Joint Source-Channel Matching for Wireless Video Transmission

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

June 30, 1999

Abstract

With the rapid growth of multimedia content in wireless communication, there is an increasing demand for efficient image and video transmission schemes. In this paper we present a joint source-channel matching scheme for wireless video transmission. The scheme jointly optimizes the source and channel coder parameters to yield the optimal transmission quality. 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. Our results show that the scheme yields excellent results and works for many different types of source and channel coder pairs.

1 Introduction

Although Shannon's separation theorem points out that source and channel coders can be optimized separately without any loss of overall performance, it is not yet known whether this result still holds for systems with finite delay, or in a broadcasting environment with multi-path fading. Thus we are motivated to utilize joint source-channel matching techniques, where system resources are assigned by trading off information and redundancy [1].

While significant work has been done on joint source-channel matching techniques for specific source and channel coders, system designers usually have to deal with preselected source-channel coder pairs, varying transmission conditions, and heterogeneous multimedia content. Thus it is desirable to develop a more general parametric-model-based 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, illustrated using a Motion-JPEG video coder and a Reed-Solomon channel coder for a BPSK channel; the result can be extended to other source and channel coders.

1.1 Joint Source-Channel Matching

Most of the image and video source coders today are optimized to give the best performance at a certain rate assuming all the coded bits are received perfectly. Likewise, most existing channel coders are characterized in terms of transmission power, data rate, and channel error probability, and are not necessarily well matched to specific source coders. The reason behind matching jointly the source and channel coder can be best represented by Figure 1.

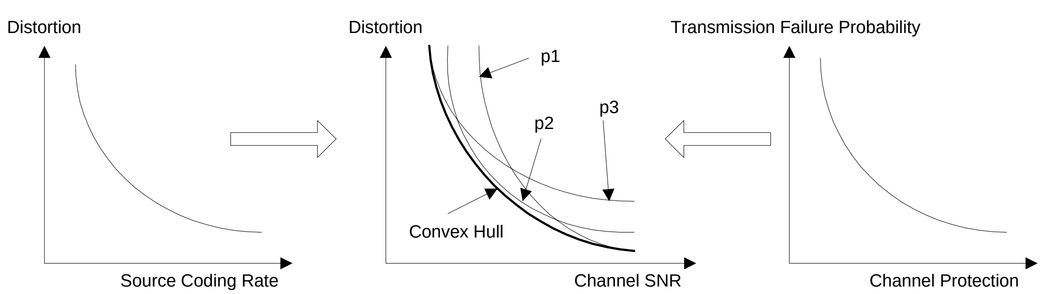

Figure 1: Joint Source-Channel Coding

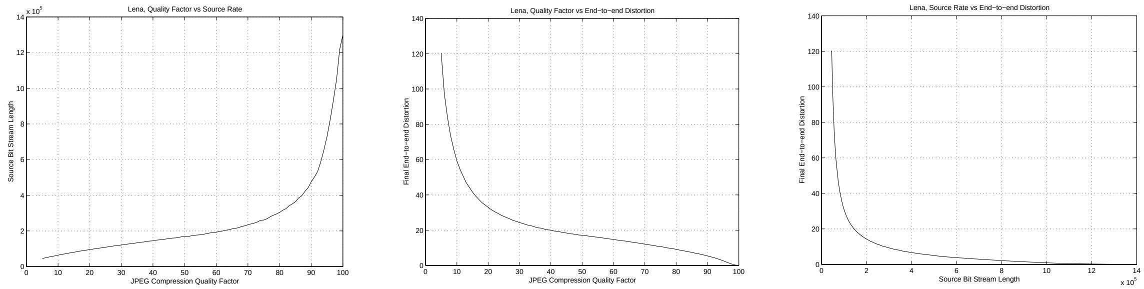

Figure 2: Quality Factor, Source Rate and End-to-end Distortion

In Figure 1, the left plot shows that increasing the source coding rate (less compression) decreases the final end-to-end distortion; the right plot shows that increasing channel protection can reduce possible channel errors, which also decreases the final end-to-end distortion. However, since 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 the way how a conventional system with fixed source rate and channel protection works for different channel situation. Each point on the convex hull of these curves corresponds to the optimal configuration with regard to the current channel situations. Research has been done on this topic with significant results: Lan and Tewfik examined the trade-off between source and transmission power [2]; Davis and Danskin described a joint source-channel allocation scheme for transmitting images losslessly over block erasure channels such as the Internet [3]; Ramchandran *et al.* studied multi-resolution coding and transmission in a broadcasting scenario [4]; Azami, Rioul and Duhamel acquired performance bounds for joint source-channel coding of uniform memoryless sources using binary decomposition [5]; Belzer, Villasenor and Girod developed a joint source-channel image coding method using trellis-coded quantization and convolutional codes [6]; Sherwood and Zeger investigated unequal error protection for the binary symmetric channel [7]; Man, Kossentini and Smith examined unequal error protection and source quantization [8]. An extensive survey on general joint source-channel matching was done by Azami, Duhamel and Rioul in [1].

All of these results have successfully shown that significant performance improvements can indeed be achieved by joint source-channel optimization. However, they are all based on some specific source and channel coder pair and thus rely heavily on detailed knowledge about the internal working mechanism of those coders. In practice, wireless multimedia links contain heterogeneous contents (audio, image, video, etc), each with its own coder, over which we may only have minimal control, such as adjusting the rate. Also, the channel coder is usually preselected by the vendor. Therefore, a general scheme which supports a wide variety of source and channel coders is highly desirable.

In this paper we propose such a scheme. The key to our approach is to perform an end-to-end optimization over both source and channel characteristics based on parametric models. The parametric models can be obtained by an online training procedure, and be applied to many source coders. In the next section we will illustrate this claim by an example, which uses Motion-JPEG video source coder and a variable Reed-Solomon channel coder.

1.2 JPEG Image Coder

The JPEG [9] 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 [10], where the bits are ordered by importance.

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

$$D = E(d) = P_{ffail} \times D_{loss} + P_{fsucc} \times D_{succ} \tag{1}$$

where D_{loss} and D_{succ} are the resulting final distortions corresponding to a failed frame transmission and a successful one respectively. P_{succ}, as defined before, is the probability of successful frame transmission.

The coding rate and resulting distortion for the JPEG coder can be related by a scalar q called the *quality-factor*; their relationship is shown in Figure 2, where the q is the source coder rate-adjusting factor; increasing q decreases the end-to-end distortion and increases the source rate.

1.3 Reed-Solomon Channel Coder and Packet-Based Approach

RS codes [11] are a well-known class of block codes with good error correction properties. We choose 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 exist 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 $m = 8$, since we usually store 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 has a fixed-length configuration, we employ a packet-based approach in channel coding, in which each transmission packet is an RS coding unit and contains both the source symbols and protection symbols. In order to calculate expected distortion, we define, for video transmission, the *frame transmission success probability* P_{fsucc} as:

$$P_{fsucc} = \prod_{n=1}^{M} P_{psucc} \tag{2}$$

$$P_{psucc} = \sum_{k=0}^{p/2} \binom{L}{k} {P_{es}}^k {P_{ss}}^{L-k} \tag{3}$$

$$P_{ss} = 1 - (1 - P_{eb})^m \tag{4}$$

$$P_{eb} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{5}$$

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, E_b is the fixed power per bit, and N_0 is the channel noise variance. In order for the above equation to be valid, sufficient interleaving must be employed so that the independence assumption is satisfied.

1.4 Rate Control

One problem unique to video transmission is the rate-control [12] [13] constraint, since the decoder usually has a finite decoding buffer-size and a constant decode-flow.

Suppose we have a video frame sequence divided into GOPs (Groups Of Pictures) of size N. Each frame is coded to have a stream length of $s(i, q_i)$, 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

$$b_i = \max(b_{i-1} + s(i, q_i) + p_i - R, 0) \tag{6}$$

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

$$b_i \le b_{max}, i = 1...N \tag{7}$$

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

2 Motion-JPEG and RS coder

We demonstrate our scheme in this paper using Motion-JPEG as the video source 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 and its ability to start decoding at any particular frame.

2.1 Problem Formulation

To formulate the problem when the source coder is Motion-JPEG, we assume that we have a series of video frames which are divided into GOPs of size N.

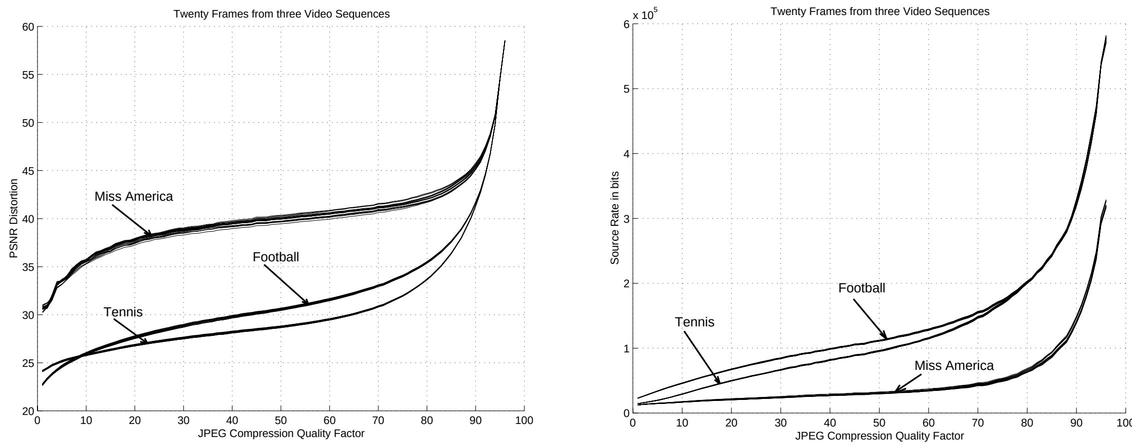

Figure 3: PSNR/Rate vs Quality-factor curves for three video sequences

We define the optimization problem as minimizing the final end-to-end distortion across two size-N vectors Q and P, where $Q = (q_1, q_2, ..., q_N)^T$ contains the quality-factors used to JPEG-encode each frame, and $P = (p_1, p_2, ..., p_n)^T$ contains the amount of protection we will use for transmitting all the packets contained in a certain frame.

However, further investigation into the problem 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 (see Figure 3). Except when there is a scene change, we can thus safely assume that almost all the frames are similar in the sense that they have identical distortion and rate functions. This leads to the natural conclusion that we can use the same quality factor and amount of protection for all the frames in the same GOP, provided that the buffer size limitation is satisfied. While this yields a slightly suboptimal solution, it is comparable in performance with the truly optimal solution but is much simpler in terms of computational complexity, since we reduced the $2N$-D optimization problem to a 2-D problem.

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

$$D = \frac{1}{N}\sum_{i=1}^{N} D(i) \tag{8}$$

$$D(i) = P_{fsucc}(p) \times d(q) + P_{ffail}(p) \times d_{conceal}(i) \tag{9}$$

$$P_{fsucc} = \prod_{n=1}^{M(q)} P_{psucc} \tag{10}$$

$$M(q) = \frac{s(q)}{L} \tag{11}$$

Here D is the final end-to-end distortion, $D(i)$ is the expected distortion for the ith frame, $P_{fsucc}(p)$ and $P_{ffail}(p)$ 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, $d_{conceal}(i)$ is the distortion when we use the $i-1$th frame in place of the ith frame, $M(q)$ is the number of packets in each frame, $s(q)$ is the frame length when coded using quality-factor q, and L is the predefined RS packet size.

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

$$b_i = max(b_{i-1} + s(q) + p - R, 0) \tag{12}$$

$$b_i \leq b_{max}, i = 1, 2, ..., N \tag{13}$$

2.2 Implementation Issues

Straightforward as the algorithm itself is, there are quite a few implementation issues remaining to be solved before the scheme becomes practical. Here we briefly mention them; for details please refer to [14].

- Obtaining the source model parameters

  We use the following method [15] [16]: first find several "control points" on the source coder parameter curves and obtain the value associated with these points, and then use piecewise polynomials

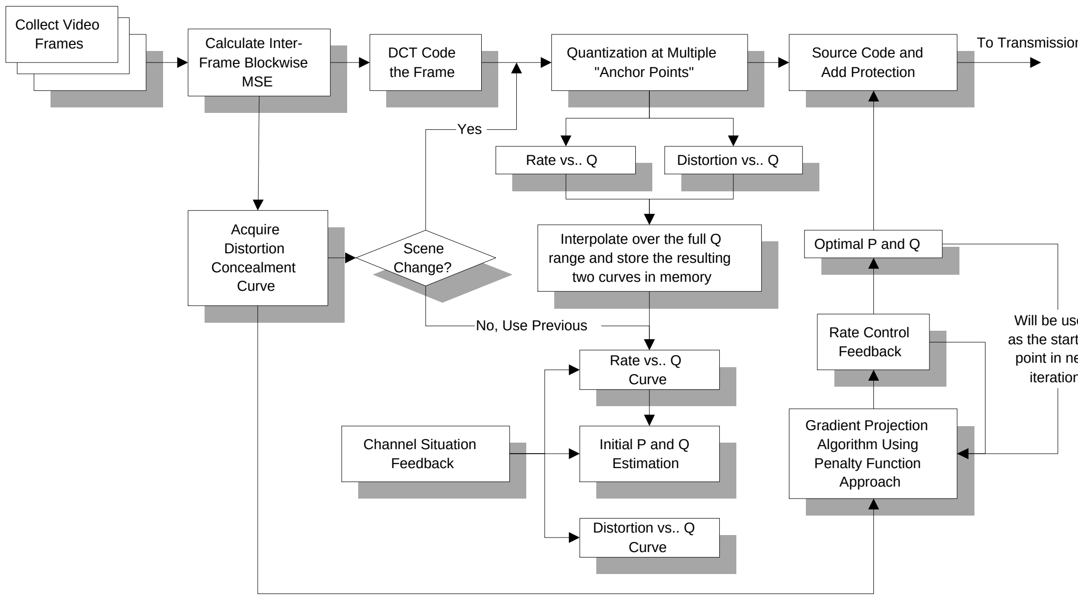

Figure 4: Real-time System Diagram

to approximate all the points in between the "control points" via interpolation. Experiments show that, if carefully chosen, only six or less points are needed to achieve a satisfactory approximation accuracy.

- Scene Change Detection

  We use the difference between frames as the detection criterion; that is, when the mean square difference between two consecutive frames exceeds a certain threshold, we say that a scene change has occurred.

- P_{psucc} (RS packet transmission success probability) approximation

  When the packet size is sufficiently large, we approximate P_{psucc} as a Gaussian random variable, thus greatly reducing the computational complexity.

- Optimization Linearization

  The optimization problem we have defined is constrained and nonlinear in nature; furthermore, the variables can only take integer values. In order to solve this problem, we transform it into an unconstrained problem. In practice we use the Penalty Function Approach [17] [18] and modify our cost function as:

$$D = \sum_{i=1}^{N} D(i) + C \times \sum_{i=1}^{N} \max(0, b_i - b_{max}) \tag{14}$$

  where the extra term is the penalty function; when C goes to infinity, this forces the solution to converge to the optimal solution to the constrained problem. In practice we use a sufficiently large number so the solution is reasonably close to the optimal one.

Figure 4 shows our proposed real-time system diagram. The system complexity is of the same order as DCT coding, and usually only no more than three iterations are needed to find the optimal soluation.

3 Results

For simulation we use the following parameters: the testing sequence is the Football sequence (which is considered a difficult sequence) in CIF format, dimension 352 by 240; we use an RS coder size of 256 symbols (2048 bits), a decoder buffer size of 21120 bits (approximately the size of two compressed

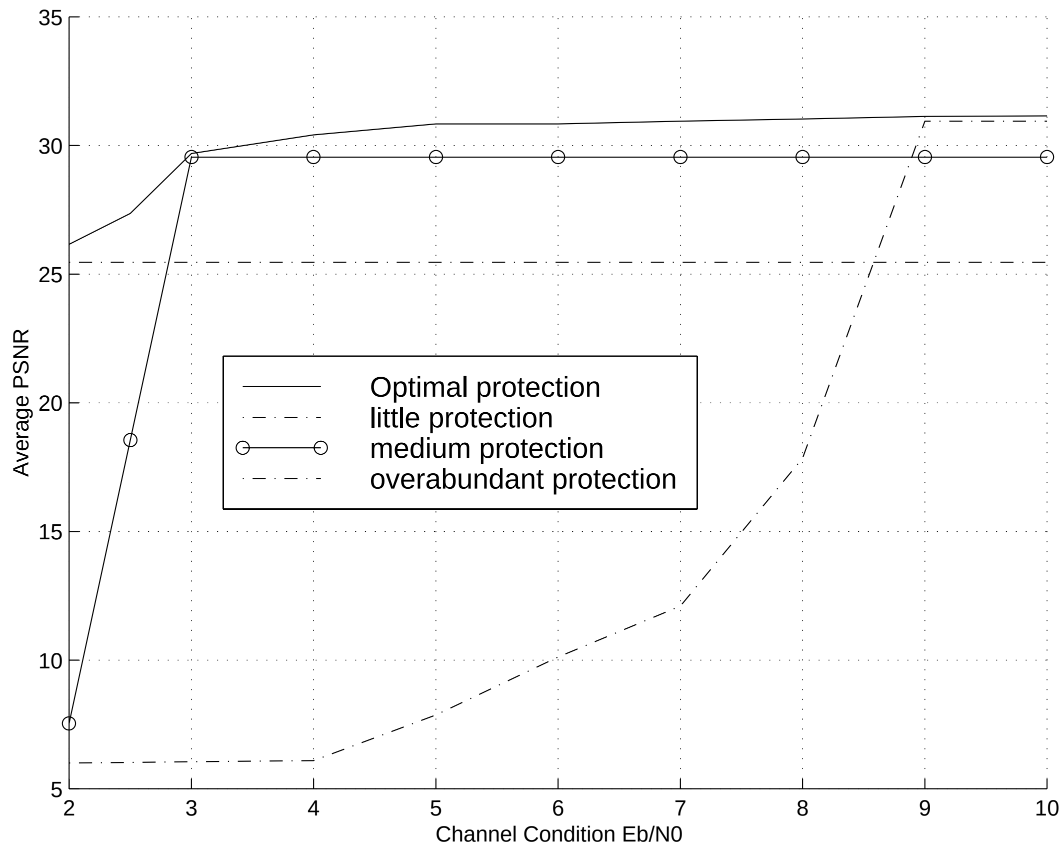

Figure 5: Simulation Results

frames), a constant decode flow of 15840 bits (approximately 1.5 times the compressed frame size), and a GOP size of 20. The result is shown in Figure 5.

The result plot shows the following:

- Fixed amount of protection (no source-channel matching)

  1. If the amount of protection is relatively low, the system behaves badly under unfavorable channel situations and becomes better only when channel SNR gets very high.
  2. If the amount of protection is unnecessarily high, the system maintains constant performance under all channel situations but can not take advantage when the channel situation is good.

- Protection determined by source-channel matching

  We can clearly see that the optimal configuration curve is always above the fixed-protection curves under all channel situations.

4 Conclusions

We have developed a general scheme to implement joint source-channel matching for wireless video transmission. The results show that our scheme makes a major difference in terms of performance in comparison with fixed-parameter systems under varying channel conditions. Our system is truly general because of our parametric-model approach, which is relatively inexpensive and can be implemented online by evaluating "control points" followed by interpolation. Thus our scheme can be applied to a large number of coders, including standard ones.

Besides the Motion-JPEG coder presented in this paper, we have also implemented our scheme using a Conditional Block Replenishment type of video source coder under a different visual quality criterion. We have also done source-channel matching for images with similiar schemes for many different channel modulations, and the results should generalize to video as well [19] [20].

For future work we plan on testing our scheme for motion-compensation-based video source coders such as MPEG and H.263, other channel coders such as RCPC coder and other channel models such as fading channel. We will also compare the results of our general scheme to those of the other schemes which are more tuned to specific source-channel coder pairs.

References

[1] S. B. Z. Azami, P. Duhamel, and O. Rioul, “Joint source-channel coding: Panorama of methods,”

[2] T. Lan and A.Tewfik, “Adaptive low power multimedia wireless communications,” *1997 Conference on Information Sciences and Systems*, 1997.

[3] G. Davis and J. Danskin, “Joint source and channel coding for image transmission over lossy packet networks,” August 1996.

[4] K. Ramchandran, A. Ortega, K. Uz, and M. Vetterli, “Multiresolution broadcast for digital HDTV using joint source/channel coding,” *IEEE Jounal on selected areas in communications*, vol. 11, Jan 1993.

[5] S. B. Z. Azami, O. Rioul, and P. Duhamel, “Performance bounds for joint source-channel coding of uniform memoryless sources using a binary decomposition,”

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

[7] G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Signal Processing Letters*, vol. 4, July 1997.

[8] H. Man, F. Kossentini, and M. Smith, “Progressive image coding for noisy channels,” *IEEE Signal Processing Letters*, vol. 4, Aug 1997.

[9] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Trans on Consumer Electronics*, Dec 1991.

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

[11] R. E. Blahut, *Digital Transmission of Information*. Addison-Wesley, 1990.

[12] C.-Y. Hsu and A. Ortega, “Rate control for robust video transmission over wireless channels,” *VCIP97*, Feb 1997.

[13] A. Ortega and M. Khansari, “Rate control for video coding over variable bit rate channels with applications to wireless transmission,” *ICIP'95*, Oct 1995.

[14] L. Qian, D. L. Jones, and K. Ramchandran, “Joint source-channel matching for wireless video transmission,” Oct 1998.

[15] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, “Cubic spline approximation of rate and distortion functions for mpeg video,”

[16] L.-J. Lin and A. Ortega, “Bit-rate control using piecewise approximated rate-distortion characteristics,” *IEEE Trans on Circuits and Systems for Video Technology*, 1998.

[17] C.-Y. Hsu and A. Ortega, “A Lagrangian optimization approach to rate control for delay-constrained video transmission over burst-error channels,” *ICASSP98*, 1998.

[18] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, “A gradient-based rate control algorithm with applications to MPEG video,” *ICIP'95*, Oct 1995.

[19] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzintsev, “Joint source-channel matching for a wirelss communications link,” *Proceedings of DCC 98*, Mar 1998.

[20] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzintsev, “Joint source-channel matching for a wirelss communications link,” *Proceedings of ICC 98*, Jun 1998.