# VIDEO MULTICAST OVER LOSSY CHANNELS BASED ON DISTRIBUTED SOURCE CODING

*Abhik Majumdar and Kannan Ramchandran**

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

## ABSTRACT

We present an algorithm for robust scalable video multicast based on distributed source coding techniques. Unlike prediction based coders, like MPEG, the proposed framework directly addresses the problem of drift due to packet losses. Building on the recently proposed PRISM video coding framework [1], we show that substantial gains are possible for video multicast over lossy channels as compared to standard codecs, without a dramatic increase in encoder complexity as the number of streams increases.

## 1. INTRODUCTION

Motivated by emerging applications in multimedia over wireless networks, this papers addresses the robust scalable video multicast problem. More specifically, we focus on the problem of designing a single video bit-stream to serve multiple clients over a wireless network with varying transmission and loss rates. At a high level, it is desirable for the codec to have the following requirements: **(1) Robustness** to packet drops, **(2) SNR scalability** for a finite number of target rates, **(3) Minimal compression hit** over unicast solution, **(4) Lack of "state explosion"** with number of targeted rates.

A significant amount of work has been done towards addressing concerns (1)-(4) and today's video coders such as H.263+, MPEG4-FGS (fine-grain-scalability), H.264, 3-D wavelet based scalable video coders [2, 3], etc. all support the use of such bit-streams. However, prediction based coders (like H.26x and MPEG) are inherently fragile in the face of channel loss since the loss of the predictor leads to errors in the subsequent frames. Packet losses lead to loss of synchronization or "drift" between encoder and decoder. The delay and latency constraints of the video application may further limit the use of ARQ and FEC schemes to recover from loss. Moreover, with FEC-based schemes one cannot guarantee that the data will surely be received and if it is not, the errors will again propagate till the next intra-frame is received. Hence, despite the fact that there are a number of innovative schemes for error resilience and error concealment for H.26x and MPEG coders, the predictive coding technique is itself somewhat ill-suited to the especially harsh conditions of a wireless channel.

While drift is the major problem with predictive coding frameworks, there is some tension between constraints (2) and (3) as well. MPEG4-FGS [4, 5] provides a near continuum of scalability in bit rate but potentially takes a big compression performance hit [6], in part due to its attempt to satisfy constraint (4) by maintaining a single predictive loop at the encoder (corresponding to the base-layer quality) to serve all bit-streams. Efficient predictive coding frameworks that need to target multiple coding rates necessarily have to keep multiple prediction loops at the encoder. Indeed, the H.263+ [7] coder uses a multiple loop structure, taking into account the presence of different quality predictors at different decoders. Consequently H.263+ bit-streams suffer less of a hit in terms of loss over the non-scalable case [8]. However, the multiple loop structure leads to added complexity and limits the number of possible rates at which the stream can be decoded [2].

The rigid computational complexity partition of predictive codecs (with the encoder performing the entire motion compensation task) ties the decoding of a video block to a single predictor. If it were possible for part of the motion compensation task to be moved to the decoder where any one of multiple predictors could be used to decode the video block then the robustness of the codec would be greatly enhanced (since decoding would no longer be tied to a specific predictor). Recently, the PRISM framework [1] was proposed for video coding based on the principles of distributed source coding. PRISM allows for a part (or all) of the motion compensation task to be moved to the decoder. The PRISM codec is inherently robust to losses in the bit-stream and significantly outperforms standard video coders, such as H.263+ for transmission over packet loss channels [9]. Other video coders based on distributed source coding techniques and exhibiting such error resilience properties have also been proposed [10, 11].

Building on the PRISM framework we propose a novel scalable video coding algorithm based on the principles of distributed video coding. The algorithm takes into account that different decoders may have predictors of different qual-

*This work was funded in part by a grant from Qualcomm Corp.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

ity available. This scalable video coder is designed specifically to provide good performance in the face of channel losses. **In particular, we will focus on the case when most of the motion compensation task is performed at the decoder.** This is of particular relevance to emerging "uplink" multimedia applications (such as users multicasting video from their cellphones).

Recently, a scalable video coding algorithm based on distributed source coding ideas was proposed in [12]. The algorithm of [12] is similar in philosophy to MPEG4-FGS and the goal seems to be to provide a progressive bit-stream that can be decoded at any rate (within a certain range). On the other hand, our objective is to solve the multicast problem where different clients can receive at specific rates.The major distinction between our work and that of [12] is that, in what is to follow in Figure 1, [12] is restricted to the special case of $Y_b = Y_g$. Further, [12] does not discuss the effect of drift (which is our main focus).

## 2. BACKGROUND ON PRISM

The PRISM video coder is based on the principle of source coding with side information. The Wyner-Ziv Theorem [13] 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. From information theory we know that for the MSE distortion measure and $X = Y + N$ where $N$ has a Gaussian distribution, the rate - distortion performance for coding $X$ is the same whether or not the encoder has access to $Y$ [13, 14].

For the problem of source coding with side information, the encoder needs to encode the source within a distortion constraint, while the decoder needs to be able to decode the encoded codeword subject to the correlation noise (between the source and the side-information). While, the results proven by Wyner and Ziv are non-constructive and asymptotic in nature, a number of constructive methods to solve this problem have since been proposed wherein the source codebook is partitioned into cosets of a channel code.

For the PRISM video coder [1], the video frame to be encoded is first divided into non-overlapping spatial blocks of size 8x8. The source $\mathbf{X}$ is the current block to be encoded. The side-information $\mathbf{Y}$ is the best (motion - compensated) predictor for $\mathbf{X}$ in the previous frame and let $\mathbf{X} = \mathbf{Y} + \mathbf{N}$. We first encode $\mathbf{X}$ in the intra-coding mode to come up with the quantized codeword for $\mathbf{X}$. Now, we do the syndrome encoding, i.e., we find a channel code that is matched to the "correlation noise" $\mathbf{N}$, and use that to partition the source codebook into cosets of that channel code. The encoder transmits the syndrome (indicating the coset for $\mathbf{X}$) and a CRC check of the quantized sequence. In contrast to MPEG, H.26x, etc., it is the decoder's task to do motion search, as it searches over the space of candidate predictors one-by-one to decode a sequence from the set labeled by the syndrome. When the decoded sequence matches the CRC check, decoding is declared to be successful. For further details please refer to [15].

**Robustness Characteristics of PRISM** : The key aspect here is that PRISM does not use the exact realization of frame $(N-1)$ while encoding blocks in frame $N$, but only the correlation statistics. Note that if the decoder does not have frame $(N-1)$ (or a part of it) due to channel loss, it might still have blocks from frame $(N-2)$. If the correlation noise between any of these blocks and the current block is within the noise margin for which the syndrome code was designed the current block can be decoded.

## 3. PROPOSED SCALABLE VIDEO CODING ALGORITHM

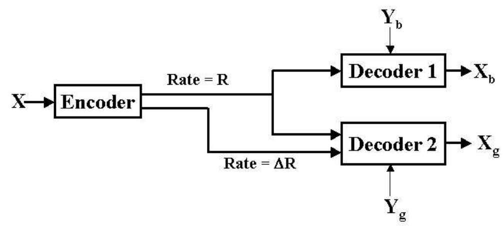


**Fig. 1**. System block diagram

Let us first consider the problem when we only have 2 clients. The set-up is as shown in Figure 1. Decoder 1 has a rate constraint of $R$, while decoder 2 has a rate constraint of $R + \Delta R$. $\mathbf{Y_b}$ and $\mathbf{Y_g}$ are the predictor blocks (from previous frame(s)) available to decoders 1 and 2 respectively. $\mathbf{Y_b}$ and $\mathbf{Y_g}$ form the side-informations for the respective decoders. Since decoder 2 receives data at a higher rate, it will have a better predictor (and hence the better side-information) than the decoder 1. $\mathbf{X_b}$ and $\mathbf{X_g}$ are the reconstructions of the source $X$ by decoders 1 and 2 respectively. $\mathbf{X_g}$ is a better reconstruction than $\mathbf{X_b}$.

Heegard and Berger [16] provided the optimal rate-distortion region for this problem for the case when $\Delta R = 0$. Steinberg and Merhav [17] have recently extended the result of [16] to cover the case of non-zero $\Delta R$, where $\mathbf{X} - \mathbf{Y_g} - \mathbf{Y_b}$ forms a Markov chain. The Markov chain implies that the lower rate user's side-information is a degraded version of the better user's. The entire optimal rate-distortion region for this problem is provided in [17]. In the interests of simplicity, we will restrict ourselves to one important operating point in this region. This point corresponds to the case where the entire rate R can be utilized by decoder 1.

The solution for this case calls for the generation of 2 codebooks $C_1$, and $C_2$. The rate of codebook $C_1$ is $R$ while that of $C_2$ is $\Delta R$. The source $X$ 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. $\mathbf{X_b}$ is the reconstruction by decoder 1 and let $\mathbf{X'_g}$ be the reconstruction by decoder 2. $\mathbf{X'_g}$ is a better reconstruction of $X$ than $\mathbf{X_b}$ due to greater estimation gains (because of the presence of the better side-information at decoder 2)[1]. Now, the codeword $W$ is decoded using $\mathbf{X'_g}$ as the side-information. Note that it would be sub-optimal here to assume that the reconstruction by decoder 2 is also $\mathbf{X_b}$ and we get a rate rebate by using the better reconstruction $\mathbf{X'_g}$ . In the multicast scenario, this system can be implemented by outputting the "base" rate $R$ and the "enhancement" rate $\Delta R$ bit-streams on two multicast ports. Clients can either choose to join only the first multicast group (thus receiving only rate $R$) or they can choose to join both multicast groups (thus receiving the full rate $R + \Delta R$).

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

Unlike the H.263+ encoder, our encoder needs to maintain a relatively small amount of "state" information relating to the statistical correlation between the current frame and the different predictors at the decoders. While details depend on the exact implementation (e.g., a single scalar quantity representing the estimated correlation noise might suffice), the key difference is that in the predictive coding framework, *deterministic* copies of each predictor frame need to be kept in the encoder state. This allows our algorithm to scale with the number of users.

### 3.1. Coding Strategy

As mentioned above, Wyner-Ziv based side-information coding requires partitioning a source codebook into good channel codes. In our earlier work of [9], we had presented a novel source coding strategy for PRISM where the source codebook is partitioned into cosets of a multilevel code. We present a short description of the encoding and decoding strategies.

Consider an $m$-level partition (see Figure 2) of a lattice. At each level $i$, a subcodebook is completely determined by a bit, $B_i$, for that level and $i-1$ bits from previous levels, $B_j; 1 \leq j \leq i-1$. Encoding may then proceed by by first quantizing $X$ to the closest point in the lattice at level 0, and then determining the path through the partition tree to the subcodebook at level $m$, that contains the codepoint representing $X$. The path will specify the source

[1] Note that this estimation gain comes from multiple independent looks at the source data [13].

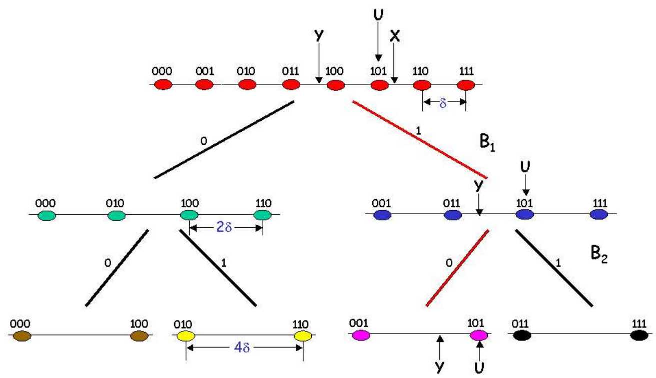


**Fig. 2**. Partitioning the Integer lattice into 3 levels. $X$ is the source, $U$ is the (quantized) codeword and $Y$ is the side-information. The number of levels in the partition tree depends on the effective noise between $U$ and $Y$ given $X$.

bits, $B_i; 1 \leq i \leq m$, that are to be transmitted to the decoder. The number of levels in the partition tree can be varied based on the effective noise between $U$ and $Y$ given $X$.

It can be shown that the Wyner-Ziv problem can be decomposed into $m$ parallel Wyner-Ziv problems where each sub-problem requires its own separate encoding rate. In fact, each level can be treated as a one level partition and a separate encoder and decoder can be designed to achieve a desired rate for that level. This allows us to perform syndrome coding with linear error correction codes at each level to compress the bits at that level down to the desired rate. As in [18, 19], we can perform multistage decoding at the decoder to recover codeword used by the encoder.

## 4. SIMULATION RESULTS

| Sequence | Codec Used | R kbps | ΔR kbps | Unicast | | Multicast | | Hit for Multicast |
|---|---|---|---|---|---|---|---|---|
| | | | | Dec.1 | Dec. 2 | Dec.1 | Dec.2 | |
| **Football** (176x128 15fps) | PRISM | 327 | 197 | 30.83 | 34.79 | 30.83 | 34.48 | 0.31 |
| | H.263+ | 327 | 197 | 31.97 | 34.57 | 31.97 | 33.76 | 0.81 |
| | PRISM | 311 | 192 | 30.45 | 34.41 | 30.45 | 34.09 | 0.32 |
| | H.263+ | 311 | 192 | 31.72 | 34.34 | 31.72 | 33.51 | 0.83 |
| **Stefan** (176x144 15 fps) | PRISM | 720 | 103 | 33.67 | 35.20 | 33.67 | 35.12 | 0.08 |
| | H.263+ | 720 | 103 | 37.23 | 38.36 | 37.23 | 37.63 | 0.73 |
| | PRISM | 795 | 80 | 34.95 | 35.29 | 34.95 | 35.23 | 0.06 |
| | H.263+ | 795 | 80 | 38.04 | 38.85 | 38.04 | 38.4 | 0.45 |

**Fig. 3**. Compression Performance Results in terms of PSNR (dB) for the 2 client/2 rate case. "Dec.1" & "Dec.2" refer to Decoder 1 & Decoder 2 respectively. As in Fig 1, Dec. 1 gets rate $R$ and Dec. 2 gets rate $R + \Delta R$. The unicast case is when the encoder sends two separate bit-streams to decoders 1 and 2.

In this section we present some preliminary results that showcase the promise of our approach. We present results for the two client/two rate case using the "uplink" PRISM framework (i.e. one in which motion compensation is performed at the decoder) and compare it to the H.263+ coder[2]

[2] Free Version of H.263+ obtained from Univ. of British Columbia.

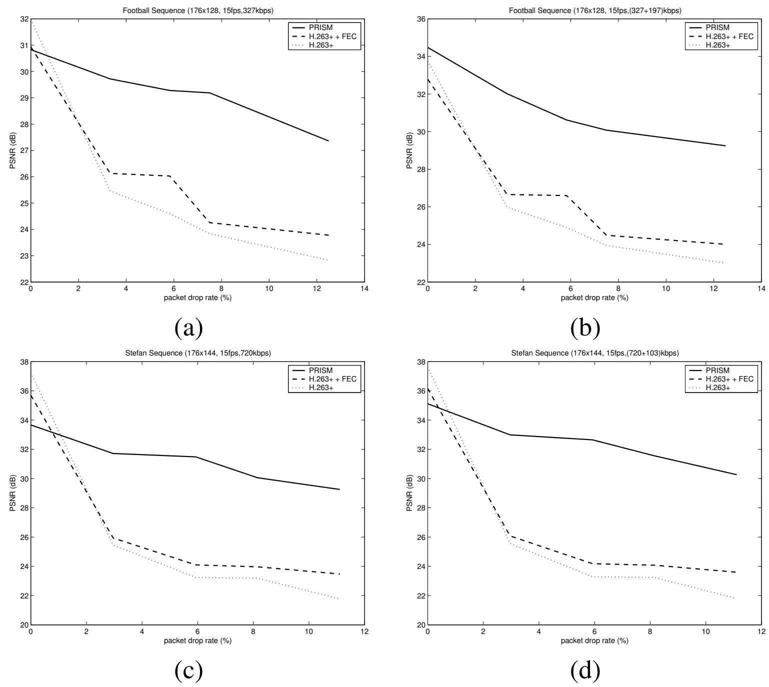


**Fig. 4**. Performance comparison (for Multicast) of scalable PRISM, H.263+, and H.263+ protected with FECs (Reed-Solomon (RS) codes used, 20% of the total rate used for parity bits) for the Football & Stefan sequences. (a) Football at Decoder 1 (rate $R = 327$kbps) (b) Football at Decoder 2 (rate $R + \Delta R = 524$kbps) (c) Stefan at Decoder 1 (rate $R = 720$kbps) (d) Stefan at Decoder 2 (rate $R + \Delta R = 823$kbps). For the FEC case, protection was given only to the base layer.

in enhancement mode. For our tests, we restrict ourselves to the case when the entire rate $R$ can be utilized by the lower rate client (Decoder 1). The results are shown in Figure 3. As can be seen from the last column of Figure 3, the scalable PRISM codec suffers a smaller hit in compression efficiency than H.263+ over their respective unicast cases.

We also tested our scheme using a channel simulator (obtained from Qualcomm, Inc.) for wireless networks conforming to the CDMA 2000 1X standard. As can be seen from Figure 4, the scalable PRISM codec is superior to scalable H.263+ as well as scalable H.263+ protected with FECs (RS codes used, 20% of total rate used for parity bits) by a very wide margin (5-8 dB). This verifies that the scalable PRISM architecture is well suited to the wireless multicast problem.

## 5. CONCLUSIONS AND FUTURE WORK

We have presented a robust scalable video coding scheme by employing the principles of distributed source coding. Currently, we are in the process of building a downlink version of the PRISM codec, designing the uplink coder to work efficiently at lower rates and running extensive tests to further validate our approach.

## 6. REFERENCES


[1] 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.

[2] S.-T. Hsiang and J. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," in *Elsevier Science, Sig. Proc. Journal: Image Communications*, vol. 16, pp. 705–724, May 2001.

[3] D. Taubman and A. Secker, "Highly scalable video compression with scalable motion coding," in *Proc. of ICIP'03*, Sept. 2003.

[4] W. Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Trans. on CSVT*, vol. 11, pp. 301–317, Mar 2001.

[5] *Final draft amendment 4, ISO/IEC JTC1/SC29/WG11, MPEG00/N33904*, 2001.

[6] W. Li, F. Ling, and X. Chen, "Fine granularity scalability in MPEG-4 for streaming video," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, 2000.

[7] *Draft Text of H.263 Version 2 (H.263+)*.

[8] M. Gallant and F. Kossentini, "Efficient scalable DCT-based video coding," in *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering*, May 1999.

[9] A. Majumdar, J. Chou, and K. Ramchandran, "Robust Distributed Video Compression based on Multilevel Coset Codes," in *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2003.

[10] A. Sehgal and N. Ahuja, "Robust Predictive Coding and the Wyner-Ziv Problem," in *Proc. DCC*, 2003.

[11] 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.

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

[13] 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.

[14] S. S. Pradhan, J. Chou, and K. Ramchandran, "A characterization of functional duality between source and channel coding," in *Proc. of Intl. Symp. on Information Theory (ISIT)*, 2002.

[15] R. Puri and K. Ramchandran, "A video coding architecture based on distributed compression principles," in *ERL Technical Report, Memorandum No. UCB/ERL M03/6*, Mar 2003.

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

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

[18] U. Wachsmann, R. Fischer, and J. Huber, "Multilevel Codes: Theoretical Concepts and Practical Design Rules," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1361–1391, July 1999.

[19] H. Imai and S. Hirakawa, "A New Multilevel Coding Method using Error-Correcting Codes," *IEEE Trans. Inf. Theory*, vol. 23, pp. 371–377, May 1977.