175	101
		25.238	22.211	25.801	24.032	102

Table 4: PSNR (dB) values for packets over a single time-varying channel of indicated bandwidth.

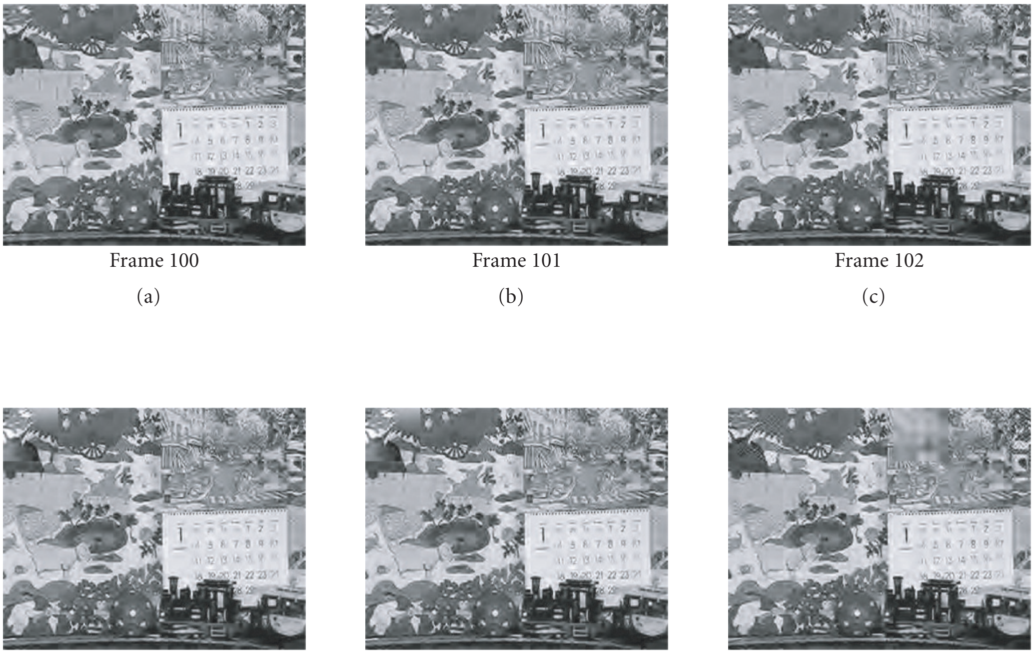
Sequence	τ (s)	B (bps)				Frame number
		45 000		50 000		
		AP	BP	AP	BP	
Mobile	5.67	16.160	12.830	18.840	14.678	100
		17.383	13.043	18.130	13.663	101
		15.831	12.579	17.654	14.461	102
	3.4	11.141	10.679	12.706	11.929	100
		11.443	10.475	12.846	11.963	101
		11.693	10.586	11.984	11.604	102
Paris	5.67	21.887	17.270	24.141	23.511	100
		22.325	17.111	22.697	22.289	101
		20.309	17.409	24.518	24.486	102
	3.4	14.180	13.690	17.732	13.704	100
		14.330	13.970	17.063	15.330	101
		13.998	13.505	15.769	13.851	102

4.3. Results for packets

Much like the results for tiles, Tables 4, 5, and 6 show the PSNR values for transmitting packets over one, two, and three time-varying channels, respectively. Again, in all cases the AP scheme outperforms the BP scheme and as transmission delay changes from 5.67 seconds down to 3.4 seconds, the PSNR values decrease accordingly. Figures 11, 12 and 13 show frames 100, 101 and 102 from video sequences Paris and Mobile illustrating the performance of the AP scheme over the BP scheme for the case of packets.

4.4. Time adaptivity

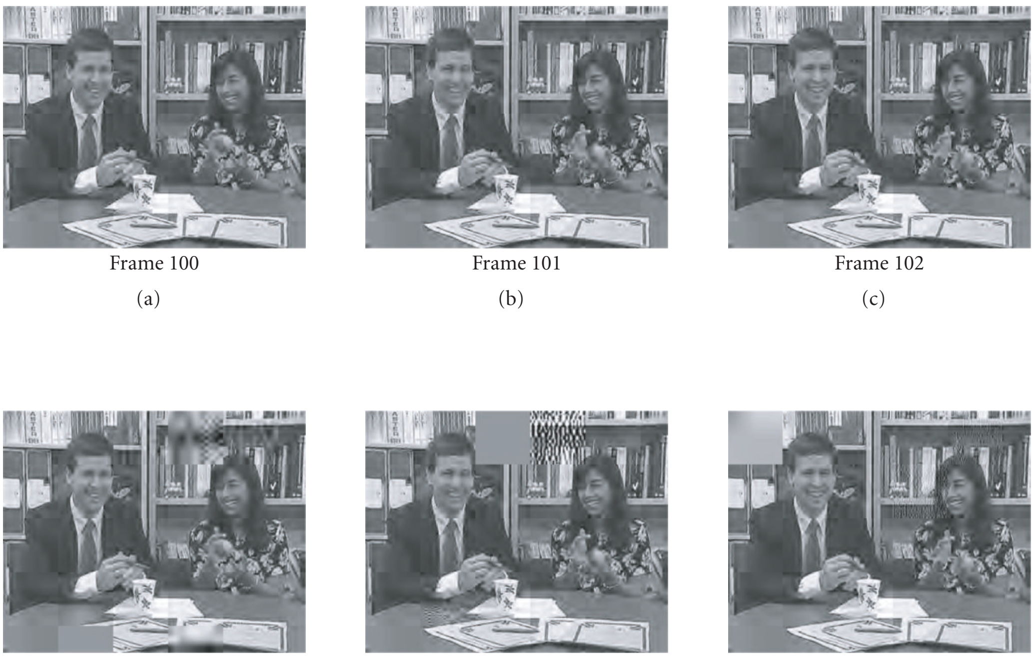
Figures 14, 15, and 16 show time adaptivity for data transmitted across one, two and three channels, respectively. In these figures graphs (a) show the selected puncture matrices,



Frame 100 (a) | Frame 101 (b) | Frame 102 (c)

Frame 100 (d) | Frame 101 (e) | Frame 102 (f)

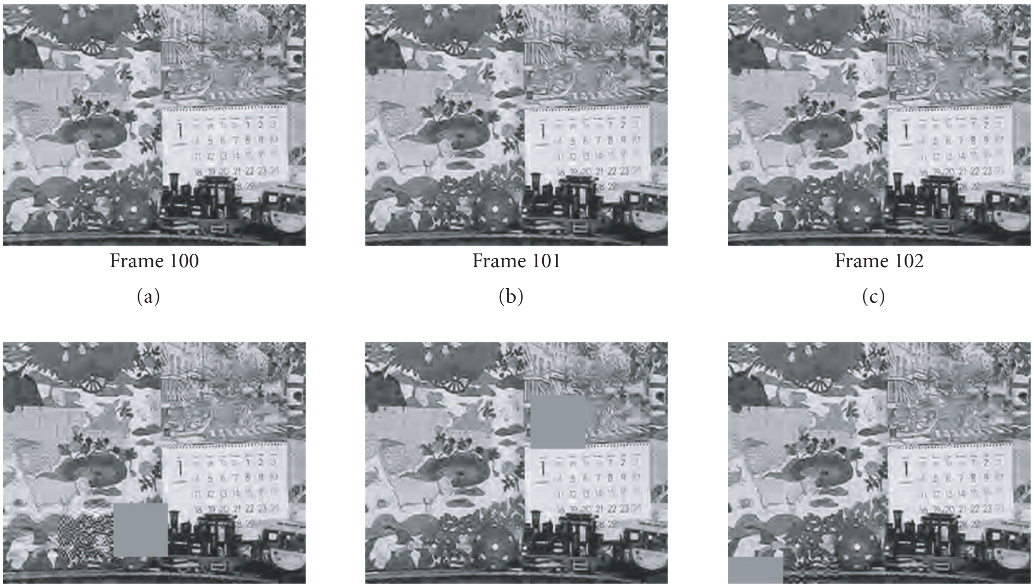
Figure 8: Decoded frames of Mobile sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with a bandwidth of 25 000 bps ($\tau \sim 10$ s).



Frame 100 (a) | Frame 101 (b) | Frame 102 (c)

Frame 100 (d) | Frame 101 (e) | Frame 102 (f)

Figure 9: Decoded frames of Paris sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with bandwidths of 20 000 and 18 000 bps ($\tau \sim 6$ s).



Frame 100 (a) | Frame 101 (b) | Frame 102 (c)

Frame 100 (d) | Frame 101 (e) | Frame 102 (f)

Figure 10: Decoded frames of Mobile sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with bandwidths of 30 000, 25 000, and 15 000 bps ($\tau \sim 10$ s).

Table 5: PSNR (dB) values for packets over two time-varying channels of indicated bandwidth.

Sequence	τ (s)	B (bps) 42 000, 45 000 AP	42 000, 45 000 BP	57 000, 60 000 AP	57 000, 60 000 BP	Frame number
Mobile	5.67	22.237	20.498	22.824	20.560	100
		21.491	18.951	22.864	19.724	101
		22.291	19.234	23.007	20.389	102
	3.4	21.105	19.120	22.537	18.248	100
		20.587	18.197	22.592	19.610	101
		20.265	18.566	22.999	19.407	102
Paris	5.67	27.288	22.445	27.511	23.448	100
		27.559	25.070	27.215	23.343	101
		27.437	21.998	27.506	21.632	102
	3.4	26.517	20.909	27.183	21.358	100
		26.750	20.050	27.007	22.011	101
		26.341	20.908	27.371	20.432	102

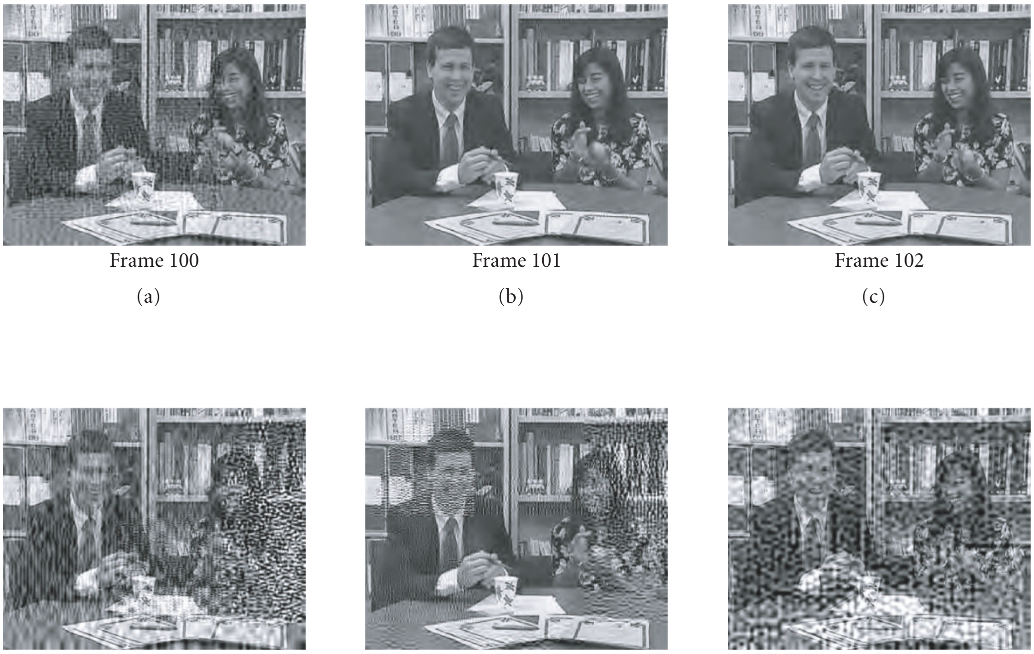
Table 6: PSNR (dB) values for packets over three time-varying channels of indicated bandwidth.

Sequence	τ (s)	B (bps) 35 000, 30 000, 32 000 AP	35 000, 30 000, 32 000 BP	50 000, 40 000, 35 000 AP	50 000, 40 000, 35 000 BP	Frame number
Mobile	5.67	20.684	20.093	22.725	19.155	100
		20.442	20.290	23.021	21.308	101
		20.971	20.790	22.967	20.380	102
	3.4	19.529	18.683	22.374	18.321	100
		20.097	18.796	22.703	19.591	101
		20.812	18.797	22.485	18.891	102
Paris	5.67	27.455	23.945	27.725	22.429	100
		27.370	23.895	27.605	24.277	101
		27.403	22.668	27.224	23.896	102
	3.4	26.626	22.198	27.169	21.300	100
		26.799	19.872	27.268	20.952	101
		27.333	22.213	26.983	21.794	102

whereas graphs (b) show the BER of the channels. Figure 14 is an example of an average iteration for frame 100 of the Mobile sequence encoded into tiles and transmitted across a single channel with a bandwidth of 25 000 bps. Figure 15 shows the timing situation for frame 100 of the Mobile sequence transmitted across two channels with bandwidths of 42 000 and 45000 bps. Finally, Figure 16 shows frame 100 of the Mobile sequence transmitted across channels of 35 000, 30 000, and 32 000 bps. It can be seen from these figures that the levels of protection used change in the same manner as the condition of the channel. As the condition of the channel worsens (higher BER), more protection is required to protect the data.

5. CONCLUSIONS

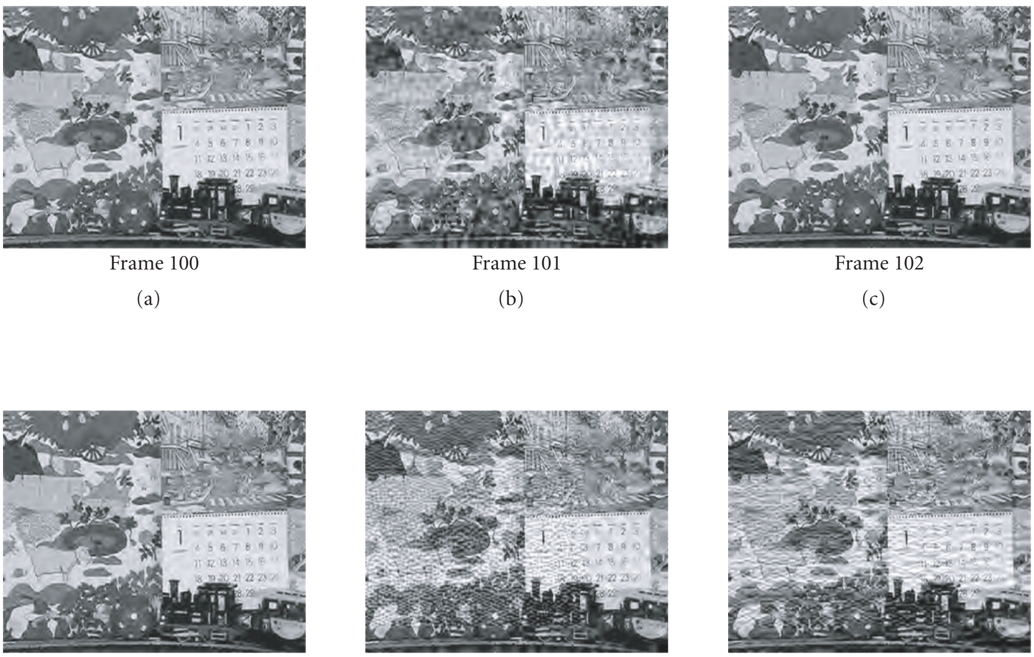
In this paper, we have examined the transmission of MJPEG2000, encoded frames of video sequences. We presented a method for adaptively adding error protection by analyzing the condition of the channel and the frame and



Frame 100 (a) | Frame 101 (b) | Frame 102 (c)

Frame 100 (d) | Frame 101 (e) | Frame 102 (f)

Figure 11: Decoded frames of Paris sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with a bandwidth of 45 000 bps ($\tau \sim 5.67$ s).



Frame 100 (a) | Frame 101 (b) | Frame 102 (c)

Frame 100 (d) | Frame 101 (e) | Frame 102 (f)

Figure 12: Decoded frames of Mobile sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with bandwidths of 42 000 and 45 000 bps ($\tau \sim 3.4$ s).


Frame 100
(a)


Frame 101
(b)

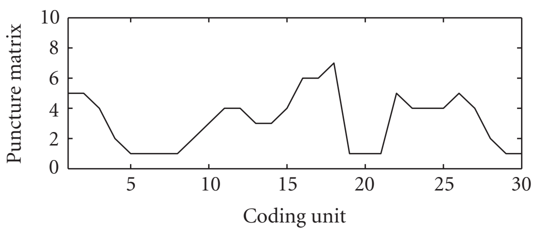

Frame 102
(c)


Frame 100
(d)

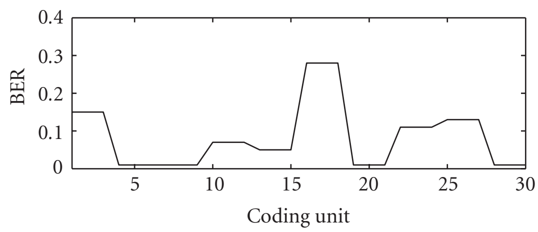

Frame 101
(e)


Frame 102
(f)

FIGURE 13: Decoded frames of Paris sequence using (a), (b), (c) adaptive protection and (d), (e), (f) basic protection, all transmitted over a single channel with bandwidths of 50 000, 40 000, and 35 000 bps ($\tau \sim 5.67$ s).

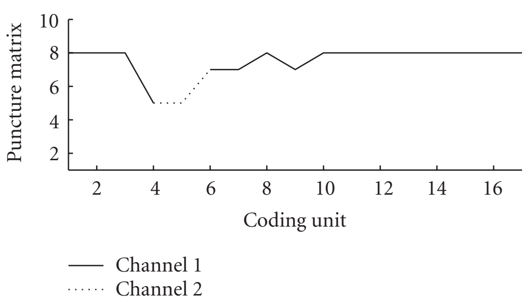


(a)

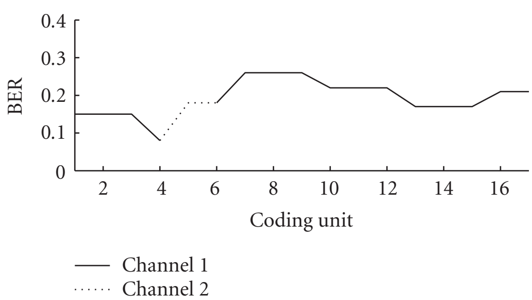


(b)

FIGURE 14: An example of time adaptivity of tiles over a single time-varying channel.



(a)



(b)

FIGURE 15: An example of time adaptivity of packets over two time-varying channels.

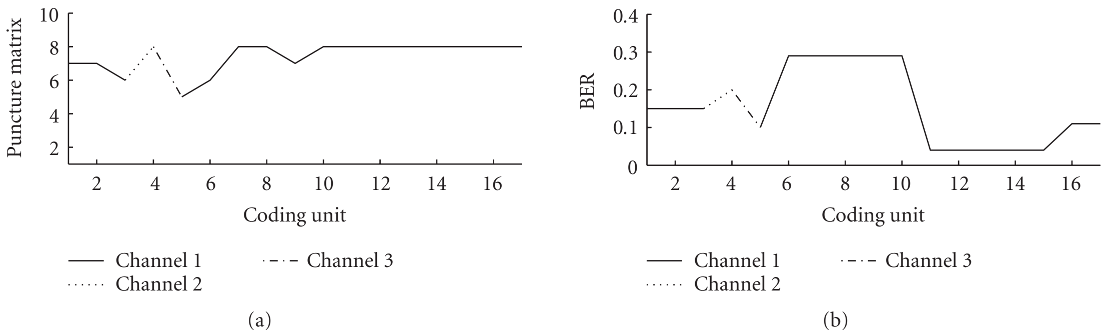


Figure 16: An example of time adaptivity of packets over three time-varying channels.

using a rate-flexible channel coder. In addition, we considered bandlimited channels, so there would be a tradeoff between the amount of reconstructed distortion and the allowable bit rate of the system. Frames from video sequences were encoded and transmission was simulated over error-prone channels. We showed that our method outperforms a method where the amount of protection available is constant regardless of the condition of the channel.

Of course, this method may not be advantageous in a real-time environment as each frame is encoded using MJPEG2000, which does not make use of efficient techniques of motion compensation. However, out method could be used effectively in a transmission situation which is not time-critical and where the end result is data storage (e.g.). In addition, the proposed method can be used in applications that require no error propagation when any frame is corrupted.

REFERENCES

[1] M. J. Ruf and J. W. Modestino, "Rate-distortion performance for joint source and channel coding of images," in *Proceedings of IEEE International Conference on Image Processing (ICIP '95)*, vol. 2, pp. 77–80, Washington, DC, USA, October 1995.

[2] T. Chuu, Z. Liu, Z. Xiong, and X. Wu, "Joint UEP and layered source coding with application to transmission of JPEG-2000 coded images," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 3, pp. 2036–2039, San Antonio, Tex, USA, November 2001.

[3] V. Chande and N. Farvardin, "Joint source-channel coding for progressive transmission of embedded source coders," in *Proceedings of Data Compression Conference (DCC '99)*, pp. 52–61, Snowbird, Utah, USA, March 1999.

[4] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," in *Proceedings of Data Compression Conference (DCC '97)*, pp. 72–81, Snowbird, Utah, USA, March 1997.

[5] B. A. Banister, B. Belzer, and T. R. Fischer, "Robust image transmission using JPEG2000 and turbo-codes," *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 117–119, 2002.

[6] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 230–239, 2004.

[7] H. Li and C. W. Chen, "Robust image transmission with bidirectional synchronization and hierarchical error correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 11, pp. 1183–1187, 2001.

[8] L. Cao and C. W. Chen, "Multiple hierarchical image transmission over wireless channels," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '99)*, vol. 1, pp. 349–353, New Orleans, La, USA, September 1999.

[9] N. Chaddha and S. Diggavi, "A frame-work for joint source-channel coding of images over time-varying wireless channels," in *Proceedings of IEEE International Conference on Image Processing (ICIP '96)*, vol. 2, pp. 89–92, Lausanne, Switzerland, September 1996.

[10] V. S. Sanchez and M. K. Mandal, "Robust transmission of JPEG2000 images over noisy channels," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 451–456, 2002.

[11] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the Internet using unequal error protection," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 121–131, 2003.

[12] T. Fukuhara, K. Katoh, S. Kimura, K. Hosaka, and A. Leung, "Motion-JPEG2000 standardization and target market," in *Proceedings of IEEE International Conference on Image Processing (ICIP '00)*, vol. 2, pp. 57–60, Vancouver, BC, Canada, September 2000.

[13] F. Dufaux and T. Ebrahimi, "Motion JPEG2000 for wireless applications," in *Proceedings of 1st International JPEG2000 Workshop*, pp. 2036–2039, Lugano, Switzerland, July 2003.

[14] S. Fossel, G. Fottinger, and J. Mohr, "Motion JPEG2000 for high quality video systems," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 787–791, 2003.

[15] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic, Boston, Mass, USA, 2002.

[16] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.

[17] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Transactions on Broadcasting*, vol. 48, no. 3, pp. 223–229, 2002.

[18] T. P.-C. Chen and T. Chen, "Adaptive joint source-channel coding using rate shaping," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, vol. 2, pp. 1985–1988, Orlando, Fla, USA, May 2002.

[19] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated DCT-based SNR scalable video," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1043–1052, 2002.

[20] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

[21] M. Bystrom and J. W. Modestino, “Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 880–890, 2000.

Scott Bezan received his Bachelor of Engineering degree in electrical engineering from McMaster University in Hamilton, Ontario, Canada, in 2002, and his Master of Applied Science degree in electrical engineering also from McMaster in 2004. During his graduate education, Scott’s research focused on image and video processing and error correction coding involved in robust multimedia transmission.

Shahram Shirani received his Bachelor of Engineering degree in electrical engineering from Isfahan University of Technology, Iran, and his Master of Science (with honor) degree in biomedical engineering from Amirkabir University of Technology, Iran, and his Ph.D. degree in electrical engineering from University of British Columbia, Canada, in 1989, 1994, and 2000, respectively. Since July 2000, he has been with the Department of Electrical and Computer Engineering, McMaster University, as an Assistant Professor. His research interests are mainly focused on image and video processing, multimedia compression and communications, medical image processing and hardware architectures for image and video processing. He has published more than 60 journal and conference papers. He is a Senior Member of IEEE, a Member of Technical Program Committee of ICIP, a Member of Technical Program Committee of ICASSP, and a Professional Engineer.