A General Joint Source-Channel Matching Method for Wireless Video Transmission

Leiming Qian, Douglas L Jones, Kannan Ramchandran
and Swaroop Appadwedula

Coordinated Science Laboratory

University of Illinois at Urbana-Champaign

1308 W. Main Street, Urbana, IL 61801

Ph. (217) 333-5860 E-Mail: lqian@uiuc.edu

Abstract

With the rapid growth of multimedia content in wireless communication, there is an increasing demand for efficient image and video transmission systems. We present a joint source-channel matching scheme for wireless video transmission which jointly optimizes the source and channel coder to yield the optimal transmission quality while satisfying real-time delay and buffer constraints. We utilize a parametric model approach which avoids the necessity of having detailed *a priori* knowledge of the coders, thus making the scheme applicable to a wide variety of source and channel coder pairs. Simulations show that the scheme yields excellent results and works for several different types of source and channel coders.

1 Introduction

The advantages of jointly optimized source-channel coding for image transmission over systems with finite delay and complexity constraints have by now been widely demonstrated (see [1] for a extensive survey as well as the references in Section 1.1); many different groups have developed methods achieving significant improvements in performance over conventional techniques for wireless image transmission. Due to typically higher data rates and a broad spectrum of applications, wireless video transmission is likely to be of even greater interest, yet relatively little work on joint source-channel video coding has appeared. Video source coders are generally more complex than image coders. Most importantly, real-time delay constraints and rate and buffer control introduce significant new challenges which must be addressed in effective joint-source channel coding for wireless video transmission.

Real-world system implementations often involve the additional constraints of preselected source-channel coder pairs (often based on standards such as H.263 or MPEG), varying transmission conditions, and heterogeneous multimedia content.

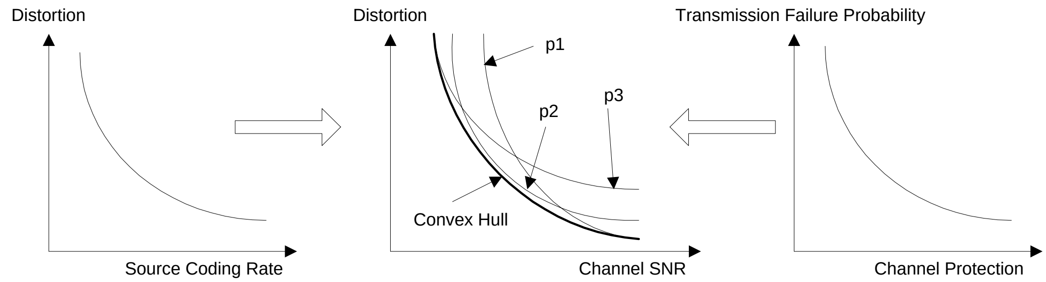

Figure 1: Joint Source-Channel Coding

Thus it is also desirable to develop a general matching scheme, which not only can be applied to a wide variety of source-channel coder pairs and varying channel conditions, but which also provides optimal visual quality under all circumstances.

This paper presents such a scheme for wireless video transmission. It utilizes a parametric model of the source coder performance to provide generality, and optimizes the end-to-end performance while including the critical delay, rate, and buffer control constraints required by video. Its effectiveness and generality is illustrated using two block-transform-based video source coders (Motion-JPEG and Conditional Block Replenishment) and the Reed-Solomon channel coder. The system should also work with other source and channel coders, such as motion-compensation-based video coders (MPEG, H.263) and convolutional channel coders.

1.1 Joint Source-Channel Matching

Most current image and video source coders are optimized to give the best performance at a certain rate while assuming all the coded bits are received perfectly. Likewise, most existing channel coders are designed for a specific channel and target BER without explicit regard for the source characteristics. Figure 1 illustrates the motivation for matching *jointly* the source and channel coder. In Figure 1, the leftmost plot shows that increasing the source coding rate (less compression) decreases the final end-to-end distortion; the rightmost plot shows that increasing channel protection can reduce possible channel errors, which also decreases the final end-to-end distortion. However, because we have a limited channel capacity, there is a fundamental trade-off between these two rates. The curves ($p1$, $p2$, $p3$) in the middle plot show how a conventional system with fixed source rate and channel protection performs for different channel situations. Depending on the signal-to-noise ratio (SNR), different parameters lead to optimal performance; a jointly optimized source-channel coder can obtain maximal performance in all regimes.

The advantages of joint source-channel coding for image transmission have been extensively studied; here we present an incomplete list of previous research: Davis and Danskin [2] described a joint source-channel allocation scheme for transmitting images losslessly over block erasure channels such as the Internet; Ramchandran *et al.* [3] studied multiresolution coding and transmission in a broadcasting scenario; Azami *et al.* [4] acquired performance bounds for joint source-channel coding of uniform memoryless sources using binary decomposition; Belzer *et al.* [5] developed a joint source-channel image coding method using trellis-coded quantization and convolutional codes; Sherwood and Zeger [6] investigated unequal error protection for the binary symmetric channel; and Man *et al.* [7] examined unequal error protection and source quantization. A general method which can be applied to most source and channel coders can be found in [8].

Joint source-channel coding for wireless video transmission is much less mature. Ortega and colleagues have made important contributions, particularly in the key area of rate control [9] [10], which is discussed next.

1.2 Rate Control

Rate control is a key issue in video transmission [9] [10], since a small, finite delay between transmission of frames must be obtained. Furthermore, in a practical implementation the decoder usually has a finite decoding buffer-size and a constant decode-flow.

Suppose we have a video frame sequence divided into groups of pictures (GOPs) of size N. Each picture or frame is coded to have a stream length of $s(i, q_i)$ in bits, where i is the frame index and q_i is the source-coding rate-control factor (for example, the quality factor in JPEG coding). For the transmission of the ith frame a certain amount of protection, denoted by p_i, is used. Then the buffer occupancy b_i after frame i is transmitted can be expressed as

$$b_i = \max(b_{i-1} + s(i, q_i) + p_i - R, 0),\ i = 1, 2, \ldots, N \tag{1}$$

where R is the constant decoding stream flow (stuffing-bits will be added to avoid underflow when b_i is smaller than zero). The buffer size constraint is then represented as

$$b_i \leq b_{max},\ i = 1, 2, \ldots, N \tag{2}$$

where b_{max} is the finite decoding buffer size. The optimization thus becomes a constrained problem.

1.3 General System Diagram

Without specifying the actual source and channel coders, our scheme can be illustrated using a general system diagram (Figure 2) which contains seven major blocks. Although the exact implementation of these blocks may differ for different coders, they take advantage of the minimal control we have on the coders such as source rate and protection symbols.

2 Motion-JPEG and RS coder

We demonstrate the flexibility and generality of our method by applying it to two different source coders. In this section, we use Motion-JPEG as the video source

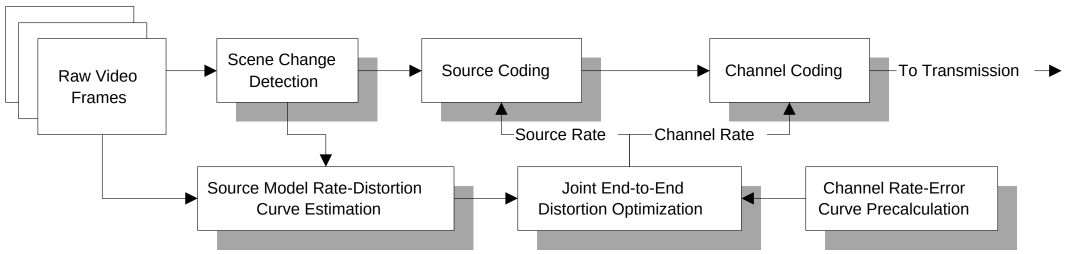

Figure 2: General System Diagram

coder and Reed-Solomon (RS) channel coder. Motion-JPEG is perhaps the simplest existing video coder; it simply codes each frame in JPEG format and transmits the coded frames sequentially. In spite of its simplicity, it is commonly used because of the constant frame quality, its ability to start decoding at any particular frame, and no frame-to-frame error propagation. The limited frame delay constraint of real-time video introduces a significant challenge over previous work involving images.

2.1 Motion-JPEG Video Coder

The JPEG [11] image coder is an efficient and popular image coding method which is also used by many video coding standards, such as Motion-JPEG and MPEG, to code individual frames. It is important to note that because of the non-progressive nature of the JPEG bitstream, a random error in the bitstream will introduce unpredictable distortion (sometimes even total image loss), as opposed to progressive coders such as SPIHT [12], where the bits are ordered by importance.

Thus we define the *expected* end-to-end distortion for JPEG-coded frames as

$$D = P_{\text{ffail}} \times D_{\text{loss}} + P_{\text{fsucc}} \times D_{\text{succ}} \tag{3}$$

where D_{loss} and D_{succ} are the resulting final distortions corresponding to a failed frame transmission and a successful frame transmission, respectively. P_{fsucc} and P_{ffail} are the probability of successful frame transmission and probability of frame transmission failure respectively.

2.2 Reed-Solomon Coder and Packet-Based Approach

Reed-Solomon (RS) codes [13] are a well-known class of block codes with good error-correction properties. We chose them to demonstrate our scheme because of their ability to correct channel burst errors, which are common in a wireless environment. An RS code defined by (n, k, t) is a length-n code which contains $k = n - 2t$ source symbols, $2t$ protection symbols, and can correct t symbol errors. There are RS codes for various n, most commonly $n = 2^m - 1$ where m is the symbol length in bits (in our scheme we will use $m = 8$, because we usually access images in bytes). An (n, k, t) code will be unable to recover the original k data symbols if more than t errors occur.

Since the RS coder creates fixed-length codes, we employ a packet-based approach in channel coding, in which each transmission packet is an RS coding unit containing both the source symbols and protection symbols. In order to calculate the expected distortion, we define, for video transmission, the *frame transmission success probability*, P_{fsucc}, as:

$$\begin{aligned}
P_{\text{fsucc}} &= \prod_{n=1}^{M} P_{\text{psucc}} && (4)\\
P_{\text{psucc}} &= \sum_{k=0}^{p/2} \binom{L}{k} {P_{\text{es}}}^{k} {P_{\text{ss}}}^{L-k} && (5)\\
P_{\text{ss}} &= 1-(1-P_{\text{eb}})^{m} && (6)\\
P_{\text{eb}} &= Q\left(\sqrt{\frac{2E_b}{N_0}}\right) && (7)
\end{aligned}$$

where P_{psucc} is the *packet transmission success probability*, M is the total number of packets to transmit, P_{ss} is the *symbol transmission success probability*, L is the packet length, p is the number of protection symbols in each packet, P_{eb} is the bit-error probability, m is the number of bits per symbols, E_b is the fixed power per bit, and $\frac{N_0}{2}$ is the channel noise variance. In these equations, the binomial pdf is used to determine the probability of packet error with the safe assumption that a bit error results in a symbol error. In order for the above equation to be valid, sufficient interleaving must be employed so that the independence assumption is satisfied.

2.3 Problem Formulation

To formulate the problem when the source coder is Motion-JPEG, we assume that we have a series of video frames that are divided into GOPs of size N. We define the optimization problem as minimizing the final end-to-end distortion across the two variables q and p while meeting the buffer constraint, where q is the quality-factor used to JPEG-encode each frame, and p is amount of protection used for transmitting all the packets contained in a certain frame.

A more general problem formulation would have a quality factor and protection amount associated with each frame. However, experimental evidence revealed that the distortion *vs.* quality-factor curve and the source rate *vs.* quality-factor curve are very similar for all the frames in the same GOP except when there is a scene change.

The final optimization problem can be formulated as finding the optimal scalars q and p which minimize the final distortion, D:

$$\begin{aligned}
D &= \frac{1}{N}\sum_{i=1}^{N} D(i) && (8)\\
D(i) &= P_{\text{fsucc}}(p)\times d(q) + P_{\text{ffail}}(p)\times d_{\text{conceal}}(i) && (9)\\
P_{\text{fsucc}} &= \prod_{n=1}^{M(q)} P_{\text{psucc}} && (10)
\end{aligned}$$

Here D is the average end-to-end distortion, $D(i)$ is the expected distortion for the ith frame, $P_{\text{fsucc}}(p)$ and $P_{\text{ffail}}(p)$ are the frame transmission success and failure probabilities when we use p protection symbols for each block, $d(q)$ is the distortion for each frame when coded using quality-factor q, and $d_{\text{conceal}}(i)$ is the distortion when we use the $i-1$th frame in place of the ith frame.

We must also satisfy the following decoding buffer-size constraints:

$$b_i = max(b_{i-1} + s(q) + p - R, 0) \quad (11)$$

$$b_i \leq b_{max}, \; i = 1, 2, \ldots, N \quad (12)$$

where $s(q)$ is the frame length when coded using quality-factor q, and L is the predefined RS packet size.

2.4 Implementation Issues

The constrained optimization problem described above is solved using a gradient-projection or penalty function algorithm, and generally requires no more than three iterations to converge. The system complexity is of the same order as DCT coding. While the basic algorithm above is conceptually straightforward, there are quite a few detailed implementation issues involved in a practical scheme. For details and the system diagram of our proposed real-time system, please refer to [14].

3 Conditional Block Replenishment and RS coder

We chose the Conditional Block Replenishment (CBR) coder as our second example because of its wide usage in video conferencing applications, where the frames consist of mostly still background and not much movement, and in multicast systems such as M-Bone on the Internet. It has certain advantages, such as coding simplicity and substantial robustness to transmission errors, over motion-compensation-based video coders like MPEG.

3.1 CBR Coder Basics

A CBR coder works in the following fashion: first, blockwise differences between two consecutive frames are calculated, and then only those blocks with a difference greater than a certain predefined threshold are coded and sent. There are several ways to encode the blocks; the most common is DCT coding, followed by quantization and symbol coding.

3.2 Problem Formulation

Instead of using just the minimal average mean square distortion as the optimization criterion, here to demonstrate the generality of our scheme, we utilize another type of criterion based on the characteristics of the human visual system (HVS). It is known that the HVS is very sensitive to image quality variation; in other words, a video sequence with a constant or near-constant level of distortion is more desirable than one with lower average distortion but higher variability [15]. Thus in our optimization scheme, we are trying to achieve an optimal balance between the average image distortion and image quality variation, while at the same time satisfying the rate-control constraint. This can be mathematically represented as:

$$
\begin{aligned}
D &= \frac{1}{N}\sum d_i(t_i) && (13)\\
V &= \sigma(\mathbf{d}), \mathbf{d} = (d_1(t_1), d_2(t_2), ..., d_N(t_N)) && (14)\\
d_i(t_i) &= P_{\text{ffail}} \times d_{\text{conceal}}(i) + P_{\text{fsucc}} \times d_{\text{succ}}(i, t_i) && (15)\\
b_i &= \max(b_{i-1} + s(q) + p - R, 0) && (16)\\
b_i &\leq b_{max},\ i = 1, 2, ..., N && (17)
\end{aligned}
$$

where D is the final average frame distortion, V is the standard deviation of the vector $\mathbf{d}$, which consists of individual frame distortion values, $d_i(t_i)$, and t_i is the threshold value used in coding the ith frame. The other variables are defined as in the Motion-JPEG section.

As is the case with the Motion-JPEG example, there are several implementation issues affecting this scheme's efficiency and effectiveness. For implementation details and a system diagram, please refer to [14].

4 Results

We demonstrate the effectiveness of the proposed methods by applying them to real video sequences transmitted over simulated additive white Gaussian noise channels. For the simulations, we use the following video parameters: the testing sequence is the 352 × 240 Football sequence (which is considered a difficult sequence); we use an RS coder size of 256 symbols (2048 bits), a decoder buffer size of 21,120 bits (approximately the size of two compressed motion-JPEG frames), a constant decode flow of 15,840 bits (approximately 1.5 times the compressed frame size), and a GOP size of 20.

4.1 Motion-JPEG and RS Coder

The end-to-end performance (PSNR) over a range of SNRs for motion-JPEG video with RS channel coding is illustrated in Figure 3. The results using our optimal matching algorithm are compared with the performance of three conventional fixed-rate/fixed-protection solutions. For SNRs below a threshold, the fixed solutions exhibit rapid deterioration to very poor performance, while above the threshold their performance saturates at a level well below the optimal. In contrast, the source-channel matched solution exhibits excellent performance over a very wide range of channel SNRs and always performs at or above the performance of any fixed system. This illustrates the advantages both of optimal parameter matching and the ability of this method to obtain maximal performance at all times under varying channel conditions.

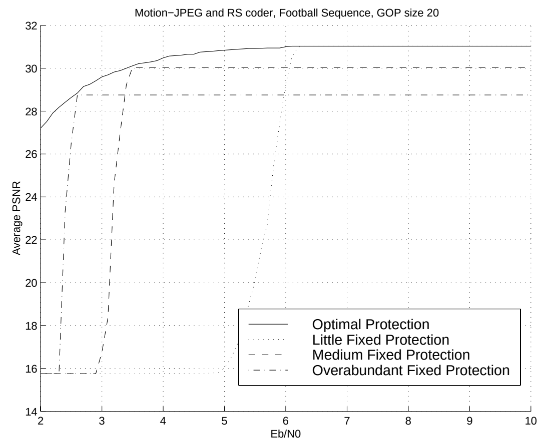

Figure 3: Motion-JPEG and RS Coder Simulation Results

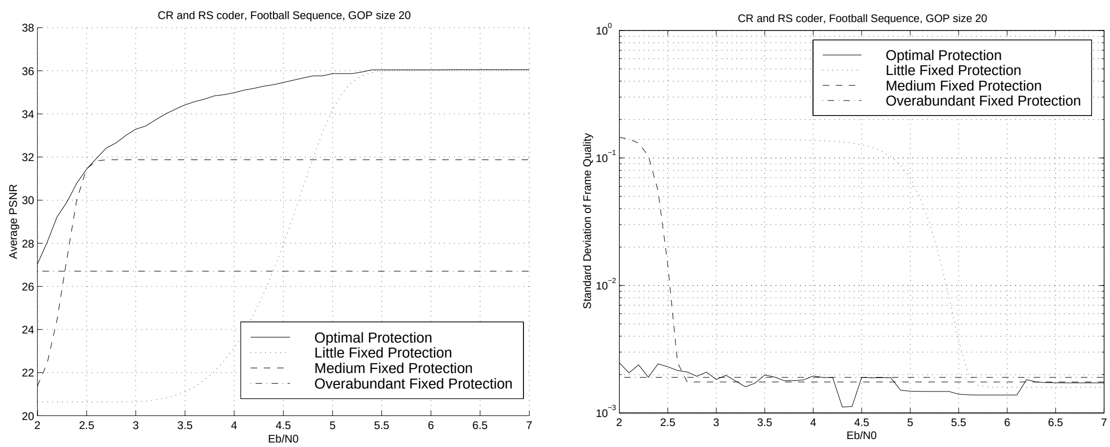

Figure 4: Conditional Block Replenishment and RS Coder Simulation Results

4.2 CBR and RS Coder

The end-to-end performance (PSNR) over a range of SNRs for CBR-coded video with RS channel coding is illustrated in Figure 4, where the standard default JPEG quantization matrix was used. (As expected, the CBR coder requires much less bandwidth than the Motion-JPEG coder to achieve the same PSNR.) The results using our optimal matching algorithm are compared with the performance of three conventional fixed-rate/fixed-protection solutions.

From the average frame PSNR plot we can reach the same conclusions as in the Motion-JPEG case: our scheme truly minimizes the end-to-end distortion in all channel situations, while systems not employing joint source-channel matching either behave badly in certain regions or fail to take advantage of a favorable channel. From the frame-quality-variation plot we observe that although our scheme does not always give the smallest variation, the variation is always small and quite acceptable, and those configurations which give smaller variations always have a much greater average frame distortion; our scheme truly strikes a balance between the average frame quality and the quality variation. By putting different weights on these two criteria, we have another degree of freedom to adapt to user requirements.

5 Conclusions

Generally higher rates, more complex coders, and, most importantly, real-time delay and buffer-size constraints, create additional challenges for joint-source channel matching for video in comparison with methods for image transmission. We have developed a general scheme to implement joint source-channel matching for wireless video transmission which offers very high performance over a wide range of channel conditions, has a computational complexity of the same order as fixed schemes, and which can be easily adapted to a wide variety of source and channel coders. Simulations with real video sequences demonstrate the dramatic performance benefits obtained over conventional fixed systems and the ability to adapt optimally to varying channel conditions. Our system is truly general because of an inexpensive parametric-model approach for characterizing the source-coder performance, which is implemented online by evaluating a few "control points" followed by interpolation. Thus our method can easily be applied to a large number of coders, including standard ones, without modification or explicit knowledge of their internal structure. In future work we will apply our scheme to motion-compensation-based video source coders such as MPEG and H.263, other channel coders such as convolutional coders, and other channel models such as fading channels.

References

[1] S. B. Z. Azami, P. Duhamel, and O. Rioul, "Joint source-channel coding: Panorama of methods," in *Proceedings of CNES Workshop on Data Compression*, Nov. 1996.

[2] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *SPIE Conference on Wavelet Applications of Digital Image Processing XIX*, Aug. 1996.

[3] K. Ramchandran, A. Ortega, K. Uz, and M. Vetterli, "Mutilresolution broadcast for digital HDTV using joint source/channel coding," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 6–23, Jan. 1993.

[4] S. B. Z. Azami, O. Rioul, and P. Duhamel, "Performance bounds for joint source-channel coding of uniform memoryless sources using a binary decomposition," in *Proceedings of European Workshop on Emerging Techniques for Communication Terminals*, pp. 259–263, Jul. 1997.

[5] B. Belzer, J. D. Villasenor, and B. Girod, "Joint source-channel coding of image with trellis coded quantization and convolutional codes," in *Proceedings of ICIP'95*, vol. 2, Aug. 1995.

[6] G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Letters*, vol. 4, pp. 72–81, Jul. 1997.

[7] H. Man, F. Kossentini, and M. Smith, "Progressive image coding for noisy channels," *IEEE Signal Processing Letters*, vol. 4, pp. 8–11, Aug. 1997.

[8] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzintsev, "Joint source-channel matching for a wireless communications link," in *Proceedings of ICC 98*, Jun. 1998.

[9] C.-Y. Hsu and A. Ortega, "Rate control for robust video transmission over wireless channels," in *Proceedings of VCIP97*, Feb. 1997.

[10] A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," in *Proceedings of ICIP'95*, Oct. 1995.

[11] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, pp. 4–21, Apr. 1991.

[12] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," in *IEEE Trans, Circuits and Systems for Video Technology*, vol. 6, Jun. 1996.

[13] R. E. Blahut, *Digital Transmission of Information.* Reading, MA: Addison-Wesley, 1990.

[14] L. Qian, "Joint source-channel matching for wireless video transmission," Master's thesis, University of Illinois at Urbana-Champaign, http://www.ifp.uiuc.edu/˜ lqian, 1998.

[15] D. T. Hoang, E. L. Linzer, and J. S. Vitter, "Lexicographic bit allocation for MPEG video," *Journal of Visual Communication and Image Representation*, Dec. 1997.