Joint Source-Channel Decoding for MPEG-4 Video Transmission Over Wireless Channels

Qingyu Chen and K.P. Subbalakshmi

***Abstract*— This paper investigates the use of joint source-channel coding for reliable transmission of MPEG-4 video over wireless channels. We present a maximum *a posteriori* probability (MAP) based, joint source-channel decoder for variable length coded intra and inter macroblocks in a MPEG-4 encoded video stream. We then provide a comprehensive evaluation of this paradigm as an alternative method for achieving error resilience. In particular our method is tested in conjunction with the MPEG-4 error resilience tools such as reversible variable length codes (RVLC), resynchronization markers and data partitioning, with and without FECs and finally without the use of RVLCs (only variable length codes). In addition, the JSCD algorithm is tested with and without the use of FECs (Reed-Solomon codes). These tests show that the best performance at high channel bit error rate is obtained when all error resilience tools are co-opted. The maximum improvement due to the proposed decoder when used without FEC is about 7 dB for RVLC and 8 dB for VLCs. These numbers are about 2.2dB and 7.5dB respectively, when the MAP decoder is used in conjunction with the Reed-Solomon codes.**

***Index Terms*— MPEG-4 video over wireless channels, joint source-channel coding, error-resilient multimedia communication, MAP decoding, RVLC decoding**

I. INTRODUCTION

MPEG-4 [1], [2] is a state-of-the-art audiovisual coding standard that adopts object based coding techniques to achieve efficient representation of video content. MPEG-4 video standard uses *motion estimation* (ME) and the *discrete cosine transform* (DCT) to reduce the redundancy of the video source. While this hybrid coding approach (block based DCT + ME) can achieve significant compression efficiency, it is very sensitive to channel noise. For example, because of the use of spatio-temporal prediction, a single erroneously recovered sample can lead to errors in the following samples in the same and successive frames. Moreover, because of the use of *variable length codes* (VLC), a single bit error can cause the decoder to parse the codeword boundaries incorrectly leading to a loss in synchronization. Such loss of synchronization adversely affects the reconstructed video quality.

Although more traditional methods like *automatic repeat request* (ARQ) and *forward error correction* (FEC) reduce the effect of channel errors, these solutions could prove to be expensive in band-limited communications of delay sensitive signals like video. On the one hand, ARQ based systems may incur significant delays which could also potentially lead to network congestion; while FEC are usually expensive in terms of bandwidth and often fine-tuned to a particular error rate, with catastrophic deterioration in performance when the actual conditions differ. It is thus of interest to search for alternative ways to decrease the error sensitivity of variable length encoded video.

This work was funded in part by the New Jersey Center for Wireless Telecommunications and the Stevens Center for Wireless Network Security.

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ, 07030 (email: qchen1@stevens-tech.edu, ksubbala@stevens-tech.edu)

The MPEG-4 standard itself provides a set of tools to deal with the error resilience problem which include: *reversible variable length codes* (RVLC) [3], [4], *resynchronization markers* (RM) and *data partitioning* (DP) [5]. Although these error resilience tools can protect the video stream from some channel errors and can recover some corrupted video frames; they can only restrict the propagation of errors to a small region of time or space and not eliminate the errors completely. Therefore, when the channel bit error rate (BER) is very high, the decoded MPEG-4 video sequence will still have many noisy blocks, which could potentially degrade the visual quality significantly.

In this paper we present a maximum *a posteriori* probability (MAP) based, *joint source-channel decoder* (JSCD) for the variable length encoded intra and inter macroblocks in a MPEG-4 encoded video stream. We then test the use of our decoder both, in conjunction with and in lieu of, some of the error resilience features of the MPEG-4 standard. The aim of this paper is threefold:

1) to develop a joint source-channel decoder for VLCs used in the MPEG-4 video streams.
2) to measure the performance of this decoder both, *in conjunction with* and *in lieu of* some of the error resilience features provided in the MPEG-4 standard.
3) to measure the performance of this decoder with and without FEC.

This paper is organized as follows: we discuss joint source-channel coding in section II and MPEG-4 error resilience tools in section III. In section IV, we describe the model for wireless channel and the model for input source. The MAP problem is formulated and a brief description of the algorithm is given in V. Extensive experimental results and comparisons are described in section VI.

II. JOINT SOURCE-CHANNEL CODING

Joint source-channel coding (JSCC) has been receiving significant attention lately as a viable solution for achieving reliable communication of signals across noisy channels. The rationale behind using such techniques is the observation that Shannon's source-channel separation theorem [6] does not usually hold under delay and complexity constraints or for all channels [7]. JSCC tries to design the source coder and channel coder in some joint way, which can provide better error protection and bandwidth utilization. JSCC schemes can be broadly classified into three different categories, *joint source-channel encoding* (JSCE) [8]-[10], *joint source-channel decoding* (JSCD) [11]-[29], [31]-[36], [38]-[39] and rate allocation strategies [40]-[42]. As the names suggest, these deal with the joint design of encoders, decoders and the rate allocation between the channel and source codes respectively. One early work in this class is by Dunham and Gray [8], where they demonstrate the existence of a joint source-channel system for special source and channel pair, by showing that a communication system using trellis encoding of a stationary, ergodic source over a discrete memoryless noisy channel can perform arbitrarily close to the source distortion-rate function evaluated at the channel capacity. Other works include an index assignment algorithm proposed for the optimal vector quantizer on a noisy channel [9] and the design of quantizers for memoryless and Gauss-Markov sources over binary Markov channels [10].

Work on rate allocation between the channel and source codes includes the optimal allocation algorithm between a vector quantizer and a channel coder for transmission over a *binary symmetric channel* (BSC) [40], the optimal source-channel rate allocation to transmit H.263 coded video with trellis-coded modulation over a slow fading Rician channel [41] and an algorithm to distribute the available source and channel coding bits among the subbands of scalable video transmitted over BSC to minimize the expected distortion [42].

JSCD schemes can be further classified into *constrained* JSCDs and *integrated* JSCDs. Constrained JSCDs are typically source decoders that are built using prior knowledge of channel characteristics while integrated JSCDs combine the source and channel decoder into one unit. One example of constrained JSCD for fixed length encoded sources is the work of Sayood and Borkenhagen [11], who investigated the use of residual redundancy left in the source after coding it with a *differential pulse code modulation* (DPCM) source coder in providing error protection over a BSC. This was then extended to include conventional source coder/convolutional coder combinations [12]. Other work in this class includes the design of a MAP detector for fixed length encoded binary Markov source over a BSC [13] and a MAP decoder for hidden Markov source [14].

Designing JSCDs for VLC encoded sources is in general more complex because the parsing of the bit stream is not known *a priori*. Some examples done in this direction for the fully interleaved, memoryless channel include the design of source-symbol synchronized Viterbi decoders for convolutional and VLC coded source [15], the exact and approximate MAP decoder for variable length encoded data transmitted over BSC [16] and soft-in soft-out dynamic decoding for VLCs [17]. Some researchers have proposed iterative JSCDs. This method serially concatenates VLC decoder and soft convolutional decoder and do decoding process iteratively. Examples include the work on symbol by symbol MAP decoding for VLC coded memoryless source by Bauer *et al* [18] and its extension for RVLCs [19]. Hedayat *et al* compared an iterative JSCC with a separable system under the same overall rate and computational complexity [20]. Kliewer *et al* designed an iterative decoder consisting of an *a posteriori* probability based channel decoder and the proposed *a posteriori* probability based VLC source decoder [21]. An example on JSCD for turbo codes is by Guivarch *et al*, who have proposed methods of incorporating *a priori* information on Huffman codes into the turbo decoder [22]. We recently developed a novel state-space structure and designed a MAP decoder for VLC encoded memoryless sources [23] and Markov sources [24], [25] transmitted over BSCs for the case where the number of transmitted words is not known to the decoder.

Design of integrated JSCDs has also been gaining significant attention lately. One example for transmission over memoryless channels includes the work of Murad and Fuja [26], who developed an integrated source-channel decoder by combining graph representations of the Markov source model, the Huffman source code decoder, and the convolutional channel decoder. This algorithm was then used to exploit the residual redundancy in *motion vectors* (MVs) [27]. Lakovic and Villasenor proposed a modified Viterbi decoding trellis [28], incorporating information about the structure of the Huffman source code. This decoder differs from [26] in that it is designed for memoryless sources. The authors then incorporated it into a turbo decoder in [29], giving better performance.

Wireless channels are characterized by bursty errors due to fading effects. Alajaji, *et al*. developed a Markov model for this channel which was named *additive Markov channel* (AMC) model [30]. A MAP detector for binary asymmetric Markov sources (not VLC) was then developed for wireless channels, modelled using the AMC model [31] and this was then applied to image transmission in [32]. A MAP based JSCD for variable-length encoded Markov sources was developed using this channel model [33], [34].

Some of the more recent work that directly relates to joint source-channel decoding for MPEG-4 coded video streams include that of Kopansky, Bystrom and Kaiser [35], [36]. The authors presented a method for decoding start codes and overhead information in a MPEG-4 bitstream using sequential decoding with soft channel values [35]. The overhead information is viewed as variable-length codes and a tree whose states represent the current count of received bits is constructed. Fano's algorithm for sequential decoding [37] is used with both soft and hard channel values in this work. They also presented a performance evaluation of MAP and sequential decoding using soft channel values in [36], where their results indicate that for codes with large Hamming distances between codewords, transmission with no channel coding may be feasible for good channels. In addition, a method of *unequal error protection* (UEP) for a MPEG-4 bitstream using both the MAP and sequential source decoders was also presented. In this method, RVLCs and overhead information are transmitted without any FEC, but the remaining bits of MPEG-4 stream are coded with a convolutional code. Then RVLCs are decoded with the MAP decoder and the overhead information is decoded with a sequential decoder. Other bits are decoded with a soft decision convolutional decoder. The results of their scheme shows comparable performance with the traditional method.

Our approach is different from [36] in that, we apply hard decision MAP decoding for VLCs in inter and intra macroblocks of the MPEG-4 bit stream. Also we study the problem of error resilient transmission of MPEG-4 video in the *wireless* environ-

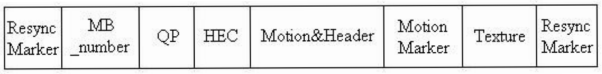

Fig. 1. Bit structure of resynchronization and data partitioning

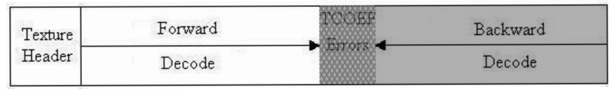

Fig. 2. Decoding RVLC in both directions reduces error region

ment. Our algorithm does not use the knowledge of the length of the bit stream (either in symbols or bits). Simulation results show that the proposed decoder provides significant subjective and objective gains (as measured in terms of the *peak signal-to-noise ratio* (PSNR)) over standard error resilience schemes. Some preliminary results were published in [38], [39].

III. MPEG-4 Video Coding and Error Resilience Tools

A set of algorithms and tools were standardized in MPEG-4 for coding and flexible representation of audio-visual data to meet the challenges of emerging and future multimedia applications and applications environments [1]. Main features of the MPEG-4 standard include object-based coding, object-based scalability, improved compression, error resilience and hybrid synthetic/natural video coding. MPEG-4 relies on an efficient content-based representation of the video data to provide the above features. A key concept in MPEG-4 is that of *video object* (VO). Each VO is characterized by intrinsic properties such as shape, texture and motion. MPEG-4 considers a scene to be composed of several VOs.

Like MPEG-1 and MPEG-2, MPEG-4 adopts DCT and quantization on a block-by-block basis to reduce the spatial redundancy within one frame [1]. In order to reduce the temporal dependence between adjacent frames, *macroblock* (MB) based motion estimation and compensation are also deployed. MBs that are coded using motion compensation are named inter MBs and MBs coded without motion compensation are named intra MBs. Both of them can be either encoded with VLCs or RVLCs depending on whether this error resilience option (RVLC) is specified or not.

A number of tools have been incorporated into the MPEG-4 video encoder to make it more error-resilient. These tools can be categorized as *resynchronization*, *data recovery*, *error concealment*, and *error refreshment* [1], [2]. Resynchronization tools attempt to keep the synchronization between the decoder and the encoder after errors have been detected. One such tool is resynchronization marker, which is inserted into the bitstream periodically. Header bits carrying information about the spatial location of the following MBs are added immediately after these resynchronization markers. When synchronization is lost, these header bits can help the decoder regain synchronization. Data partition can be utilized both for data recovery and error concealment. Data partitioning involves the separation of the video data within a video packet into two parts: motion vectors (MV) and DCT data. This is done using a unique motion marker. When errors are detected solely in the DCT field of a MB, it is reconstructed using just the corresponding MV, which will result in better quality than simple replacement from the previous frame. An example illustrating the bitstream organization with resynchronization and data partitioning is given in Figure 1. As seen in this figure, a video packet is composed of a resynchronization maker, header and motion information (MV), motion marker and texture information (DCT data). Header information is provided at the start of a video packet. These header information include MB number (MB_number) of the first MB in this packet, quantization parameter (QP) for the first MB and the header extension code (HEC). HEC is essentially a sign bit which indicates whether additional video object plane (VOP) level information like timing information, temporal reference, VOP prediction type and some other information will be available in this header. The header extension feature enables the decoder to correctly utilize data contained in the current packet without reference to other packets.

RVLCs are a class of entropy codes that can be uniquely decoded in both the forward and reverse directions [3], [4] such that, when the decoder detects an error while decoding the bit stream in the forward direction, it proceeds to the end of the stream and decodes the bit stream in the reverse direction until it encounters an error. Although RVLCs could potentially reduce the compression efficiency achievable by VLCs, it provides substantial improvement in error resilience. Therefore, the decoder can recover some of the data that would otherwise have been discarded. Figure 2 shows how the RVLC works. Here, the decoder starts to decode the information from left to right (marked by an arrow in the figure). When an error occurs within the DCT coefficients region, the decoder stops decoding in the forward direction and jumps to the next resynchronization marker and proceeds to decode in the reverse direction. The erroneous region is now limited to the cross hatched area instead of being spread out as in the case when only variable length codes are used (indicated by the shaded area).

In this paper, resynchronization markers, data partitioning and RVLC are used when we encode the MPEG-4 video. We propose a JSCD algorithm for RVLC coded streams thus using JSCD in conjunction with the MPEG-4 error resilience options. We also compare the JSCD developed for VLCs [33] along with the use of data partitioning and resynchronization markers with the RVLC case - both with and without the MAP decoder. The error refreshment in the MPEG-4 standard is designed to refresh the reference frame when errors occur. In this paper, we don't consider error refreshment techniques, since it does not have a direct bearing on the performance of our algorithm.

IV. Channel and Source Models

A. Channel Model

Wireless channels are characterized by fading which implies that the bit errors occurring in the wireless channels have memory. Alajaji, et al. [30] proposed an *additive Markov channel* (AMC) model for slow fading wireless channels. According to this model, the channel can be described by the equation:

$$Y_i = X_i \oplus Z_i, \qquad \forall i = \{1, 2, ...\} \tag{1}$$

where X_i, Z_i and Y_i are binary and represent the input, the noise and the channel output bit, respectively and $\oplus$ denotes the binary XOR operation. The noise process Z_i is assumed to be a stationary, ergodic, Markov process, independent of the input. For a channel with error probability ϵ and correlation coefficient ρ_c, the transition probabilities $Q(z_n|z_{n-1}) \equiv \Pr\{Z_n = z_n|Z_{n-1} = z_{n-1}\}$ and the marginal probabilities of the noise bits $Q(z_n) \equiv \Pr\{Z_n = z_n\}$ are given by:

$$\begin{bmatrix} Q(0|0) & Q(1|0) \\ Q(0|1) & Q(1|1) \end{bmatrix} = \frac{1}{(1+\delta)} \begin{bmatrix} 1-\epsilon+\delta & \epsilon \\ 1-\epsilon & \epsilon+\delta \end{bmatrix} \quad (2)$$

with $Q(1) = \epsilon = 1 - Q(0)$ and $\delta = \rho_c/(1-\rho_c)$ is a correlation parameter. In this work we use this model for the wireless channel.

B. Source Model

Video compression algorithms try to eliminate the redundancy in the video signal; however, most practical schemes are not completely successful in removing this redundancy due to complexity constraints. This leads to residual redundancy remaining in the source after compression. This redundancy may be attributed to the non-uniform source distribution or to the source memory, or both [31]. Joint source-channel decoders try to make use of this redundancy to reduce the effect of the channel noise in the decoded signal. In order to design a JSCD that is matched to the source, we need to determine if the source has any memory in it. In order to do this we calculate the covariance of the source.

In this work, we have four different sources: inter and intra MBs coded using VLCs or RVLCs. To determine if the sources have memory, we treat the input source as a stochastic process X_n, and compute its covariance using the following equation:

$$C_{n,n+1} = E\{X_n X_{n+1}\} - E\{X_n\}E\{X_{n+1}\} \quad (3)$$

A non-zero covariance would indicate that the source has memory. Table I gives the covariances of the four different sources for 16 frames of the foreman sequence (QCIF resolution), which is our test sequence. This table indicates that the sources are not memoryless and hence we model them as 1-D Markov processes and apply the MAP decoder developed in [33] for entropy coded Markov sources over wireless channels to both sources.

TABLE I
COVARIANCES OF SOURCES

	VLC coded	RVLC coded
Intra MB	-342.1	-626.9
Inter MB	168.8	454.0

V. PROBLEM FORMULATION AND THE ALGORITHM

The problem of decoding a sequence of symbols transmitted over a noisy channel can be cast as a multiple decision problem. Finding the maximum *a posteriori* probability (MAP) sequence then becomes equivalent to minimizing the probability of decision error for the sequence. The MAP decoder searches over all possible transmitted sequences to find the most probable one among them. Considering the computational complexity, MAP problems are usually cast in a dynamic programming format. For variable-length encoded sequences, the partitioning of the bit stream into component codewords is not obvious, making the dynamic programming formulation more challenging. However, once an appropriate state-space is defined this problem can be solved using a trellis based algorithm that prevents the complexity from escalating with time.

Here we briefly overview the maximum *a posteriori* probability (MAP) problem for a 1-D first order, variable-length encoded Markov source transmitted over a wireless channel, modelled as an AMC [33], [34]. Let $\mathbf{C}$ denote the set of all possible B-bit sequences of variable-length codewords of a first order Markov source. The j^{th} such sequence is then denoted by $\mathbf{c}_j^{n(j)} = \{c_{j,i}\}_{i=1}^{n(j)}$, where $c_{j,i}$ is the codeword corresponding to the i^{th} symbol in the transmitted stream and $n(j)$ is the total number of codewords in this sequence. Similarly, $r_{j,i}$ is the i^{th} symbol of the received bit-stream under the same partition as $\mathbf{c}_j^{n(j)}$. The task of the decoder is to take the received bit stream and to search through the members of $\mathbf{C}$ for the most probable transmitted sequence, which we denote by index $\hat{j}$.

Since there are many ways in which to partition the received bit stream to yield different index sequences, the problem is to find the "best" possible index sequence, given the source and channel statistics. We note that the different partitions of the received bit stream will, in general, lead to different number of codewords in the index sequence for a variable-length encoded signal. More formally, the MAP problem for the AMC (MAP-AMC) is to determine the optimal sequence $\mathbf{c}_{\hat{j}}^{n(\hat{j})} = \{c_{j,1}, ..., c_{j,n(\hat{j})}\}$ according to:

$$\mathbf{c}_{\hat{j}}^{n(\hat{j})} = \arg\max_{\mathbf{c}_j^{n(j)}} [\Pr(c_{j,1})\epsilon^{Z_{j,1}}(1-\epsilon)^{(1-Z_{j,1})} \times \prod_{k=2}^{n(j)} \Pr(c_{j,k}|c_{j,(k-1)}) \prod_{i=2}^{B} \Pr(Z_{j,i}|Z_{j,(i-1)})] \quad (4)$$

where $\Pr(c_{j,1})$ is the probability that codeword $c_{j,1}$ was transmitted first, $\Pr(c_{j,k}|c_{j,k-1})$ is the probability that the codeword $c_{j,k}$ was transmitted immediately after the codeword $c_{j,k-1}$ and $Z_{j,i}$ is the i^{th} noise bit under the j^{th} partition, which is determined by the received bit stream, the specific partition and Equation 1. This maximization over j, effectively searches through all possible error sequences and bit stream partitions.

The above maximization can be cast as a dynamic programming problem. A good state-space needs to be defined to suit the variable-length codewords. In [24], authors presented an effective state-space consisting of two classes of states: the complete and the incomplete states to deal with variable length coded transmission over BSCs. This state-space is a generalization of the one developed in [43] for variable dimension vector quantization. It was later shown [34] that the same state-space can be used for the wireless channel case as well, providing that some bit-error probability history is maintained. According to this state-space definition, the decoder is said to be in a complete state if the most recent bit that it received is the last bit of a codeword. If the last received bit does not terminate a codeword, then the decoder is said to be in an incomplete state.

The time evolution of the state-space gives rise to a trellis. Two operations take place at each stage of trellis. One consists of examining the metrics of all the paths, entering each node (state-stage pair) in the trellis and the other consists of looking for a merger of the paths and the actual declaration of decoded codewords (if there is a merge). For complete states, the path update step involves finding the best metric path leading to the state and retaining it; whereas, for the incomplete states, we must retain the metric values of all the paths back to the last complete state. At the end of sequence, the maximum metric path ending in a complete state is the decoded stream. The algorithm can be stated formally as follows:

Initialize:

- Set stage number: $i \leftarrow 1$
- Set all path metrics to zero.
- Set all paths to NULL.

Input: Bit at i^{th} stage.

Update path metrics:

- Complete states: add the smallest incremental metric.
- Incomplete states: retain metric values of all its parent states (parent states are complete states in previous stages that can lead to this incomplete state at this stage).

Update noise-bits:

- Complete states: compare input bits and decoded codewords to get the error-bits information.
- Incomplete states: retain noise-bits information of all its parent states.

Update paths:

- Complete states: concatenate the path segment with the smallest incremental metric.
- Incomplete states: retain links to all parents.

Merge check and output:

- If $i >= l_{max} + 1$
  check for path merges from complete and incomplete states and declare the common ancestor to be part of the decoded stream. l_{max} is the maximum length of codewords.
- If there are no more bits
  STOP.
  Else
  $i \leftarrow i + 1$
  go to **Input**.

The complexity of trellis based algorithm is generally considered to be equal to the number of states in the state-space. In this case, the complexity would be $N + l_{max} - 1$. It is also possible to decrease the complexity of the algorithm to $N + (l_{max} - 1)/g$, when the greatest common divisor of the lengths of the codewords in the codebook, g is greater than one, along the lines of reasoning in [43].

Note that since a RVLC is a special form of VLC, the state-space and the algorithm developed for decoding VLCs is applicable to RVLCs as well.

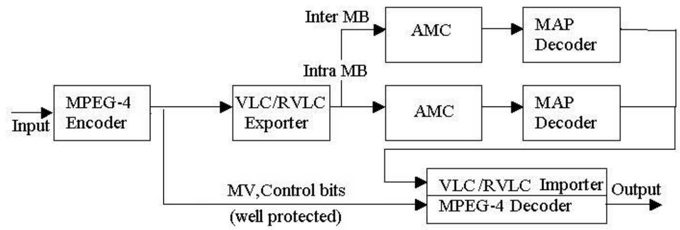

Fig. 3. Experimental Set-up for evaluating the performance of the MAP decoder without channel coding

VI. Experimental Setup and Results

One aim of this paper is to compare the error resilience measures in the MPEG-4 codec with some constrained JSCD paradigms and to evaluate the merits of this approach. We also want to evaluate the performance when the JSCD approach is used in conjunction with channel coding. Hence the experiments described in this section are designed to study the performance of the joint source-channel decoder both separately and in conjunction with some of the MPEG-4 error resilience features as well as traditional error control techniques like the FEC.

A. *JSCD without channel coding*

In this section, we present four experiments that were conducted: encode the MPEG-4 inter and intra coded MBs using VLCs and decode them with: 1) a naïve decoder and 2) the MAP decoder developed for VLCs over AMCs. We then use the RVLC option of the MPEG-4 encoder for both the inter and intra MBs and decode the received bit streams using: 3) the standard MPEG-4 decoder and 4) our decoder for RVLC codes. The experimental set-up is depicted in Figure 3. The bit stream corresponding to inter and intra MB are corrupted separately through eight instances of the wireless channel (simulated using different random seeds). Two error resilience tools: data partitioning and resynchronization markers are deployed in all experiments. We assume that the remaining portions of the MPEG-4 video stream (MV, Control bits, etc) are very well protected using standard forward error correcting schemes. Since the motion vector does not have much memory in them, designing a JSCD for this portion of the stream may not prove to be effective and the control bits are too sensitive to errors. Our test sequence consists of 16 frames of the foreman sequence coded as one object with two I-frames (the 1^{st} and 9^{th}) and fourteen P-frames. We choose frames 284 - 299 from the 400 frames of the entire test sequence because this section exhibits a lot of activity and a scene change. The rates of the encoded sequences for all experiments are kept the same at 0.164 bits per pixel. The average PSNR of the Y component of the compressed stream (with no channel corruption) is 33.57 dB using RVLCs and 34.38 dB using VLC. We use four different channel bit error rates (ϵ) : $10^{-2.5}, 10^{-3}, 10^{-3.5}$ and 10^{-4} to corrupt the encoded stream. The channel correlation coefficient

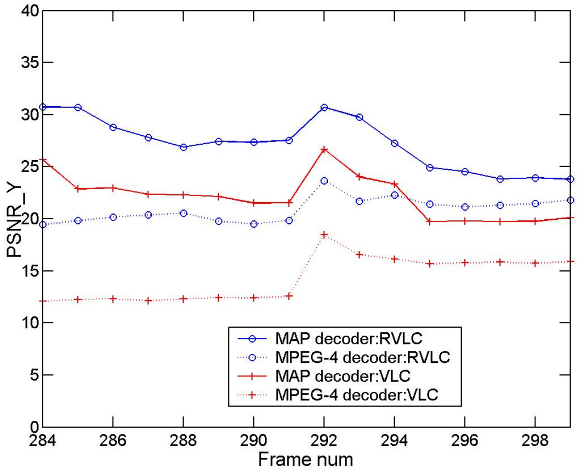

Fig. 4. PSNR of Y component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$

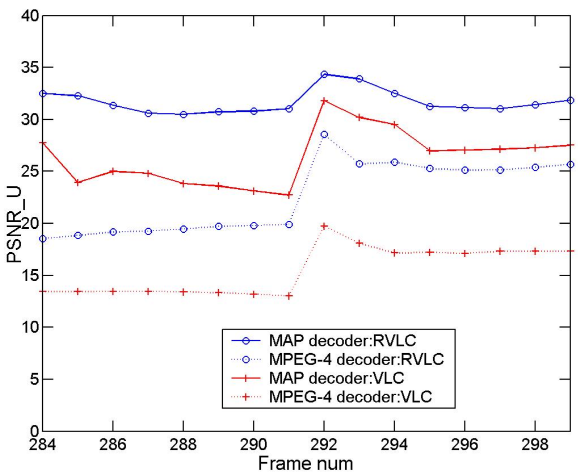

Fig. 5. PSNR of U component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$

(ρ_c) is set at 0.8. The results presented here are the average of eight realizations of the channel, corresponding to eight different error patterns. Figures 4, 5 and 6 provide a frame-by-frame comparison of the performance of the MAP decoder with the MPEG-4 decoder in terms of the average PSNR for the Y, U and V components of the decoded video frames when the inter and intra MBs are coded using (a) VLC and (b) RVLC codes. The channel error rate is fixed at $10^{-2.5}$ for these plots. Just as expected, the scheme employing RVLC does better than the one that uses only VLC. We also observe that for both the VLC and the RVLC cases, the MAP decoder performs much better than the standard MPEG-4 decoder.

In order to see the difference in improvement due to the MAP decoder for VLC and RVLC, the average PSNR *improvements* offered by the MAP decoder over the standard MPEG-4 decoder for the Y, U and V components for VLC and RVLC are given in Table II. As we can see, the improvement due to the

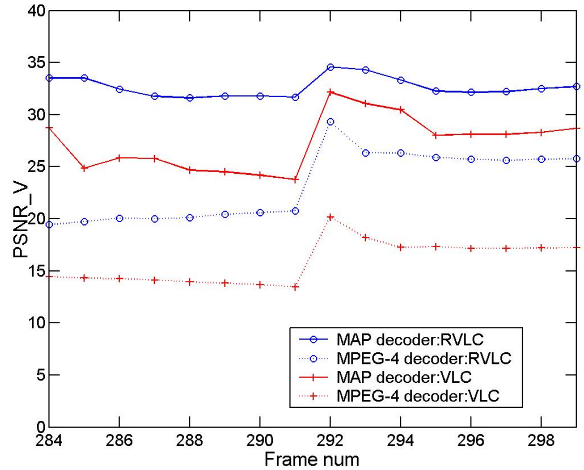

Fig. 6. PSNR of V component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$

TABLE II
AVERAGE PSNR IMPROVEMENTS OF MAP DECODER FOR VLC AND RVLC

	PSNR_Y	PSNR_U	PSNR_V
VLC	6.0	7.0	7.0
RVLC	4.6	5.4	5.6

MAP decoder is better for the VLC coded sequence than the RVLC coded sequence. Because the RVLC itself offers some error resilience and so the additional improvement due to the MAP decoder is reduced, although it is still significant. Figure 7 plots the average PSNR of the Y-component (PSNR_Y) of the whole video sequence at different channel bit error rates. This graph also shows the PSNR_Y of the sequence for the error free case, capturing only the source code induced noise, for both the RVLC encoded and VLC encoded case. From this figure, we see that MAP decoder performs significantly better than the standard MPEG-4 decoder, reaching 8 dB improvement when $\epsilon = 10^{-2.5}$ for the VLC coded streams and 6.4 dB improvement for the RVLC coded streams. Finally, in Figure 8 we observe the actual MAP and MPEG-4 decoded frames (frame number 284) for the VLC and RVLC coded sequence when $\epsilon = 10^{-2.5}$, giving a more perceptual idea of the performance of the decoders. The reconstructed frame in VLC and RVLC option without channel error are also included. As we can see, the MAP decoder seems to correct most of the block errors which the MPEG-4 error resilience tools is unable to handle for both RVLC and VLC coded frames.

B. JSCD with channel coding

In this part, channel coding is also incorporated in our scheme. The main purpose of this experiment is to evaluate the performance of the JSCD paradigm when FEC is also employed for error resilience. In MPEG-4 coded video streams, the headers and control bits are the most sensitive part because they control the decoding procedure. Among the remaining

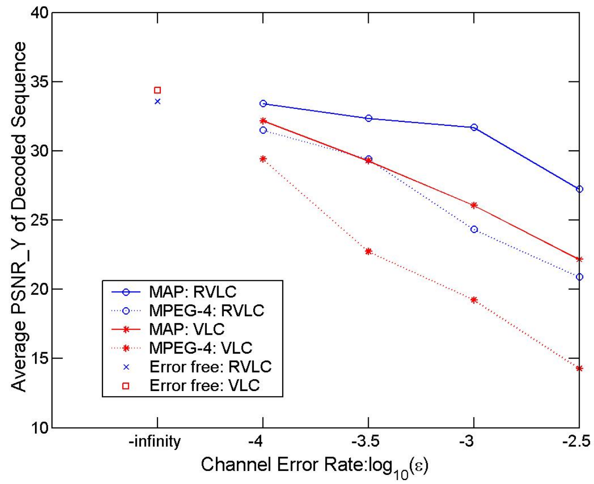

Fig. 7. Average PSNR of Y component for VLC and RVLC of the foreman sequence

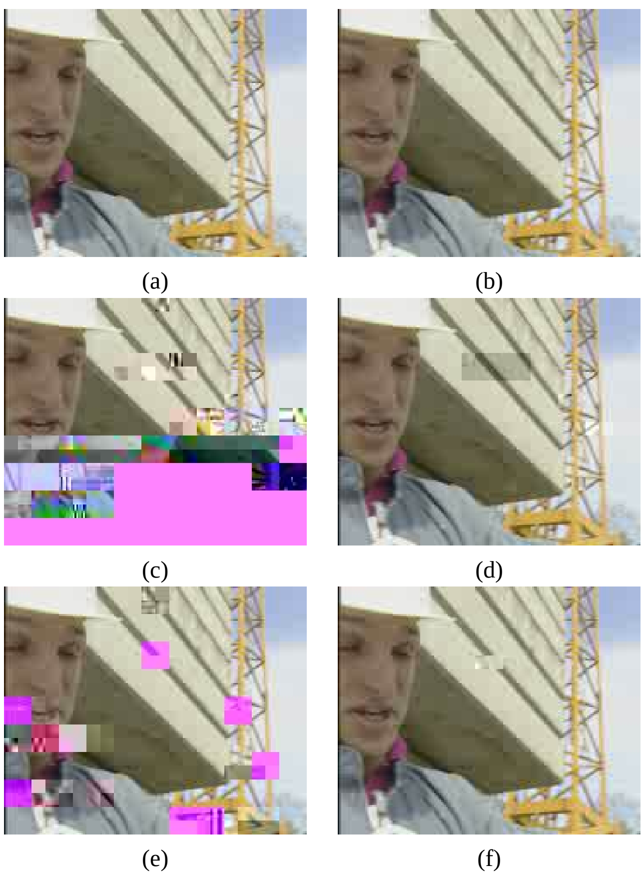

Fig. 8. Reconstructed frames without FEC.(a) Error free frame with VLC option. (b) Error free frame with RVLC option. (c) MPEG-4 decoder with VLC option. (d) MAP decoder with VLC option. (e) MPEG-4 decoder with RVLC option. (f) MAP decoder with RVLC option.

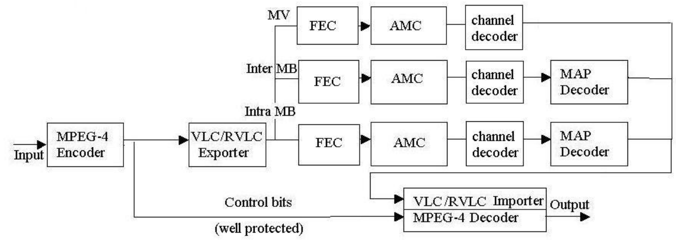

Fig. 9. Experimental Set-up for MAP decoder with FEC

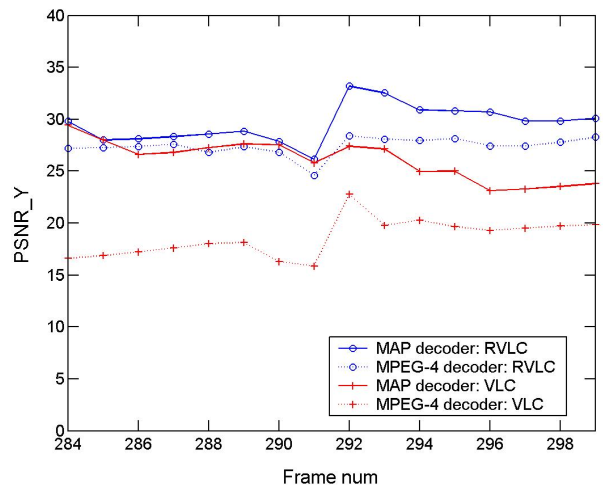

Fig. 10. PSNR of Y component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$ with FEC

bits, MVs are more important than the DCT coefficients, because motion vectors can be used for some error concealment approaches. For example, when a MB in the current frame is damaged, the decoder can copy the corresponding MB from the previously decoded frame based on the MV for this MB. Due to this inherent differences in the importance of the coded video stream, we use an unequal error protection (UEP) scheme: the $(127, 111)$ Reed-Solomon (RS) code for the motion vector and a weaker $(31, 27)$ RS code for the inter and intra MBs. Like in the earlier simulation, we assume that the header and control bits are very well protected using a powerful channel code and hence are not corrupted. The experimental set-up is depicted in Figure 9. MAP decoding is applied for the inter and intra MB after the channel decoder.

Since FECs can correct some of the errors in the received bit stream, the effective error rate at the channel decoder output will be less than the actual error rate. We use a set of training frames to estimate this error rate. To keep the overall bit rate the same as in the earlier simulation (0.164 bits per pixel), the source rate is reduced to 0.143 bits per pixel. As before, we also use data partitioning and resynchronization markers for added resilience. Other experimental setup and parameters are kept the same.

Figures 10, 11 and 12 give the PSNR for the Y, U and V components of decoded video frames for VLC and RVLC for both the MAP decoder and the regular MPEG-4 decoder when chan-

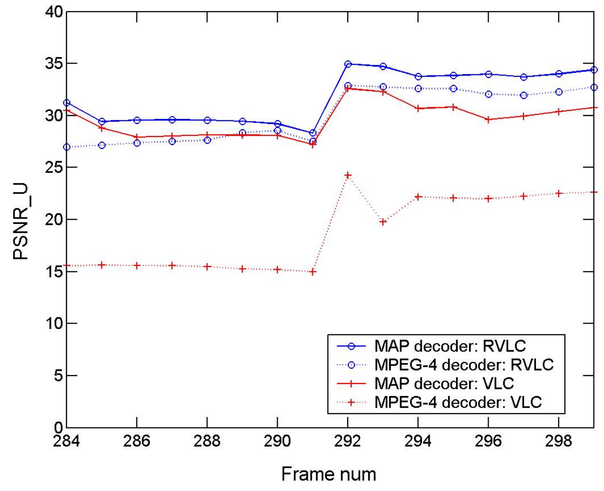

Fig. 11. PSNR of U component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$ with FEC

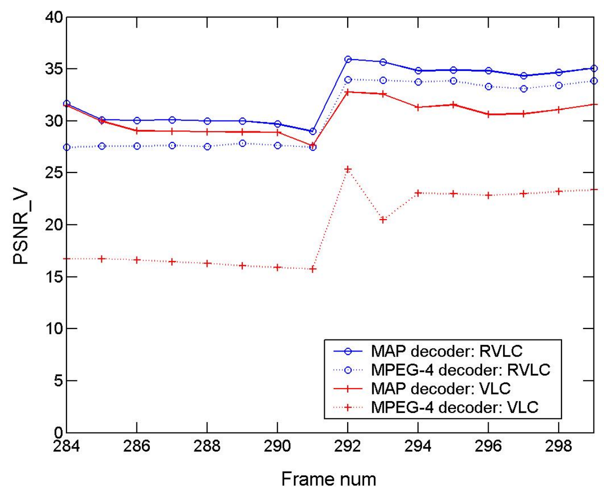

Fig. 12. PSNR of V component for VLC and RVLC coded MBs for the Foreman sequence at $\epsilon = 10^{-2.5}$ with FEC

nel error rate is $10^{-2.5}$. We observe that for both the VLC and the RVLC cases, the scheme employing FEC and MAP decoder performs much better than the FEC scheme without the MAP decoder. Figure 13 gives the average PSNR of the entire video sequence for all the schemes. The average PSNR of the MAP decoded RVLC and VLC sequence is compared for schemes with and without FEC in this graph. This figure also shows the performance of the standard MPEG-4 decoder for comparison. From this figure, we can observe that MAP + FEC for RVLC and MAP for RVLC perform best. When the channel error rate is high ($> 10^{-3}$), FEC is working well in conjuction with MAP for the RVLC. However when the channel error rate is too low ($< 10^{-4}$), MAP for RVLC is better. In addition, we see that MAP + FEC for VLC and MPEG-4 + FEC for VLC perform the second best where MAP for VLC is not good. Hence we see that the MAP decoder can be used to advantage in conjunction with FEC, RVLC or both. Figure 14 shows frame number 284

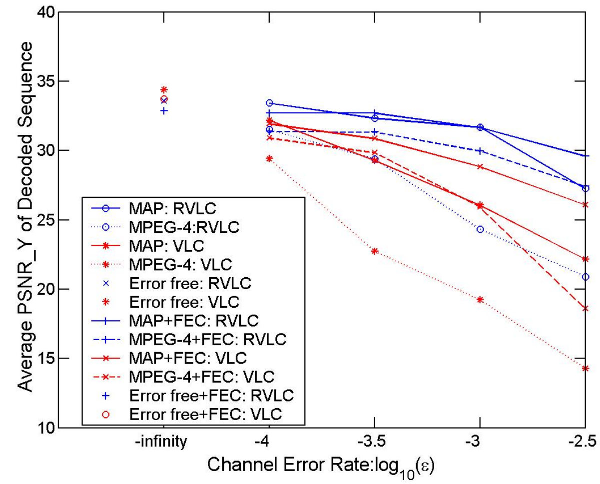

Fig. 13. Average PSNR of Y component for VLC and RVLC of the foreman sequence with FEC

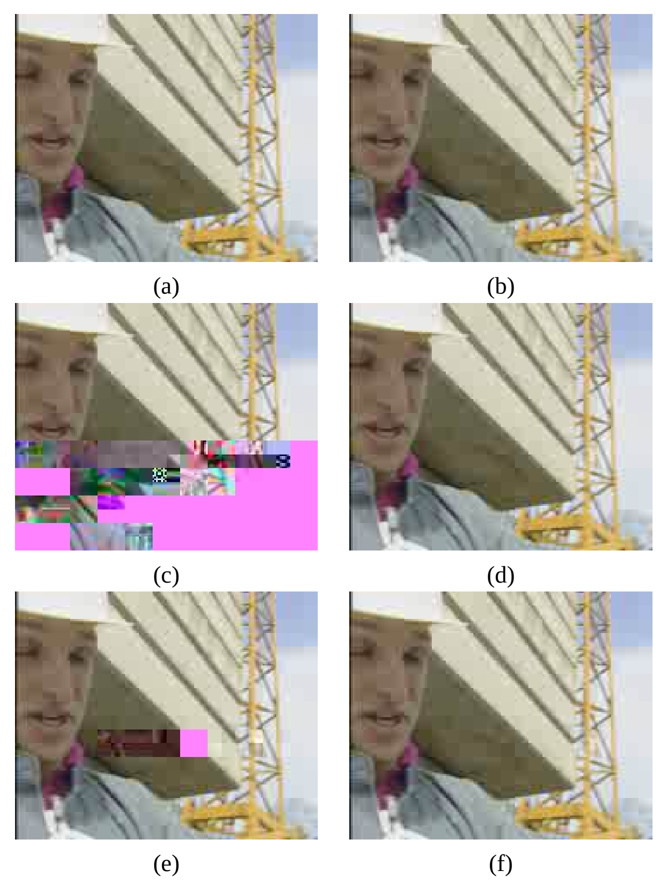

Fig. 14. Reconstructed frames with FEC. (a) Error free frame with VLC option. (b) Error free frame with RVLC option. (c) MPEG-4 decoder with VLC option. (d) MAP decoder with VLC option. (e) MPEG-4 decoder with RVLC option. (f) MAP decoder with RVLC option.

of the foreman sequence. Here the inter and intra macro blocks are coded using the VLC or RVLC and then channel coded using the RS code described earlier. The figure shows the performance of the MAP decoder and the MPEG-4 decoder when the channel error rate $\epsilon = 10^{-2.5}$. As expected, the scheme that includes a MAP decoder does perceptually better than the one that does not for both VLC and RVLC coded frames.

VII. Conclusions

This paper studied the problem of error resilient transmission of MPEG-4 encoded video over wireless channels. We developed a MAP decoder algorithm for the RVLC and VLC coded inter and intra macro blocks. Several experiments were conducted to study the applicability of the proposed JSCD algorithm for MPEG-4 transmission over wireless channels. It was seen that the proposed decoder can be incorporated in conjunction with or in lieu of the several error resilience features proposed in the MPEG-4 standard. We also showed through simulations that the proposed decoder compares favorably with traditional forward error correcting codes. From our experiments we conclude that the best performance is achieved when all error resilient features are co-opted when channel bit error rate is high. At low channel error rate, proposed MAP decoder for RVLC without FEC is the best. The worst performance is reported for the scheme that only uses resynchronization markers and no other error protection (like FEC, RVLC, or MAP decoding).

References

[1] *Information Technology: Coding of audio-visual objects*, ISO/IEC JTC1/SC29/WG11, Mar. 1998.

[2] *MPEG-4 video verification model version 16.0*, ISO/IEC JTC1/SC29/WG11, Mar. 2000.

[3] Y. Takishima, M. Wada, and H. Murakami, “Reversible variable length codes,” *IEEE Transactions on Communications*, vol. 43, pp. 158–162, 1995.

[4] J. Wen and J. D. Villasenor, “Reversible variable length codes for efficient and robust image and video coding,” in *Data Compression Conference*, 1998, pp. 471–480. [Online]. Available: citeseer.nj.nec.com/wen98reversible.html

[5] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, “Error resilient video coding techniques,” *IEEE Signal Processing Magazine*, vol. 17, pp. 61–82, July 2000.

[6] C. E. Shannon, “The mathematical theory of communication,” *Bell Sys. Tech. Journal*, vol. 28, pp. 379–423, Oct. 1949.

[7] S. Vembu, S. Verdu, and Y. Steinberg, “The source-channel separation theorem revisited,” *IEEE Transactions on Information Theory*, vol. 41, pp. 44–54, Jan. 1995.

[8] J. G. Dunham and R. M. Gray, “Joint source and noisy channel trellis encoding,” *IEEE Transactions on Information Theory*, vol. 27, pp. 516–519, July 1981.

[9] N. Farvardin, “A study of vector quantization for noisy channels,” *IEEE Transactions on Information Theory*, vol. 36, pp. 799–809, July 1990.

[10] N. Phamdo, F. Alajaji, and N. Farvardin, “Quantization of memoryless and Gauss-Markov sources over binary Markov channels,” *IEEE Transactions on Communication*, vol. 45, pp. 668–674, June 1997.

[11] K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of joint source channel coders,” *IEEE Transactions on Communications*, vol. 39, pp. 839–846, June 1991.

[12] K. Sayood, F. Liu, and J. D.Gibson, “A constrained joint source/channel coder design,” *IEEE Transactions on Communications*, vol. 12, pp. 1584–1593, Dec. 1994.

[13] N. Phamdo and N. Farvardin, “Optimal detection of discrete Markov sources over discrete memoryless channels,” *IEEE Transactions on Information Theory*, vol. 40, pp. 186–193, Jan. 1994.

[14] M. Park and D. J. Miller, “Low-delay optimal MAP state estimation in HMMS with application to symbol decoding,” *IEEE Signal Processing Letters*, vol. 4, pp. 289–292, Oct. 1997.

[15] N. Demir and K. Sayood, “Joint source/channel coding for variable length codes,” in *IEEE Data Compression Conference*, Snowbird, UT, 1998, pp. 139–148.

[16] M. Park and D. J. Miller, “Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation,” *IEEE Transactions on Communications*, vol. 48, pp. 1–6, 2000.

[17] J. Wen and J. D. Villasenor, “Utilizing soft information in decoding of variable length codes,” in *Data Compression Conference*, Snowbird, UT, 1999, pp. 131–139.

[18] R. Bauer and J. Hagenauer, “Symbol by symbol MAP decoding of variable length codes,” in *3rd ITG Conference Source and Channel Coding*, Munich, Germany, Jan. 2000.

[19] ——, “Iterative source/channel-decoding using reversible variable length codes,” in *IEEE Data Compression Conference*, 2000, pp. 93–102.

[20] A. Hedayat and A. Nosratinia, “On joint iterative decoding of variable-length codes and channel codes,” in *IEEE Conference on Communications, Control, and Computing*, Oct. 2002.

[21] J. Kliewer and R. Thobaben, “Combining FEC and optimal soft-input source decoding for the reliable transmission of correlated variable-length encoded signals,” in *IEEE Data Compression Conference*, 2002, pp. 83–91.

[22] L. Guivarch, J.-C. Carlach, and P. Siohan, “Joint source-channel soft decoding of Huffman codes with turbo-codes,” in *Data Compression Conference*, 2000, pp. 83–92.

[23] K. P. Subbalakshmi and J. Vaisey, “Optimal decoding of entropy coded memoryless sources over binary symmetric channels,” in *IEEE Data Compression Conference*, Mar.-Apr. 1998, p. 573.

[24] ——, “On the joint source-channel decoding of variable-length encoded sources: the BSC case,” *IEEE Transactions on Communications*, vol. 49, pp. 2052–2055, Dec. 2001.

[25] ——, “Joint source-channel decoding of entropy coded Markov sources over binary symmetric channels,” in *IEEE International Conference on Communications*, vol. 1, June 1999, pp. 446–450.

[26] A. H. Murad and T. E. Fuja, “Joint source-channel decoding of variable-length encoded sources,” in *IEEE Information Theory Workshop*, Killarney, Ireland, June 1998.

[27] ——, “Exploiting the residual redundancy in motion estimation vectors to improve the quality of compressed video transmitted over noisy channels,” in *Proc. Int. Conf. Image Processing*, Oct. 1998.

[28] K. Lakovic, J. Villasenor, and R. Wesel, “Robust joint Huffman and convolutional decoding,” in *IEEE Vehicular Technology Conference*, 1999, pp. 2551–2555.

[29] K. Lakovic and J. Villasenor, “Combining variable length codes and turbo codes,” in *IEEE Vehicular Technology Conference*, 2002, pp. 1719–1723.

[30] F. Alajaji and T. Fuja, “A communication channel modeled on contagion,” *IEEE Transactions on Information Theory*, vol. 40, pp. 2035–2041, Nov. 1994.

[31] F. Alajaji, N. Farvardin, and T. Fuja, “Detection of binary Markov sources over channels with additive Markov noise,” *IEEE Transactions on Information Theory*, vol. 42, pp. 230–239, Jan. 1996.

[32] P. Burlina and F. Alajaji, “An error resilient scheme for image transmission over noisy channels with memory,” *IEEE Transactions on Image Processing*, vol. 7, pp. 593–600, Apr. 1998.

[33] K. P. Subbalakshmi and J. Vaisey, “Optimal decoding of entropy coded Markov sources over channels with memory,” in *The 33rd Annual Conference on Information Sciences and Systems*, vol. 2, Mar. 1999, pp. 624–629.

[34] ——, “On the joint source-channel decoding of variable-length encoded sources: The additive markov channel case,” *IEEE Transactions on Communications, (To Appear)*.

[35] A. Kopansky and M. Bystrom, “Sequential decoding of MPEG-4 coded bitstreams for error resilience,” in *The 33rd Annual Conference on Information Sciences and Systems*, Mar. 1999.

[36] M. Bystrom, S.Kaiser, and A.Kopansky, “Soft source decoding with applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1108–1120, Oct. 2001.

[37] R. M. Fano, “A heuristic discussion of probabilistic decoding,” *IEEE Transactions on Information Theory*, vol. IT-9, Apr. 1963.

[38] K. P. Subbalakshmi and Q. Chen, “Joint source-channel decoding for MPEG-4 coded video over wireless channels,” in *IASTED International Conference on Wireless and Optical Communications*, July 2002, pp. 617–622.

[39] Q. Chen and K. P. Subbalakshmi, “Trellis decoding for MPEG-4 streams over wireless channels,” in *Proc. SPIE Electronic Imaging: Image and Video Communications and Processing*, Jan. 2003, pp. 810–819.

[40] B. Hochwald and K. Zeger, “Tradeoff between source and channel coding,” *IEEE Transactions on Information Theory*, pp. 1412–1424, Sept. 1997.

[41] M. Bystrom and J. W. Modestino, “Combined source-channel coding for transmission of H.263 coded video with trellis-coded modulation over a slow fading Rician channel,” in *IEEE International Symposium on Information Theory*, Aug. 1998, p. 12.

[42] G. Cheung and A. Zakhor, “Bit allocation for joint source/channel coding of scalable video,” *IEEE transactions on Image Processing*, vol. 9, pp. 340–356, Mar. 2000.

[43] A. Makur and K. P. Subbalakshmi, “Variable dimension VQ encoding and codebook design,” *IEEE Transactions on Communications*, vol. 45, no. 8, pp. 897 – 900, Aug. 1997.

PLACE PHOTO HERE

Qingyu Chen obtained his B.E. and M.E. degree from Huazhong University of Science and Technology, Wuhan, China in 1998 and 2001, respectively, both in Electrical Engineering. Since September 2001, he has been working toward the Ph.D. degree in Electrical Engineering at the Stevens Institute of Technology, Hoboken, NJ. His current interests include joint source-channel coding, error resilience in video transmission and multimedia networking.

PLACE PHOTO HERE

K.P. Subbalakshmi received the B.Sc degree in Physics from the University of Madras, India in 1990, the M.E in Electrical Communication Engineering from the Indian Institute of Science in 1994 and the PhD degree from the School of Engineering Science, Simon Fraser University, Canada in 2000. Since Fall 2000, she has been an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, where she co-directs the *MSyNC*: Multimedia Systems Networking and Communications Laboratory. Dr. Subbalakshmi leads research projects in information security, joint source-channel, multiple description coding and multimedia networking funded by the National Science Foundation, Air Force Research Laboratory, New Jersey Center for Wireless Telecommunications, the Stevens Center for Wireless Network Security and industry partners. She has been an active participant in several international conference program committees and organizations.