ON ENHANCING MPEG VIDEO BROADCAST OVER WIRELESS NETWORKS WITH AN AUXILIARY BROADCAST CHANNEL

Jiajun Wang, Abhik Majumdar, and Kannan Ramchandran

University of California, Berkeley
{junewang,abhik,kannanr}@eecs.berkeley.edu

ABSTRACT

We study the problem of enhancing a baseline standards-compliant video-over-wireless broadcast system through the use of an auxiliary wireless broadcast channel. Our proposed solution is based on the information-theoretic concept of broadcast source coding which subsumes the concept of source coding with side-information. We integrate into this framework an analytically tractable mechanism for dynamically incorporating channel loss. Empirical validation of our proposed algorithm on a typical 3G wireless network broadcast channel simulator reveals its superior performance over pure FEC-based solutions by an average of 2-4 dB in PSNR.

1. INTRODUCTION

Motivated by the emergence of multicast and broadcast applications for video-over-wireless, such as streaming of television channels to cellular phones, we address the problem of real-time video broadcast over lossy wireless channels in a standards-compliant setting. While today's video standards (such as MPEG[1]) can compress video very efficiently, the compressed bit-stream is fragile to channel losses. This fragility is a direct consequence of the prediction-based coding framework that underlies MPEG, since in order to decode the current frame correctly, all previous frames need to have been received correctly. Packet losses thus lead to predictor mismatch or "drift" between encoder and decoder. This problem is exacerbated in a wireless channel where packet losses are far more frequent and bursty than in a wire-line network. In this work, we propose an algorithm, that scales with the number of users, for enhancing the delivered video quality by mitigating the effect of drift.

As shown in Figure 1, the predictive video bit-stream is broadcast over a "main" channel, and an "auxiliary" broadcast channel is available for enhancing the overall reconstructed quality. Such a set-up allows a user who does not have access to the auxiliary channel decoder to still decode the predictively coded video. Since different clients have different available bandwidths, we allow the encoder to have multiple multicast groups on the auxiliary channel with each client independently deciding the set of multicast groups that it can subscribe to based on its own constraints (see Figure 2).

The goal of this work is to find the best strategy for using the auxiliary channel to reduce the drift in a multicast setting. We pose the problem as one of source coding with side-information [1] to leverage the presence of the MPEG decoded video while encoding for the auxiliary channel. Unlike the unicast setting, multicast naturally means that different clients would have obtained different video qualities through the main MPEG channel, due to varying processing capacities, bandwidths, channel loss rates, etc. Consequently, it is not possible in general for all the receivers to be maximally satisfied. In this work, we aim to make these trade-offs between the receivers in a rate-distortion efficient manner.

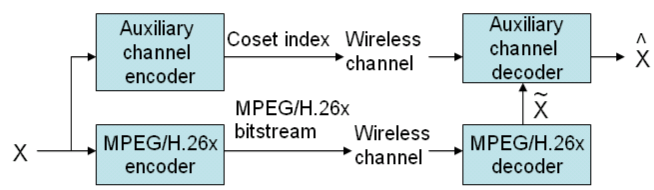

Fig. 1. auxiliary channel encoding/decoding block diagram. The predictive decoder reconstructs the input $\mathbf{X}$ as $\mathbf{Y}$. $\mathbf{Y}$ serves as side-information to the auxiliary channel decoder which outputs $\hat{\mathbf{X}}$ as the final reconstruction.

Drift reduction in a lossy unicast setting has been studied in the literature. In particular, the works of [2, 3, 4] invoke principles from source coding with side information [1] to solve this problem. However, the multi-user nature of the broadcast problem makes the problem more involved. We address this challenge by using two important concepts. The first involves broadcast source coding concepts from multi-user information theory in [5, 6]. We additionally need to incorporate the effects of losses on the broadcast channel into the above framework. To this end, we leverage concepts from [7] in order to estimate the relevant correlation structures. This allows for a quantitative framework to deal with the overall problem.

Automatic Repeat Request (ARQ) or Forward Error Correction codes (FEC) or a combination of both are often used to alleviate the problem of drift. However, delay and latency constraints of the video application may limit the use of ARQ and FEC schemes. ARQ schemes also require a feedback channel and are ill-suited to multicast/broadcast scenarios. FEC-based schemes can mitigate the probability of error but cannot guarantee error-free operation: when errors do occur, they propagate until the next intra-refresh. Even when FECs are used, therefore, there is still a need to mitigate the effects of error propagation, especially when there is a stringent latency constraint.

2. RELATED WORK

As mentioned previously, while drift reduction in a lossy unicast setting has been studied in the literature [2, 3, 4], to the best of our knowledge the problem of broadcasting over an auxiliary channel to mitigate drift has not been studied. In this section, we review

This work was funded in part by NSF Grant CCR-0330514.

[1]In this paper, by MPEG we refer to state-of-the-art predictive video coding solutions.

some techniques that have used principles from source coding with side information for video coding.

Recently, joint source-channel coding techniques based on multi-user information theory have been proposed to tackle the problem of drift [8, 2, 9, 3]. These techniques make use of the principles of source coding with side-information within a Wyner-Ziv framework [1] (or a modification of it in the case of [8]). While [8, 9] are full-fledged video codecs that eschew the predictive coding framework, [2, 3] retain the predictive coding framework but send some extra information to mitigate the drift. A scalable video coding algorithm based on distributed source coding ideas was proposed in [10]. The algorithm of [10] is similar in philosophy to MPEG4-FGS where the goal is to provide a progressive bit-stream that can be decoded at any rate (within a certain range). A similar algorithm is also proposed in [11]. However, [10, 11] do not discuss the effect of drift. [12] deploys the algorithm in [10] to provide a recipe for reliable video multicasting.

2.1. Background on Wyner-Ziv

The Wyner-Ziv Theorem [1] deals with the problem of source coding with side-information. The encoder needs to compress a source X when the decoder has access to a source Y. X and Y are correlated sources and Y is available only at the decoder. While, the results proven by Wyner and Ziv are non-constructive and asymptotic in nature, a number of constructive solutions have since been proposed wherein the source codebook is partitioned into cosets of a channel code that is matched to the correlation noise N. The number of partitions or cosets depends on the statistics of N. The encoder communicates the coset index to the decoder. The decoder then decodes to the codeword in the coset that is jointly typical with the side-information. Specifically for the problem at hand, we use the concepts described in [13] and partition the source codebook into cosets of a multilevel code (as in [14]).

3. PROPOSED ALGORITHM

In this section we present our proposed algorithm for drift reduction by sending a second description of the video over an "auxiliary channel". Let us first consider the problem when we only

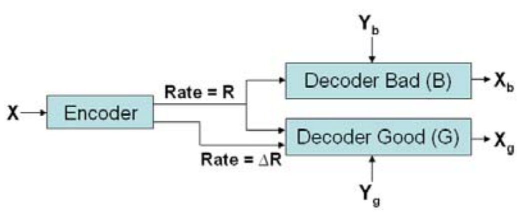

Fig. 2. System block diagram.

have 2 clients. The setup is shown in Figure 2. The "bad" decoder (later referred as Decoder B) has an auxiliary channel rate constraint of R, while the "good" decoder (Decoder G) has a constraint of $R + \Delta R$. $\mathbf{Y_b}$ and $\mathbf{Y_g}$ are the MPEG/H.26x reconstructions available to decoders B and G respectively (as shown in Figure 1). $\mathbf{Y_b}$ and $\mathbf{Y_g}$ serve as the side-information for the respective auxiliary channel decoders. Decoder G is assumed to have a better side-information due to various possible reasons as mentioned in Section 1. $\mathbf{X_b}$ and $\mathbf{X_g}$ are the final reconstructions of the source X by decoders B and G respectively. $\mathbf{X_g}$ is a better reconstruction than $\mathbf{X_b}$.

There are two aspects to our algorithm - the estimation of the correlation between the source $\mathbf{X}$ and side-informations $\mathbf{Y_b}$ and $\mathbf{Y_g}$ and the design of the codebook(s) according to the correlation statistics for a broadcast setting. In Section 3.1, we discuss the correlation estimation algorithm. In our earlier work in [4], we had developed a correlation estimation algorithm for the corresponding unicast case based on concepts from [7]. For the broadcast case, we will leverage the algorithm of [4] to find the correlation between each encoder-decoder pair. In Section 3.2 we detail the the codebook design.

3.1. Correlation Estimation for the Auxiliary Channel

Figure 1 shows the block diagram for an individual auxiliary channel encoder/decoder pair. We assume the correlation structure is $\mathbf{X} = \mathbf{Y} + \mathbf{Z}$ where $\mathbf{X}$ is the current block and $\mathbf{Y}$ is the decoder reconstruction (possibly corrupted by channel losses) that serves as side-information for the auxiliary channel decoder. By operating in the DCT domain, we are able to make the simplifying assumption that the components of the correlation noise vector $\mathbf{Z}$ are independent. We model the components as Gaussian distributed. As described in Section 2.1, the auxiliary channel encoder needs to find the number of partitions of the source codebook (i.e. the number of cosets) for each component in $\mathbf{X}$, for which it needs to know the variances of each component of $\mathbf{Z}$.

For both the main and auxiliary channels, we assume an independent packet erasure channel and that packets are independently decodable. When a block is lost it is replaced by the block in the same position in the previous frame[2]. The packet loss probability on the main and auxiliary channels are p and q respectively.

Following the notation of [7], let $g_n^{i,k}$ be the original value of the i^{th} DCT coefficient in the k^{th} block of the n^{th} frame and let $\hat{g}_n^{i,k}$ denote its encoder reconstruction, i.e. $\hat{g}_n^{i,k}$ is the quantized representation of $g_n^{i,k}$. Let this coefficient be re-constructed as $\tilde{g}_n^{i,k}$ by the predictive decoder. $\tilde{g}_n^{i,k}$ is a random variable for the encoder. Let us first look at the situation when there is no auxiliary channel. There are two cases - the block is either intra or inter (predictive) coded.

If the block was intra-coded then:

$$\tilde{g}_n^{i,k} = \begin{cases} \hat{g}_n^{i,k} & : \; w.p. \;\; 1-p \\ \tilde{g}_{n-1}^{i,k} & : \; w.p. \;\; p \end{cases} \tag{1}$$

If the block was inter(predictive)-coded then:

$$\tilde{g}_n^{i,k} = \begin{cases} \tilde{g}_{n-1}^{i,j} + \hat{e}_n^{i,k} & : \; w.p. \;\; 1-p \\ \tilde{g}_{n-1}^{i,k} & : \; w.p. \;\; p \end{cases} \tag{2}$$

where $\hat{e}_n^{i,k} = \hat{g}_n^{i,k} - \hat{g}_{n-1}^{i,j}$. $g_{n-1}^{i,j}$ is the predictor for $g_n^{i,k}$, i.e. the best predictor for the k^{th} block in the n^{th} frame is the j^{th} block in the $(n-1)^{th}$ frame. Note that even when the MPEG stream gets through for a particular block, $\tilde{g}_n^{i,k}$ may not equal $\hat{g}_n^{i,k}$ since there may have been previous errors.

Now for designing the auxiliary channel encoder, we look at the particular case when the predictive bitstream does get through for the current block, but the block may nonetheless be in error due to previous errors. Specifically the expected distortion for this case is:

$$d_n^{i,k} = E[(g_n^{i,k} - (\tilde{g}_{n-1}^{i,j} + \hat{e}_n^{i,k}))^2] \tag{3}$$

We can calculate $d_n^{i,k}$ for all the DCT coefficients and code for this correlation noise at the auxiliary channel encoder. After the DCT

[2] Any other error concealment scheme may also be used.

coefficients are refined using the auxiliary channel, the reconstruction at the decoder will be:

$$\tilde{g}_n^{i,k} = \begin{cases} \hat{g}_n^{i,k} & : \quad w.p. \quad (1-p)(1-q) \\ \tilde{g}_{n-1}^{i,j} + \hat{e}_n^{i,k} & : \quad w.p. \quad q(1-p) \\ \tilde{g}_{n-1}^{i,k} & : \quad w.p. \quad p \end{cases} \tag{4}$$

To compute $d_n^{i,k}$ using (3), we need to compute $E[\tilde{g}_{n-1}^{i,j}]$ and $E[(\tilde{g}_{n-1}^{i,j})^2]$ (since $\hat{e}_n^{i,k}$ and $g_n^{i,k}$ are already known at the encoder). The auxiliary channel encoder can compute $E[\tilde{g}_{n-1}^{i,j}]$ and $E[(\tilde{g}_{n-1}^{i,j})^2]$ using equations (1), (2), and (4) in a simple recursive algorithm (similar to that used by [7]) and thus compute $d_n^{i,k}$ using (3). For further details, please refer to [4].

In the multicast/broadcast scenario, $E[\tilde{g}_{n-1}^{i,j}]$ and $E[(\tilde{g}_{n-1}^{i,j})^2]$ need to be modified according to each decoder's specific channel condition (represented by different packet loss probabilities p and q) and auxiliary channel rate available. The number of DCT coefficients to be refined may also vary. The encoder will need to separately maintain $E[\tilde{g}_{n-1}^{i,j}]$ and $E[(\tilde{g}_{n-1}^{i,j})^2]$ for the good and bad decoders. Note that we only need to keep running estimates of the correlation statistics for each client. No deterministic information regarding the exact realizations of the decoded frames of different clients needs to be kept. This leads to only a small increase in the amount of state information maintained at the server which allows our algorithm to scale with the number of clients.

3.2. Broadcast Coding Algorithm for the Auxiliary Channel

We now discuss the codebook design for the setup of Figure 2. We assume that $\mathbf{X} \rightarrow \mathbf{Y_g} \rightarrow \mathbf{Y_b}$ forms a Markov chain which implies that decoder G's side-information is a degraded version of decoder B's. Figure 3 shows a qualitative representation for the achievable distortions for decoders B and G for a fixed rate. In the interests of simplicity, we will restrict ourselves to the important operating point where the entire rate R can be utilized by decoder B.

The correlation noise ($d_n^{i,k}$), estimated as described in Section 3.1, determines the required number of partitions of the source codebook or number of cosets and hence the rate required to refine each coefficient. Since the rate on the auxiliary channel is very limited, an inverse waterfilling strategy [15] is used to minimize the (expected) end-to-end distortion. In practice, we approximate the inverse waterfilling strategy by refining the lowest frequency DCT coefficients first within the allowed rate[3] as detailed in [4].

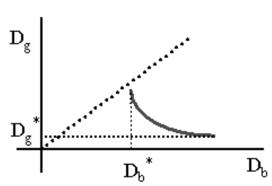

Fig. 3. Qualitative curve representing the achievable distortions for Decoders B and G for a fixed rate. D_b and D_g are the achievable distortions for Decoders B and G respectively.

Having determined the number of partitions of the source codebook, we generate two codebooks C_1, and C_2 of rates R and ΔR respectively. Since the entire rate R can be utilized by decoder B, we use the correlation noise estimates for decoder B to design C_1, while the correlation noise estimates for decoder G are used to design C_2. The source $\mathbf{X}$ is quantized using C_1 and C_2 to generate the codewords U and W respectively. Conceptually the decoding process is as follows: the codeword U is first decoded by both decoders. At this point, decoder B obtains its final reconstruction $\mathbf{X_b}$. After decoding U, decoder G decodes W using $(U, \mathbf{Y_g})$ as the side-information to obtain its final reconstruction $\mathbf{X_g}$. It was shown in [5, 6] that this strategy achieves the optimal rate-distortion point. The entire optimal rate-distortion region for this problem is provided in [6].

This multicast system can be implemented by outputting the "base" rate R and the "enhancement" rate ΔR bit-streams on two multicast ports. Clients can choose to join either only the first multicast group (thus receiving only rate R) or both multicast groups (thus receiving the full rate $R + \Delta R$).

Multiple users: The extension to more than 2 users is relatively straight-forward. Suppose there is a third client in the system with a rate constraint of $R + \Delta R_1 + \Delta R_2$. Then we will encode the R and ΔR_1 bit-streams just like in the two-client case while the new ΔR_2 bit-stream will be coded keeping in mind the better reconstruction that the third client has after it has decoded the R and ΔR_1 bit-streams.

4. EXPERIMENTAL RESULTS

In this section, we present simulation results that demonstrate the robustness features of our proposed approach and its suitability for the multicast scenario. We use a H.263+ coder[4] as the predictive video codec. For our tests we use a wireless channel simulator obtained from Qualcomm, Inc. This simulator adds packet errors to multimedia data streams transmitted over wireless networks conforming to the CDMA 2000 1X standard[5].

The experimental setup is as in Figure 2. The rate on the main channel is R_m, the auxiliary channel available to Decoders B and G is of rates R and $R + \Delta R$ respectively. We compare the following 3 scenarios for both decoders: **1) H.263+** encoded at a rate equal to the total allocated to the main and auxiliary channels, **2) Full auxiliary channel rate allocated to FEC** (we use Reed-Solomon codes), and **3) Proposed algorithm**, modified by allocating part of the rate on the auxiliary channel to FEC (RS codes used). For both (2) and (3) the latency constraint is 1 frame.

The first set of tests was run on the football sequence (352×240, 15 fps, GOP size 30) where $R_m = 900$kbps, $R = 20\%$ of R_m and $R + \Delta R = 25\%$ of R_m[6]. The loss rate for the "bad" decoder was 8.5% and 4.7% for the "good" one[7]. As shown in Figure 4(a) and (b), our proposed algorithm is able to recover from packet losses rapidly unlike the FEC-only solution. For the good decoder, the proposed algorithm achieves an average PSNR that is 4.5 dB higher than that of the FEC-only approach. For the bad decoder, the proposed algorithm achieves an average PSNR that is 3.7 dB higher than that of the FEC-only approach. Figure 4(a) also shows the corresponding packet drop pattern of the simulated channel, which shows that FEC performs poorly in the event of

[3]The rate allowed on the auxiliary channel is determined through a rate control algorithm. We omit the description of the rate control algorithm due to lack of space.

[4]Free version of H.263+ obtained from Univ. of British Columbia.

[5]The packet error rates are determined by computing the carrier to interference ratio of the cellular system.

[6]Evolving standards for video over cellular networks (such as 3GPP) typically allocate extra rate of about 25% for FECs and/or other error correcting mechanisms.

[7]Note that this is a bursty channel and these are only average error rates

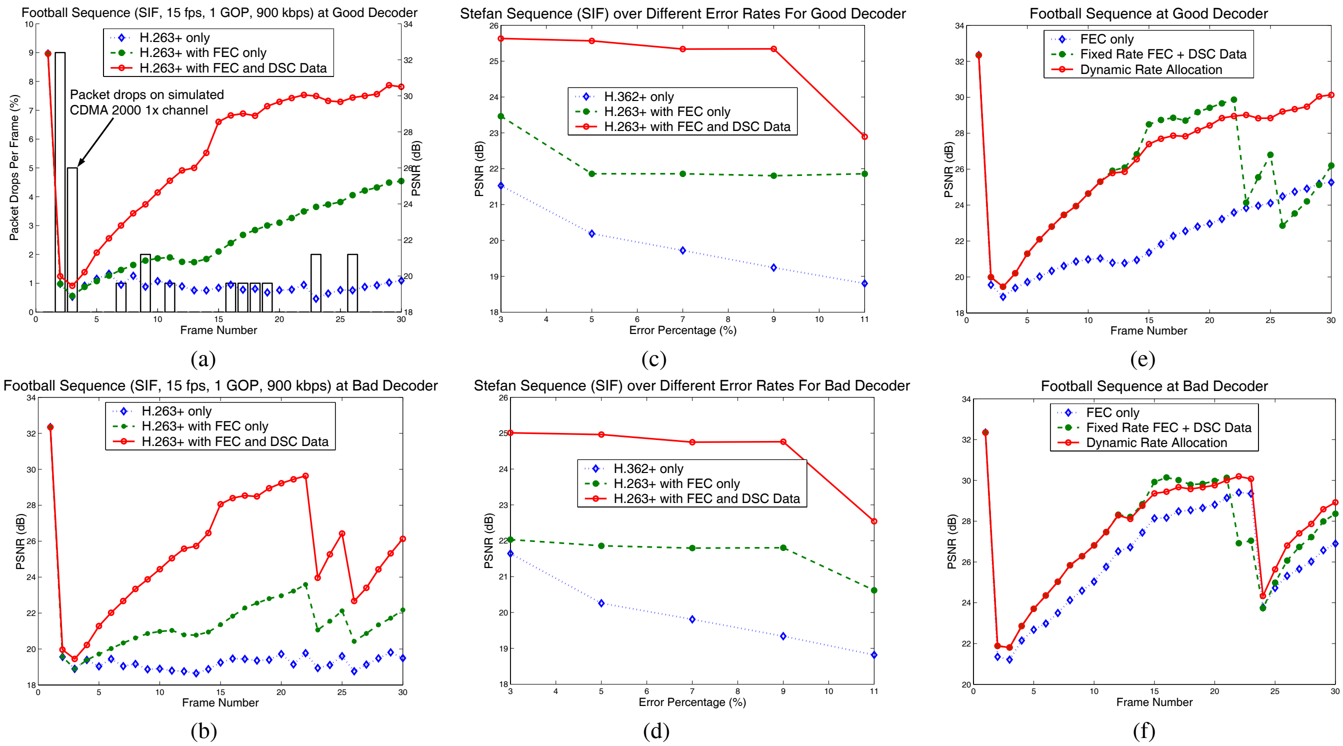

Fig. 4. Performance comparison of the 3 different schemes for the "good" and "bad" decoders, where main channel rate (R_m) is 900 kbps, auxiliary channel rates are $R = 180$kbps and $\Delta R = 60$kbps. (a) and (b): Football, loss rate of 4.7% and 8.5% for "good" and "bad" decoders respectively. (a) also shows the packet error pattern of the channel. (c) and (d): Performance comparisons over different packet drop rates for "good" and "bad" decoders. (e) and (f): Performance comparisons when dynamic rate allocation is used for "good" and "bad" decoders. (All results obtained using simulated CDMA 2000 1x channel.)

a burst of errors under a low latency constraint. Figure 4(c) and (d) show the typical performance of the 3 methods over a range of packet loss rates for Stefan sequence (352×240, 15 fps, GOP size 30) for both "good" (25% extra rate) and "bad" (20% extra rate) decoders. It should be noted that the performance of our proposed algorithm is highly correlated to the burst length of packet drops as pointed out in [4].

While the above results illustrate that distributed coding can effectively correct for errors, it cannot prevent drift from occurring. On the other hand, FEC can prevent drift from happening but cannot correct for errors once the number of packet drops exceeds its correction capacity. Hence an effective strategy for any client would be to switch between FEC and distributed coded data on an as-needed basis. This can be achieved by serving FEC and distributed coded data in separate multicast groups. Each client can then subscribe to only the FEC group(s) before any drift occurs. Once drift does occur, it would switch to the distributed coded data group(s). After the drift has been sufficiently corrected, it can switch back to subscribing to only the FEC group(s). Figure 4(e) and (f) shows the efficacy of such a dynamic switching approach over a fixed allocation between FEC and distributed coded data (and of course it is better than using FECs only).

5. CONCLUSIONS & FUTURE WORK

We have designed and implemented a standards-compliant video-over-wireless multicast/broadcast system that scales with the number of users and caters to the need of heterogeneous receivers based on multi-user information theoretic principles of broadcast source-coding. Our ongoing work focuses on finding an optimal receiver-driven strategy for dynamically switching between FEC and distributed coded data multicast groups.

6. REFERENCES

[1] A. D. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, pp. 1–10, Jan 1976.

[2] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-ziv Coding of Video: An Error-Resilient Compression Framework," *IEEE Trans. on Multimedia*, vol. 6, pp. 249–258, 2004.

[3] A. Aaron, S. Rane, D. Rebollo-Monedero, and B. Girod, "Systematic Lossy Forward Error Protection for Video Waveforms," in *Proc. IEEE Int. Conf. Image Proc.*, 2003.

[4] A. Majumdar, J. Wang, K. Ramchandran, and H. Garudadri, "Drift reduction in predictive video transmission using a distributed source coded side-channel," in *Proc. ACM Multimedia*, 2004.

[5] C. Heegard and T. Berger, "Rate distortion when side information may be absent," *IEEE Trans. Inf. Theory*, vol. 31, pp. 727–734, Nov 1985.

[6] Y. Steinberg and N. Merhav, "On Successive Refinement of the Wyner-Ziv Problem," *Submitted to IEEE Trans. Inf. Theory*.

[7] R. Zhang, S. L. Regunathan, and K. Rose, "Optimal intra/inter mode switching for robust video communication over the Internet," in *Proc. 33rd Ann. Asilomar Conf. on Sig. Syst. Comp.*, 1999.

[8] R. Puri and K. Ramchandran, "PRISM: A New Robust Video Coding Architecture based on Distributed Compression Principles," in *Allerton Conference on Communication, Control and Computing*, 2002.

[9] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv Coding of Video: Applications to compression and error resilience," in *Proc. IEEE Data Compression Conf.*, 2003.

[10] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Proc. of VCIP'04*, Jan. 2004.

[11] H. Wang and A. Ortega, "WZS: Wyner-Ziv scalable predictive video coding," in *Proc. Picture Coding Symposium (PCS)*, 2004.

[12] Q. Xu, V. Stankovic, and Z. Xiong, "Wyner-ziv video compression and fountain codes for receiver-driven layered multicast," in *Proc. Picture Coding Symposium (PCS)*, Dec. 2004.

[13] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. on Information Theory*, March 2003.

[14] A. Majumdar, J. Chou, and K. Ramchandran, "Robust Distributed Video Compression based on Multilevel Coset Codes," in *Proc. Asilomar*, Nov. 2003.

[15] S. S. Pradhan and K. Ramchandran, "Enhancing analog image transmission systems using digital side information: a new wavelet based image coding paradigm," in *Proc. IEEE Data Compression Conf.*, 2001.