Wyner–Ziv Coding of Video: An Error-Resilient Compression Framework

Anshul Sehgal, Ashish Jagmohan, and Narendra Ahuja, *Fellow, IEEE*

***Abstract*—This paper addresses the problem of video coding in a joint source-channel setting. In particular, we propose a video encoding algorithm that prevents the indefinite propagation of errors in predictively encoded video—a problem that has received considerable attention over the last decade. This is accomplished by periodically transmitting a small amount of additional information, termed coset information, to the decoder, as opposed to the popular approach of periodic insertion of intra-coded frames. Perhaps surprisingly, the coset information is capable of correcting for errors, without the encoder having a precise knowledge of the lost packets that resulted in the errors. In the context of real-time transmission, the proposed approach entails a minimal loss in performance over conventional encoding in the absence of channel losses, while simultaneously allowing error recovery in the event of channel losses. We demonstrate the efficacy of the proposed approach through experimental evaluation. In particular, the performance of the proposed framework is 3–4 dB superior to the conventional approach of periodic insertion of intra-coded frames, and 1.5–2 dB away from an ideal system, with infinite decoding delay, operating at Shannon capacity.**

I. Introduction

PREDICTIVE encoding is a well-known source coding technique for efficient low-latency removal of temporal redundancy in audio and video compression systems. Unfortunately, in the context of transmission over lossy channels, it is also a fragile compression technique, as the successful decoding of any symbol is dependent on the successful decoding of all preceding symbols. The key problem in the communication of predictively encoded media is that of predictive mismatch. Predictive mismatch refers to the scenario in which there is a mismatch between the predictor symbol at the encoder and the decoder, leading to an erroneous reconstruction of *all* subsequent reconstructed symbols at the decoder.

Over the last few decades, numerous approaches have addressed the problem of predictive mismatch, but with limited success. In our opinion, this is mainly because predictive coding has traditionally been viewed as a source coding problem. The current work represents a radical departure from the aforementioned view by framing predictive coding as a decoder side-information problem. We shall show that the proposed framework affords us the ability to recover from predictive mismatch, thus avoiding catastrophic decoding failure, in the event of channel losses. Moreover, framing predictive coding in this manner vastly enhances robustness and flexibility as regards to channel behavior while, perhaps surprisingly, resulting in minimal loss in source coding efficiency.

Conventional methods for communication of media over lossy channels consist of approaches that use forward-error correction (FEC) [2] or automatic repeat-request (ARQ) [3], or both [4], [5], to protect against channel erasures.[1] ARQ is advantageous because it retransmits only lost packets, and does not waste any bandwidth. However, ARQ is unsuitable for real-time scenarios since it requires adequate delay between encoding and decoding, for (possibly multiple) retransmissions of lost packets.

FEC, on the other hand, facilitates correction of channel erasures by transmitting redundant parity information. It is known that both the Internet and the wireless channel exhibit burst erasure characteristics—due to congestion and fading, respectively. Thus, ensuring error-free *real-time* delivery of predictively encoded media over these channels requires FEC commensurate with the worst-case behavior of the channel. While heavily protecting critical parts of the encoded stream such as control information and motion vectors (in the case of video) is plausible, heavily protecting the entire stream seems wasteful, especially if error bursts occur only rarely. On the other hand, if only the critical information in a stream is heavily protected, losses in the residual information, while occasional, would still severely impact the reconstructed quality of the predictively encoded media.

In the present work, we propose a joint source-channel coding framework for video which possesses the benefits of both ARQ and FEC, without their respective restrictive assumptions. In particular, the video data is communicated real-time to the decoder. Errors introduced in the decoded stream, due to channel losses, are allowed to propagate, and are corrected at special video frames, denoted "peg frames," that occur periodically in the encoded stream. The peg frames include a *small* amount of additional information, termed "coset information," that is capable of correcting reconstruction errors arising from predictive mismatch. In the proposed framework, propagation of errors is thwarted without wasteful over-protection or increased latency, thereby combining the benefits of both FEC and ARQ.

A significant contribution of the current work is to elucidate the link between predictive encoding and the Wyner–Ziv side-information paradigm; while both have been known in the signal processing literature for nearly three decades, this connection has been hitherto unperceived. In recasting the predictive coding problem as a Wyner–Ziv side-information problem,

Manuscript received January 15, 2003; revised October 13, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pascal Frossard.

The authors are with the Beckman Institute, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: anshul@vision.ai.uiuc.edu; jagmohan@vision.ai.uiuc.edu; ahuja@vision.ai.uiuc.edu).

Digital Object Identifier 10.1109/TMM.2003.822995

[1]The publications cited are a representative, albeit small, subset of those in the literature.

we view the predicted symbol as a corrupted version of the predictor symbol. Thus, the predicted symbol can be decoded using the predictor symbol by employing suitable Wyner–Ziv codes, also known as coset codes. As we shall see later, the key advantage of the proposed framework is that the system can recover from predictive mismatch by efficiently correcting for errors after they occur, thus avoiding catastrophic decoding failure without incurring wastage of bandwidth.

Another noteworthy contribution of the current work is the integration of the proposed approach in a practical video coding standard. As regards practical viability, the most important milestone that the current work achieves is its flexibility in adapting to channel conditions with minimal loss in source-coding performance. We emphasize here that this paper does not prescribe a solution to the end-to-end media delivery problem. This, in general, would be based on the specifics of the application at hand. Our goal in the current paper is the *design of an efficient channel-aware source-coding* algorithm that facilitates recovery from predictive mismatch.

The paper is organized as follows. Section II frames the problem of predictive coding as a variant of the Wyner–Ziv problem. Section III provides a brief description of related work on avoiding predictive mismatch in predictive coding. Section IV introduces the proposed approach through a simple example. Details of the proposed codec design are outlined in Section V. Section VI reports the results of experimental evaluations of the rate-distortion performance of the proposed codec. Conclusions are drawn in Section VII.

II. Proposed Framework

Consider two successive symbols, x_{n-1} and x_n, in the stream to be encoded predictively, where the subscript n denotes the time index. Let $\tilde{x}_{n-1}$ denote the possibly erroneous decoder reconstruction of x_{n-1}. In conventional predictive coding, it is imperative for the encoder to have knowledge of $\tilde{x}_{n-1}$ in order to encode x_n predictively. However, in the context of real-time transmission, the encoder might not have knowledge of $\tilde{x}_{n-1}$, since the encoder is unaware of the error, if any, in the decoder's reconstruction. In such a scenario, conventional predictive coding fails due to the occurrence of predictive mismatch. This problem can be circumvented by an application of the Wyner–Ziv theorem. Let us briefly study the Wyner–Ziv theorem.

Consider the communication of a continuous random variable X, in the presence of a correlated random variable Y, available to the encoder and the decoder (as shown in Fig. 1, with the switch in the "on" position). Denote the rate-distortion function for X as $R_X(D)$, where distortion is measured as the mean squared error. Denote the rate-distortion function for the compression of X with the knowledge of Y as $R_{X|Y}(D)$. Information Theory shows that

$$0 \leq R_{X|Y}(D) \leq R_X(D) \quad \forall D > 0. \tag{1}$$

Thus, the knowledge of the correlated side information Y at the encoder and the decoder reduces the required rate from $R_X(D)$ to $R_{X|Y}(D)$ bits. Hence, the knowledge of Y allows

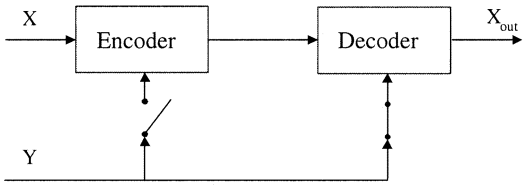

Fig. 1. Block diagram depicting the scenarios where: (a) the side information is available to the encoder and the decoder—the switch is in the "on" position and (b) the side information is available to only the decoder—the switch is in the "off" position.

for better compression of X. Practical video coders realize the gains in (1) by predictively encoding a video frame using the preceding frame as side information available at both the encoder and the decoder.

In [6], Wyner and Ziv considered the communication of X when the knowledge of Y is restricted to the decoder (Fig. 1 with the switch in the "off" position). Denoting the rate-distortion function for this case as $R_{\mathrm{WZ}}(D)$, Wyner and Ziv showed that

$$R_{X|Y}(D) \leq R_{\mathrm{WZ}}(D) \leq R_X(D) \quad \forall D > 0 \tag{2}$$

and $R_{X|Y}(D) = R_{\mathrm{WZ}}(D)$ for the special case in which X, Y are jointly Gaussian random variables. The interested reader is referred to [7] and [8] for a theoretical understanding of the Wyner–Ziv problem and the properties of coset code constructions that approach the Wyner–Ziv bound. Equation (2) shows that if there were some practical means to encode X without knowledge of side information Y, then the performance of such a system would be superior to that of independently encoding X. Conventional predictive coding does not provide such a means, since knowledge of Y at the encoder is required for correct operation, to serve as a predictor for X. Realizing the gains as promised by the Wyner–Ziv theorem in practical video encoding applications is one of the main themes of this paper.

Specifically, denote $Y = \tilde{x}_{n-1}$, the possibly erroneous, decoder side information that is unknown to the encoder, but known to the decoder. Further, let $X = x_n$, the Wyner–Ziv source variable. From (2), it can readily be seen that the encoder can compress x_n at a rate $R_{\mathrm{WZ}}(D)$, such that

$$R_{x_n|\tilde{x}_{n-1}}(D) \leq R_{\mathrm{WZ}}(D) \leq R_{x_n}(D), \quad \forall D > 0. \tag{3}$$

While compression in this manner is possibly inferior to the case when the encoder has knowledge of the decoder predictor $\tilde{x}_{n-1}$, it is superior to compressing x_n independently. Further, the Wyner–Ziv theorem shows that in the absence of knowledge of $\tilde{x}_{n-1}$ at the encoder, x_n cannot be compressed any further. The aforementioned reasoning can also be applied to decode x_n at the decoder using an erroneous decoder reconstruction $\tilde{x}_n$ as side information—the utility of this interpretation will be illustrated in Section IV.

While (3) shows that recasting predictive coding as a decoder side-information problem is beneficial, it does not provide a construction to achieve these gains in practice. In the sequel, we will describe the design of a framework which translates the gains promised by (3) into practice. We will also tackle the multitude of challenges that arise in incorporating the proposed formulation in a practical video encoding system.

III. An Illustrative Example

Consider the communication of a predictively encoded source sequence from a server to a receiver over a packet-erasure channel. Thus, packets injected into the channel have a nonzero probability of not reaching the receiver. Packets that do arrive at the receiver, are decoded without errors. The stipulation of real-time streaming or multicasting on such a channel precludes the use of ARQ.

The server wishes to communicate a first-order scalar process $\{v_k\}$, such that $v_k = v_{k-1} + n_k$ where $v_k \in \mathcal{Z}, n_k \in \mathcal{Z}, k \in \mathcal{Z}^+$ to the receiver. In the present example, the process $\{v_k\}$ satisfies the constraint

$$|v_k - v_{k-1}| \leq d/2, \quad d/2 \in \mathcal{Z}^+ \tag{4}$$

where $d/2 \in \mathcal{Z}^+$ is an arbitrary number. Thus, successive samples of $\{v_k\}$ are correlated. The server puts the symbol $t_k = v_k - v_{k-1}$ in a packet and injects the packet into the channel at time instant k. Owing to the lossy nature of the server–receiver link, there is no guarantee that t_k will arrive at the receiver. In the event the receiver does not receive t_k, it reconstructs $v'_k = v'_{k-1}$, where receiver reconstructions are denoted with a prime symbol ($'$). If the receiver successfully acquires t_k, it reconstructs v_k as $v'_k = t_k + v'_{k-1}$, where v'_{k-1} denotes the receiver's reconstruction of v_{k-1}. If $v'_{k-1} = v_{k-1}, v_k$ is reconstructed without any error. On the other hand, if $v'_{k-1} \neq v_{k-1}$ (due to channel erasures at time instants $l \leq k-1$), then $v'_k \neq v_k$, leading to a distorted reconstruction of v_k at the receiver. It is emphasized that while the channel itself is an erasure channel (i.e., a packet injected into the channel is received without any errors, or not received at all), the loss of a packet causes an *erroneous*, distorted reconstruction of the source symbol v_k at the receiver. Moreover, this distortion is propagated over time to the receiver's reconstruction of each subsequent symbol $v'_j, j > k$, even if the receiver receives the subsequent difference symbols $t_j, j > k$ correctly. This phenomenon is referred to as predictive mismatch. Our aim in the present paper is to ensure that even if a particular symbol is reconstructed erroneously, the error introduced is not propagated indefinitely.

Before describing the solution in the context of the above example, we introduce some terminology. We identify some symbols in the sequence $\{v_k\}$ as "peg symbols" and associate an "epoch" with each peg symbol. In the sequel, for clarity, if v_k is a peg symbol, we denote it as ϑ_k, i.e., $\vartheta_k = v_k$ if v_k is a peg symbol. For example, every tenth symbol could be defined as a peg symbol, i.e., v_j is a peg symbol, if $j \bmod 10 = 0$, and we rewrite v_j as ϑ_j. The epoch of a peg symbol is defined as the set of symbols in between the current and the preceding peg symbol. Our aim is to ensure that errors introduced in the reconstruction of symbols are propagated no further than the nearest peg symbol.

In the proposed approach, for a peg symbol ϑ_k, the encoder not only transmits $t_k = \vartheta_k - v_{k-1}$ to the receiver, it also transmits $\varsigma_k(m) = \vartheta_k \bmod (md+1)$, where $m \in \mathcal{Z}^+$ is a parameter to be determined and d is as defined in (4). The coefficient $\varsigma_k(m)$ is referred to as the coset index of ϑ_k. Using a fixed-rate code, encoding $\varsigma_k(m)$ would require $\log_2(md+1)$ bits. As we shall show, reception of coset index $\varsigma_k(m)$ allows the receiver to correctly reconstruct ϑ_k, as long as the number of erasures in the epoch of ϑ_k were less than m, *irrespective of the position of the m erasures*.

Consider the case in which the receiver had r erasures between two peg symbols ϑ_q and $\vartheta_p, q < p$. We explain the decoding procedure for peg symbol ϑ_p with the help of the following lemmas.

Lemma 1: For peg symbols ϑ_q and ϑ_p, such that $q < p$, if $\vartheta'_q = \vartheta_q$ the following holds:

$$|\vartheta'_p - \vartheta_p| \leq rd/2. \tag{5}$$

Proof: Given in [1]. □

In the above lemma, the condition $\vartheta'_q = \vartheta_q$ implies that there were no errors in the receiver's reconstruction of ϑ_q. Next, we show that if $r \leq m$, then successful reception of coset index $\varsigma_p(m)$ allows the receiver to correctly reconstruct ϑ_p.

Lemma 2: Define $\bar{v}_p$ as

$$\bar{v}_p = \arg \min_{y: y \bmod (md+1) = \varsigma_p(m)} |y - \vartheta'_p|. \tag{6}$$

If the number of erasures $r \leq m, \bar{v}_p = \vartheta_p$. Thus, ϑ_p is decoded correctly.

Proof: The proof is given in [1]. Here, we give the intuition behind (6). We make two observations: 1) It is noted that (6) quantizes ϑ'_p to a point y, where $y \in \mathcal{R}$, where $\mathcal{R}\{\cdots, \varsigma_p(m) - (md+1), \varsigma_p(m), \varsigma_p(m) + (md+1), \cdots\}$, using a nearest neighbor quantizer. Note that $\varsigma_p(m) = \vartheta_p \bmod (md+1)$. Thus, from $\varsigma_p(m)$, the receiver knows that $\vartheta_p \in \mathcal{R}$. Thus, ϑ_p and y, both belong to $\mathcal{R}$. Consequently, as long as the decoder's reconstruction ϑ'_p is within distance $(md+1)/2$ of ϑ_p, it can decode ϑ_p perfectly by performing the quantization operation. Also, from Lemma 1, ϑ'_p is *guaranteed* to be within distance $(md+1)/2$ from ϑ_p if $r \leq m$. Thus, the receiver is able to decode ϑ_p perfectly, thereby thwarting the propagation of error. □

Thus, a cogent selection of the coset index parameter m by the server allows the receiver to decode ϑ_p perfectly, and eliminates the propagation of error due to predictive mismatch. The above discussion is predicated on two important assumptions—namely, that the encoder correctly selects the coset index parameter m and that the coset index is correctly received by the receiver. The latter can be ensured with an arbitrarily high probability by using high-redundancy FEC codes to protect the coset index. We shall see in Section V that the amount of coset information for a typical video stream is small enough for this to be a viable proposition.

We address the issue of selection of the parameter m, by considering the case of real-time streaming over an erasure channel which drops packets independently, with probability $\epsilon, 0 \leq \epsilon \leq 1$. Intuitively, if the epoch duration, denoted as N, is large enough, with a high probability, we would expect the channel to drop $N\epsilon$ packets during the epoch, leading to $N\epsilon$ erasures. Thus, from Lemma 2, if the encoder sets $m = N\epsilon$, there is a high probability that the decoder would be able to correctly reconstruct the peg symbol. We formalize this notion through the following lemma.

Lemma 3: For an independent packet-erasure channel with erasure probability ϵ, if decoding is performed using coset index $\varsigma_p(m)$

$$\bar{v}_p = \vartheta_p \quad \text{with probability } 1 - p_e \tag{7}$$

where $\bar{v}_p$ is as defined in Lemma 2 and p_e in (7) is given by

$$p_e = \sum_{j=m+1}^{N} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}. \tag{8}$$

Further, as $N \longrightarrow \infty, m = N\epsilon \Rightarrow \bar{v}_p = \vartheta_p$ with probability 1, i.e., $\lim_{N\to\infty} p_e = 0$. We refer to p_e as the probability of predictive mismatch.

Proof: Given in [1]. □

Lemma 3 implies that in the limit of large epoch duration, transmitting the coset index corresponding to parameter $m = N\epsilon$ guarantees that the decoder will be able to thwart the propagation of predictive mismatch with probability one. However, it is noted that the cumulative distortion of the source symbols between successive peg symbols scales with N. Thus, setting N large, with $m = N\epsilon$, would ensure a low probability of predictive mismatch, but at the expense of large cumulative distortion. For a fixed N, the probability of predictive mismatch decreases with m, while the rate required for transmission of the coset index, $\log_2(md+1)$, increases with m. In the limiting case, if we set $m = N$, the coset index m will always be able to correct for erasures, since there cannot be more than N erasures among N transmitted symbols. Thus, an intricate tradeoff exists between the epoch duration, the probability of predictive mismatch, and the cumulative distortion of the nonpeg source symbols. A detailed theoretical analysis of this tradeoff is beyond the scope of this paper. The focus of this paper will, instead, be on the design of a practical video codec based on this principle.

The communication methodology delineated above follows from the formulation of predictive coding as a Wyner–Ziv problem, as described in Section II. The erroneous reconstruction of the peg-frame ϑ'_p serves as side information available only at the decoder, and ϑ_p is the source symbol to be communicated to the decoder. The coset index $\varsigma_p(m)$ can be used by the decoder, in conjunction with the decoder side information ϑ'_p, to reconstruct the source symbol ϑ_p. An alternative approach for thwarting the propagation of predictive mismatch is to transmit an independently encoded symbol instead of $\varsigma_p(m)$. Equation (3) suggests that the proposed communication methodology should provide superior performance to that provided by communicating independently coded symbols. This can alternately be understood by noting that, if the entropy of the source symbols v_k is larger than that of the difference symbols, t_k, communication of independently coded symbols would lead to a wastage of bandwidth. In the case of video, we shall demonstrate the verity of this statement in Section V.

To recap, using the above example, we have demonstrated that the proposed approach can eliminate the propagation of predictive mismatch, without knowing which symbols were erased by the channel.

It is noted that other than the naive scheme of independently encoding the peg samples, there is no scheme in literature that can *recover* from predictive mismatch. When FEC is used for real-time transmission, it aims to *preempt* predictive mismatch altogether by protecting each transmitted symbol heavily, thereby leading to significant loss in coding efficiency. It is noted that unlike the proposed approach, protecting only the peg symbols using FEC does not solve the problem. Heavy FEC protection of the difference symbol t_p, corresponding to a peg symbol ϑ_p, only ensures that t_p is decodable. This, however, does not guarantee that ϑ_p will be correctly decoded, since the predictor symbol v'_{p-1} (predictor for ϑ_p) could very well still be erroneous due to channel losses in the epoch of the peg symbol ϑ_p. Section V shall further demonstrate the inadequacy of FEC schemes for practical streaming scenarios with stringent delivery deadlines.

Lastly, it is noted that the illustrative example assumed that the process $\{v_k\}$ satisfies (4). This, in general, cannot be ensured. Nonetheless, as shown in [9], if vector Gauss–Markov processes are considered, (4) holds in an asymptotic sense. Also, the "mod"-operation based coset code used in the preceding example is relatively simple. In the proposed video codec, we define cosets in binary Hamming spaces. We briefly elucidate the link between the "mod"-based coset code, and cosets defined on Hamming spaces, by revisiting the decoding operation defined in Lemma 2. In Lemma 2, we showed that the knowledge of the coset index $\varsigma_p(m)$ *constrains* the set of reconstruction points of ϑ_p from the set of integers $\mathcal{Z}$ to the set $\mathcal{R} = \{\ldots, \varsigma_p(m) - (md+1), \varsigma_p(m), \varsigma_p(m) + (md+1), \ldots\}$. Following this, we observed in (6) that, at the client, determining the point *nearest* to ϑ'_p in the set $\mathcal{R}$ leads to successful decoding of ϑ_p. Analogously, in binary Hamming spaces, we consider a string of n source bits $\mathcal{S}_p$, similar to the peg symbol ϑ_p, that can take on one of 2^n values from the set $\{0,1\}^n$, and a decoder side-information string $\mathcal{S}'_p \in \{0,1\}^n$, akin to ϑ'_p. We define coset-*bits*, analogous to the coset-index $\varsigma_p(m)$, that *constrain* the set of reconstruction points of $\mathcal{S}_p$ from 2^n to a set $\mathcal{R}'$, such that determining the point in $\mathcal{R}'$ that is *nearest* to the decoder's side information $\mathcal{S}'_p$ results in successful decoding. The only question that remains is the method used to generate the coset bits using only $\mathcal{S}_p$ (recall that the coset index $\varsigma_p(m)$ was generated using only ϑ_p), and the set $\mathcal{R}'$ corresponding to the generated coset bits. If we denote the parity[2] of the source bits $\mathcal{S}_p$ as a coset bit, knowledge of the coset bit at the client *constrains* the value of $\mathcal{S}_p$ from 2^n to 2^{n-1} reconstruction points. Similarly, by appropriately defining k parity bits, we can constrain the set of reconstruction points of $\mathcal{S}_p$ from 2^n to 2^{n-k}, thus the parameter k in Hamming space is equivalent to the parameter m in Lemma 2. Further, if we alter the distance metric in (6) from the Euclidean distance to the Hamming distance, Lemma 2 carries over to the case of parity-check codes unaltered. Thus, in the design of our video codec, we will define cosets on parity-check codes, in particular the low-density-parity-check (LDPC) code, and use a soft decoding algorithm on these codes to recover from predictive mismatch.

IV. Design of the Proposed Codec

The encoding approach for video closely mimics the illustrative example from the previous section. The proposed approach

[2]The parity of a string of bits is defined as the XOR operation between them.

is most useful in scenarios where ARQ cannot be used due to stringent delivery deadlines. These scenarios include real-time streaming of stored media, interactive streaming, multicasting, streaming of stored video with limited decoder buffering, etc. We collectively refer to these scenarios as real-time streaming scenarios. We define "peg frames" and their associated epochs in an analogous fashion to the corresponding quantities in Section III. Drift errors in the decoder (the decoder is analogous to the receiver in Section III) can be corrected at each peg frame. We now describe the encoding procedure for the video sequence in detail.

A. Preliminaries

Non-peg frames are encoded using the H.26L, TML 8 [10] video encoder. Denote the jth frame of the video sequence as I_j, the jth reconstructed frame at the encoder as $\hat{I}_j$. As in conventional H.26L encoding, $\hat{I}_{j-1}$ is used as the predictor for encoding frame I_j. We denote the quantized displaced frame difference between I_j and $\hat{I}_{j-1}$ as $\hat{P}_j$. The quantized displaced-frame difference is also referred to as the residual between frames I_j and $\hat{I}_{j-1}$ in the sequel. For emphasis, if frame I_p is a peg frame, we denote it as $\mathcal{I}_p$. Errors introduced in bidirectionally encoded frames are not propagated, hence for clarity, we consider that the first frame I_0 is intra-encoded, while all subsequent frames $I_j, j > 0$ are encoded as predictively coded frames. As stated in Section 1, the motion vectors and the control information for each frame are adequately protected to ensure that these can be decoded at the receiver. The proposed approach strives to correct for predictive mismatch caused by the loss of residual frame data in the reconstructed video.

While encoding video frames that are not peg frames, the H.26L encoder remains unaltered. As summarized in Fig. 2, peg-frame encoding proceeds as follows. Recall that in the illustrative example, for a peg symbol, the encoder communicated two symbols to the receiver: 1) the difference symbol $t_p = \vartheta_p - v_{p-1}$ and 2) the coset index $\varsigma_p(m)$. Akin to this, for each peg video frame $\mathcal{I}_p$, we communicate packets corresponding to: 1) the quantized displaced frame difference $\hat{P}_p$ and 2) coset information for $\mathcal{I}_p$. The quantized displaced frame difference for a peg frame is generated as in conventional H.26L encoding. We now provide a detailed description of the algorithm for generating coset information.

B. Coset Encoding Algorithm

Our design of the video codec relies on the use of good code constructions for the Wyner–Ziv paradigm. Recently, numerous authors have proposed code constructions for the Wyner–Ziv problem, known as coset codes [11]–[14]. It is noted that among the coset codes available in the literature, powerful coset codes based on turbo codes [12], [14] and LDPC codes [13], [14] perform superior to the trellis-based constructions of [11]. While the proposed framework does not inherently require the use of a specific code construction, the use of good code constructions results in improved rate-distortion performance. In this regard, the work of [13] has demonstrated that in the high-rate[3] regime,

[3]The rate of a (n, k) code is defined as k/n. In a high-rate code, $k/n \approx 1$

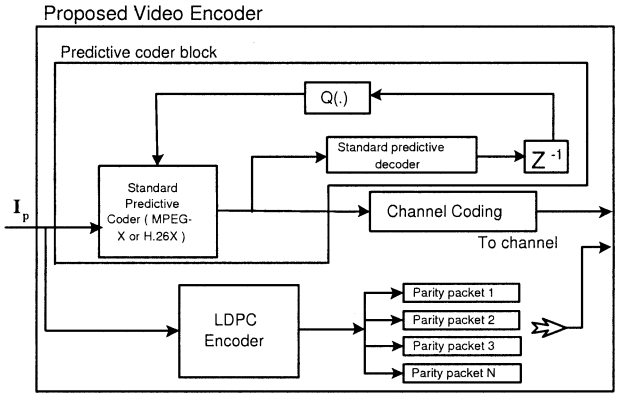

Fig. 2. Block diagram providing an overview of the proposed encoder for a peg frame. Quantizer $q(\cdot)$ in the text is denoted as $Q(\cdot)$ in this diagram.

LDPC codes outperform turbo codes by a large margin. Further, LDPC codes have certain advantages over turbo codes. They have a significantly lower frequency with which decoding errors remain undetected. Moreover, they are also robust to perturbations in the estimated a priori probability distribution of the codewords. Owing to these advantages, the proposed solution is based on LDPC coset codes.

A block diagram of the proposed coset encoding algorithm is depicted in Fig. 3. The proposed approach for generating the coset information for peg frame, $\mathcal{I}_p$, requires that the transform-domain coefficients of $\hat{\mathcal{I}}_p$ take on values from a discrete set, i.e., they take on values from the set of the quantizer reconstruction points.

We observe that, in general, motion compensation and quantization are noncommutable operations. In other words, even if the transform-domain coefficients of a video frame $\hat{I}_{j-1}$ were to take on values from a discrete set. After motion compensation, it cannot be guaranteed that the resultant predictor will still take values from a discrete set. As a consequence, the reconstructed video frame $\hat{I}_j$, formed by adding the displaced frame difference to the *motion-compensated* predictor $\hat{I}_{j-1}$, will not take on values from a discrete set. Accordingly, to satisfy the requirement that the transform-domain coefficients of each reconstructed peg frame $\hat{\mathcal{I}}_p$ take on values from a discrete set, $\hat{\mathcal{I}}_p$ is *requantized* using a quantizer that is known to both the encoder and the decoder. We denote this requantized peg frame as $q(\hat{\mathcal{I}}_p)$. Video frame $q(\hat{\mathcal{I}}_p)$ is obtained by applying the H.26L forward transform to each 4×4 block of $\hat{\mathcal{I}}_p$. The resultant transform-domain coefficients are quantized using the H.26L dead-zone quantizer. Encoding of the subsequent video frame, I_{j+1} proceeds by using the requantized peg frame $q(\hat{\mathcal{I}}_p)$ as the reference frame. As can be gleaned from the above discussion, in the absence of any channel erasures, the only loss in performance of the proposed approach, over an unmodified H.26L encoder is due to the requantization of the peg frames. As we shall show in Section V, this loss is small.

Coset information for peg frame $\hat{\mathcal{I}}_p$ is generated by applying LDPC coset codes to the transform-domain coefficients of the requantized peg-frame $q(\hat{\mathcal{I}}_p)$. We describe the generation of the coset information for $\hat{\mathcal{I}}_p$ with the help of the following notation. Application of the 4×4 H.26L forward transform to an image block results in the generation of 16 transform-domain coef-

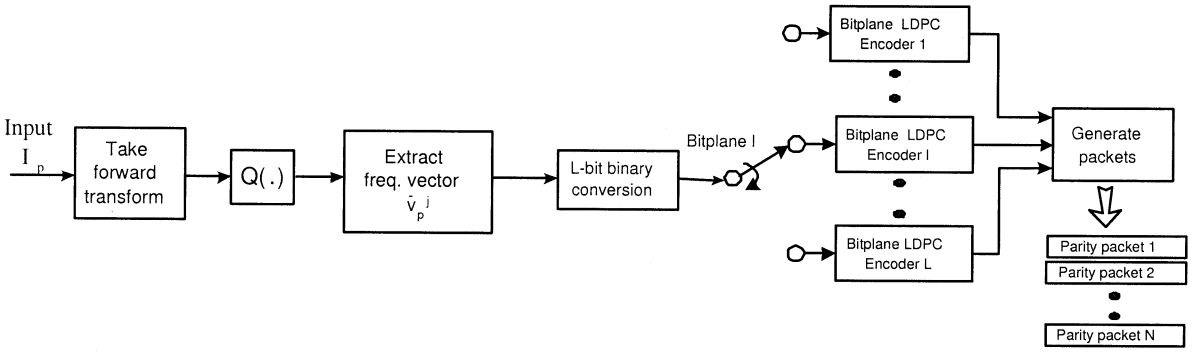

Fig. 3. Block diagram of proposed coset encoder, depicting the generation of coset information for a generic frequency vector. Quantizer $q(\cdot)$ in the text is denoted as $Q(\cdot)$ in this diagram.

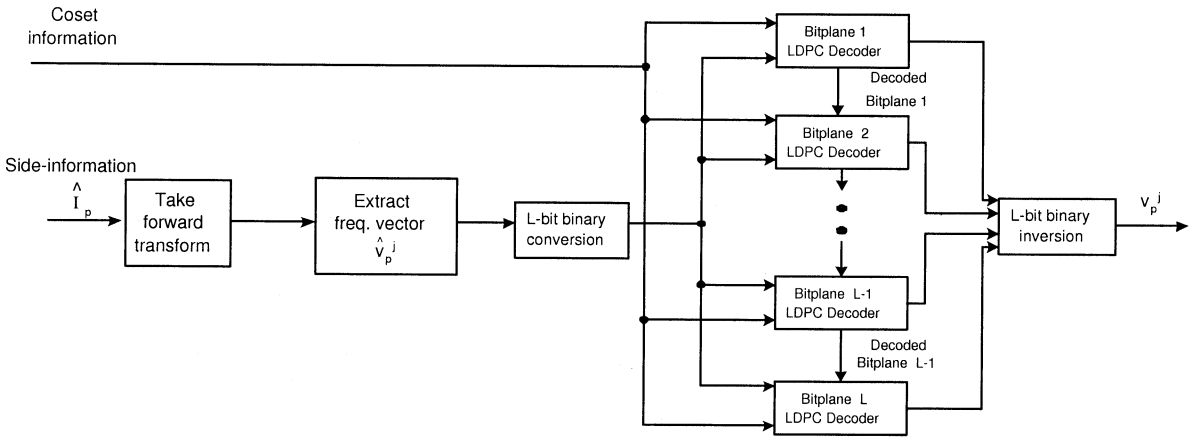

Fig. 4. Block diagram of proposed coset decoder, depicting the reconstruction of a generic frequency vector of a peg frame.

ficients. Conceptually, a transformed image can be thought to consist of 16 vector frequency components, with the kth vector, $0 \leq k \leq 15$, containing the kth transform-domain coefficient of each 4×4 block. We denote the kth transform-domain vector frequency component of image $\hat{\mathcal{I}}_p$ as $\hat{\mathbf{v}}_p^k$. We denote the length of $\hat{\mathbf{v}}_p^k$ as q, where q is the number of 4×4 blocks in an image. The corresponding vectors for the requantized frame $q(\hat{\mathcal{I}}_p)$ are denoted as $\bar{\mathbf{v}}_p^k$. As stated earlier, LDPC encoding is performed on $\bar{\mathbf{v}}_p^k$. The encoding and decoding of each frequency vector $\bar{\mathbf{v}}_p^k$ is identical and independent of other frequency vectors, hence we drop the superscript from $\bar{\mathbf{v}}_p^k$ in the sequel and describe the encoding and decoding procedure for a generic frequency vector $\bar{\mathbf{v}}_p$.

Fig. 3 shows a block diagram of the LDPC encoder. Each symbol of vector $\bar{\mathbf{v}}_p$ is converted into its L-bit binary representation. Vector $\bar{\mathbf{v}}_p(l)$ denotes the vector containing the lth bit plane of $\bar{\mathbf{v}}_p, 0 \leq l \leq L$, where L denotes the number of bit-planes. Each bit-plane vector $\bar{\mathbf{v}}_p(l)$ is encoded using a $p \times q$ systematic LDPC encoder. The parity bits $p-q$ generated by the LDPC encoder constitute the coset information for $\bar{\mathbf{v}}_p(l)$, [13]. A *coset vector* for $\bar{\mathbf{v}}_p$ is generated by concatenating the parity bits for each bit-plane vector $\bar{\mathbf{v}}_p(l), 0 \leq l \leq L$ of $\bar{\mathbf{v}}_p$.

Recall that the above operations are performed for each $\bar{\mathbf{v}}_p^k, 0 \leq k \leq 15$. A *coset packet* is generated by concatenating the coset vectors of each frequency vector $\bar{\mathbf{v}}_p^k, 0 \leq k \leq 15$ of video frame $q(\hat{\mathcal{I}}_p)$. Appropriate control information to extract the parity bits for each frequency-bit-plane vector $\bar{\mathbf{v}}_p^k(l)$ is also included in each packet.

In our implementation, we use the algorithm in [15] to generate the parity-check matrix $\mathbf{H}$, and the generator matrix $\mathbf{G}$ of the LDPC code. Then, the $p-q$ parity bits of the product $\bar{\mathbf{v}}_p(l) \cdot \mathbf{G}$ constitute the coset information for the bitplane vector $\bar{\mathbf{v}}_p(l)$. The size of the LDPC code (p) decides the amount of coset information generated and is related to the efficacy of the transmitted coset information in correcting for previous channel erasures. Thus, the code-size p provides a tradeoff akin to that provided by the coset parameter m in Section III. In the proposed framework, the frequency vectors are multiply encoded using varying values of p to generate multiple coset packets. During the streaming session, depending on the estimated channel characteristics, appropriate coset packets (i.e., packets corresponding to appropriate values of p) are selected for transmission to the decoder. In case encoding is performed during the streaming session, an appropriate value of p can be used instead of generating multiple coset packets.

C. Coset Decoding Algorithm

In this section, we describe the decoding algorithm for peg frames, both in the absence and in the presence of erasures. Fig. 4 gives a schematic representation of the coset decoding algorithm. All nonpeg frames are reconstructed using a conventional H.26L decoder.

In the absence of erasures, the decoder reconstructs peg frame $\hat{\mathcal{I}}_p$ by adding the displaced-frame difference $\hat{P}_p$, to $\hat{I}_{p-1}$. Recall that the encoder encodes the subsequent video frame $\hat{I}_{p+1}$ using the requantized peg frame $q(\hat{\mathcal{I}}_p)$. Thus, the decoder requantizes

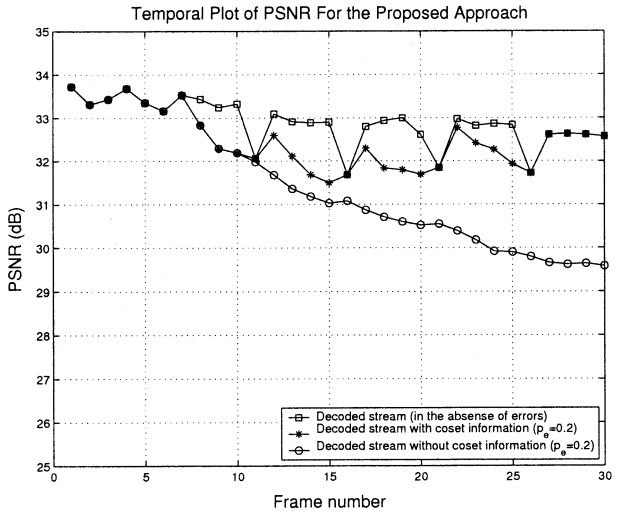

Fig. 5. Temporal PSNR plot comparing the performance of the proposed approach in the presence, and the absence of errors. An error burst with $p_e = 0.2$ was introduced between video frames 7 and 22 in this simulation.

$\hat{\mathcal{I}}_p$ to yield $q(\hat{\mathcal{I}}_p)$. The video frame $q(\hat{\mathcal{I}}_p)$ is then used as the reference frame for decoding $\hat{I}_{p+1}$. In the absence of erasures, the decoder and the encoder use the same predictors, and there is no propagation of error.

Next, we describe the decoding procedure in the presence of erasures. As in the erasure-free case, an initial estimate of the peg frame, denoted $\hat{\mathcal{I}}'_p$, is generated by adding the received displaced frame difference $\hat{P}_p$, to the motion-compensated decoder reconstruction of the previous frame $\hat{I}'_{p-1}$. In the presence of erasures, $\hat{I}'_{p-1}$ is liable to be erroneous and parts of $\hat{P}_p$ are liable to be erased, in which case the reconstruction $\hat{\mathcal{I}}'_p$ will, in general, not be equal to the encoder's reconstruction, $\hat{\mathcal{I}}_p$.

Note that in order to avoid predictive mismatch in the event of erasures, the receiver does not need to reconstruct $\hat{\mathcal{I}}_p$; instead, reconstruction of $q(\hat{\mathcal{I}}_p)$ would suffice, since the encoder would have used $q(\hat{\mathcal{I}}_p)$ as the predictor for the subsequent video frame. The receiver reconstructs $q(\hat{\mathcal{I}}_p)$ by using the received coset information to correct its erroneous estimate $\hat{\mathcal{I}}'_p$. Thus, the estimate $\hat{\mathcal{I}}'_p$ serves as side information (in the context of Wyner–Ziv coding), which can be used in conjunction with coset information to reconstruct the source information $\hat{\mathcal{I}}_p$. The erroneous estimate $\hat{\mathcal{I}}'_p$ is corrected by performing LDPC decoding using the received coset information, which consists of LDPC parity bits.

Decoding of a generic frequency vector $\hat{\mathbf{v}}_p$ proceeds by using the decoder reconstruction $\hat{\mathbf{v}}'_p$ as side information (extracted from $\hat{\mathcal{I}}'_p$) and the parity bits as the coset information. The decoder extracts the parity bits from the coset packet for each bitplane l of each frequency vector $\hat{\mathbf{v}}^k_p(l)$ and performs side-information decoding as depicted in Fig. 4. The bit planes of $\hat{\mathbf{v}}_p$ are sequentially decoded. The sequential decoding structure of LDPC decoders is required since successful decoding of bitplane l yields information about the symbol to be reconstructed, which can be used as additional side information for decoding the subsequent bitplanes $l+1$ through L.

In this manner, the decoder can eliminate predictive mismatch, provided it receives the coset information. Thus, it is imperative that coset information is adequately protected prior to transmission. In Section V, we shall show that the coset information constitutes but a small fraction of the bit rate of the H.26L encoded stream. Thus, it is viable to protect it heavily. The crucial observation is that this coset information suffices irrespective of the position of the erasures in the epoch of a peg frame. In the absence of knowledge of the position of the erasures, the standard FEC solution would require heavy protection of the entire H.26L encoded stream and is thus unviable.

V. Results

We investigate the overall performance of the proposed approach in this section. Efficient elimination of predictive mismatch in a practical application entails the usage of numerous other system components, such as using FEC codes and/or using a rate-distortion optimized transmission policy, etc., in conjunction with the proposed approach. Therefore, the efficacy of the proposed solution in a practical application would be influenced by numerous other factors as well. In light of this, we report results on the performance of a baseline system based on the proposed framework, so as not to shadow the results by other decisions made in the streaming system.

Simulation results are provided for one hundred frames of the Foreman sequence encoded at 30 Hz, transmitted over a packet erasure channel. We present simulation results for an i.i.d erasure channel which facilitate the comparison of the proposed approach with readily computable Information Theoretic capacity bounds. Thus, in our simulation studies, we consider the transmission of the encoded video over an i.i.d packet erasure channel, with packet-drop probability p_e.

Prior to presenting the simulation results, we give a brief description of the parameters of the proposed system, henceforth referred to as System 1.

A. System 1: Error-Recovery Using Coset Codes

In this system, the error-recovery mechanism outlined in the previous sections is used to eliminate predictive mismatch. The video sequences were encoded in the IPPP... format, with peg frames defined at regular intervals. The quantization reconstruction routine in the H.26L encoder was altered to reconstruct each point precisely (as opposed to the linear approximation made in the original H.26L code), which lead to an improvement in performance of approximately 0.1 dB. Loop filtering was applied to the video frame just prior to displaying it, but the predictor frame was not loop filtered. This led to a minor improvement in the rate-distortion performance as well (less than 0.1 dB). Each frame of the video sequence was split into eight slices—each slice constituted one packet. Thus, one video frame generated eight packets, in all.

LDPC codes, as proposed in [15], were used in the proposed system. The parity check matrix of each code was generated using a pseudorandom seed. Following this, the algorithm proposed in [15] eliminates all cycles of length four in the bipartite graph of the code. Sixty-four such LDPC codes with varying redundancy were used in the simulations. The codebooks were made available to the encoder and the decoder prior to streaming. FEC was used to ensure that the coset information associated with each peg frame arrives at the decoder with a

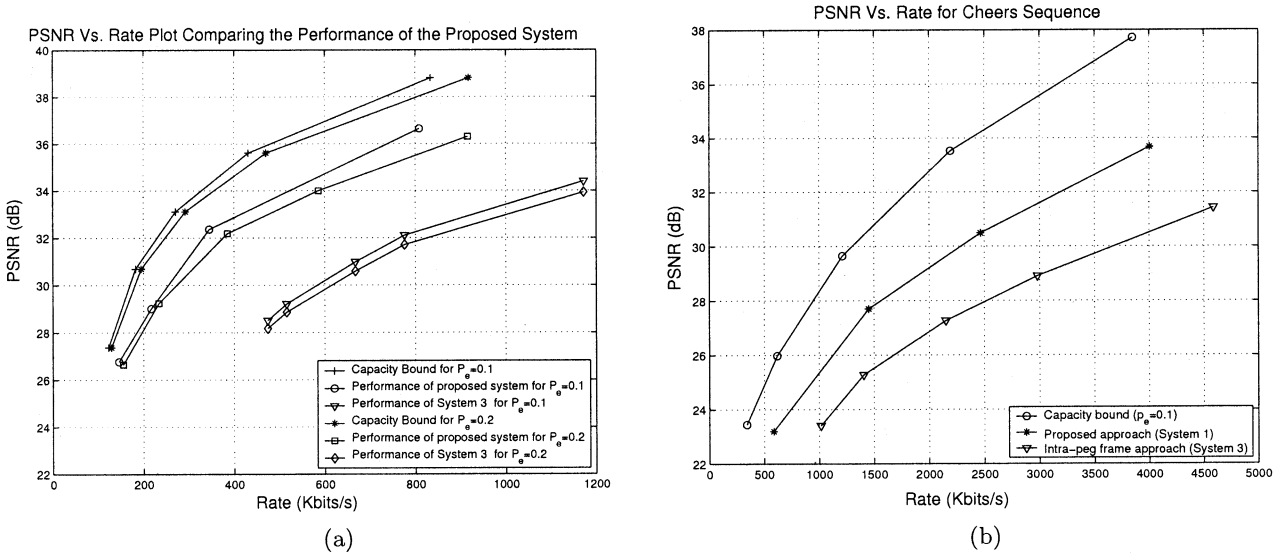

Fig. 6. (a) Distortion-rate comparison of proposed system with capacity bound (System 2) and the intra-frame approach (System 4). (b) Distortion-rate comparison of the proposed approach with the capacity bound (System 2), and the intra-frame approach (System 4) for the Cheers sequence.

very high probability. In our simulations, we use Reed–Solomon based protection of the coset information. The parameters of the code were chosen to ensure that the probability that the coset information does not arrive at the decoder (due to channel erasures) was upper bounded by 10^{-3}.

Our first set of simulations demonstrates that the proposed approach is indeed capable of recovering from predictive mismatch.

Recovery From Predictive Mismatch: In this experiment, we transmit the video data over an erasure channel, with an error burst between frames 7 and 22, with an independent packet loss probability of $p_e = 0.2$ during the burst. Fig. 5 plots the temporal variation in peak signal-to-noise ratio (PSNR) for this simulation. As can be seen, the proposed system *completely recovered* from the error burst at (peg) frame 25. We also plot the variation in PSNR with frame index for the same encoded stream when coset information is not used for correcting for drift errors. As can be seen from Fig. 5, when the coset information is not transmitted, the system is unable to recover from predictive mismatch. We also note that in conventional predictive coding, though the PSNR of the decoded frames improves over time after an error burst, annoying visual artifacts persist in the decoded stream long afterwards. In the proposed approach, however, once the decoder has recovered from the error burst, the visual quality of the video is not degraded in any way.

In the next set of simulations, we compare the performance of the proposed approach (System 1) with that of the following three systems.

B. System 2: An "Ideal" Rate-Distortion Optimized System Working at Capacity

This hypothetical system provides an upper bound on the performance of *any* streaming system. The performance of such a system is computed using the rate-distortion characteristics of the stream and the characteristics of the channel. Denoting the packet-drop probability of the independent packet-drop channel as p_e, the capacity of this channel is given by $C = 1 - p_e$. Thus, to successfully transmit 1 byte on the channel, the server has to typically transmit $1/C$ bytes. Denoting one point on the rate-distortion curve of the source as $R(D_i) = R_i$, the performance of this ideal system for the same distortion is given by $R(D_i) = R_i/C$. However, operating at $(R_i/C, D_i)$ requires that there be infinite time before the delivery deadline of each data unit. Since we are considering only "one-shot" transmission scenarios, the performance of the proposed approach is expected to be worse than that of this ideal system. However, comparing results with such a bound benchmarks the proposed approach, facilitating future comparisons.

C. System 3: Error Recovery Using Intra-Coded Frames

This system mitigates the propagation of error by using intra-coded frames instead of peg frames. In order to make a fair comparison between System 1 and this system, intra-coded frames were placed at the same positions as the peg frames in the encoded stream. Further, as in System 1, Reed–Solomon based FEC was used to protect the intra-coded frames as well. An adequate amount of protection was used to ensure that the probability that an intra-coded frame is not decodable at the decoder was 10^{-3} (as was done in System 1).

Next, we perform a comparative study of the performance of the three systems described above.

PSNR Versus Rate Performance Comparison (for p_e Fixed): Our first experiment simulates the communication of the compressed stream and the coset packets across an erasure prone channel (it is noted that coset packets are liable to be dropped as well). Fig. 6(a) plots the PSNR versus rate graphs for $p_e = 0.1$ and $p_e = 0.2$. Also plotted in the figure are the capacity bounds of System 2 for $p_e = 0.1$ and $p_e = 0.2$. As can be discerned from Fig. 6(a), the proposed system comes within 1–2.5 dB of the capacity bound, depending on the rate, while allowing real-time transmission. Fig. 6(a) also compares the performance of the proposed system with System 3. As can be seen from the graph, the proposed system performs 3.5–4 dB better than System 3.

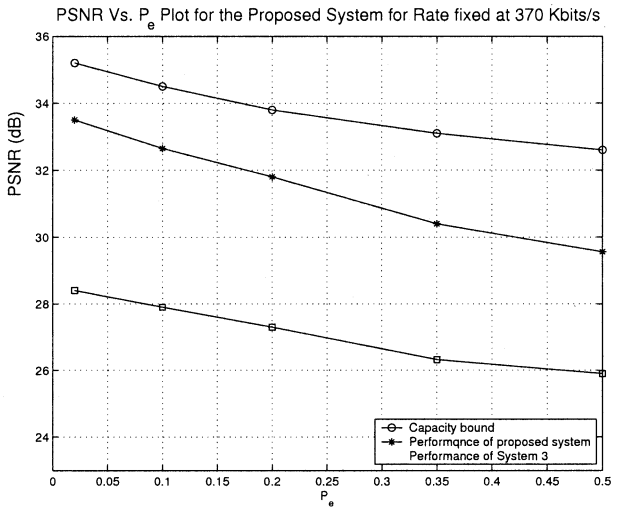

Fig. 7. PSNR versus P_e for proposed approach with other systems, with rate fixed at 370 kbits/s.

PSNR Versus Rate Performance for the "Cheers'" Sequence (for p_e Fixed): Fig. 6(b) plots simulation results of the complete operational system for the high-innovation "Cheers" sequence (analogous to Fig. 6(a) for the "Foreman" sequence). As can be seen from the graph, the disparity between the proposed approach and the capacity bound is larger for a high motion sequence such as Cheers. As compared to the results with Foreman, its performance is closer to the performance of System 3—this is only to be expected, since the amount of innovation has less of an impact on the gap between System 2 and System 3 as compared to the performance of the proposed approach. From the figure, it is noted that the proposed approach is 2–4 dB better than System 3. Among the test sequences on which the simulations were run, the performance was the worst for the Cheers sequence.

PSNR Versus p_e Performance Comparison (for Rate Fixed): Fig. 7 plots the variation in PSNR as a function of the channel erasure probability p_e for the same experiment as above (for the Foreman sequence). Also plotted is the capacity bound of System 2. As can be discerned from the figure, the performance of the proposed system deteriorates with an increase in the probability of channel failure p_e. This is because, for large p_e, a large number of packets are dropped, as a consequence, even for an epoch as small as five frames, the cumulative distortion of the nonpeg video frames results in a large average distortion of the transmitted video. Also plotted in the figure is the corresponding plot for the intra-peg frame approach (System 3).

In summary, the experiments demonstrate that the proposed approach is able to recover from errors in the event of channel failure, thus eliminating the propagation of error without overly sacrificing the performance of the system in the absence of channel failures.

VI. Related Prior Work

There exists a substantial amount of work in the literature on mitigating the effect of predictive mismatch [16]–[22]. However, none share the paradigm proposed in this work. The work of [21] and [22] addresses the problem of predictive mismatch in the context of multiple description coding of video under a highly restrictive set of assumptions on the channel behavior. The work of [20] and [19] proposes an adaptive intra-refresh strategy, whereby a carefully selected subset of macro-blocks are intra-coded to avoid the indefinite propagation of errors in a video stream. There also exist error-concealment techniques [23], [24] that lessen the impact of predictive mismatch by exploiting models of natural images. The performance of these techniques is limited by the assumptions made by their respective models. Layered coding techniques circumvent the problem of mismatch [16] by assuming that the network provides a guaranteed low-bandwidth link on which a coarse representation of each video frame of the predictively encoded stream is transmitted, while refinement of the coarse representation is transmitted on a variable delay, possibly lossy link between the encoder and the decoder. Predictive mismatch is expressly avoided by using only the coarse representation of a video frame as the predictor; this leads to compression inefficiency. Moreover, the current day public Internet does not provide a guaranteed bandwidth link, required for the coarse representation.

In contrast to the aforementioned algorithms, the current work is based on the fact that even a degraded erroneous predictor has information in common with the image to be predicted.

VII. Conclusion

In our opinion, the most important contribution of this work is the elucidation of the link between Wyner–Ziv side-information coding and predictive encoding of video. Another important contribution of the current work is the design of a video encoder that espouses the concepts outlined in Section III. Further, the simulation results demonstrate that a carefully designed video codec is capable of performing competitively with the state-of-the-art solutions.

This paper introduces the concepts behind Wyner–Ziv coding of video through the example of robust video coding in the context of transmission over erasure prone channels. The concepts outlined in this paper, however, are not limited to this application. Indeed, other important video coding applications such as multiple description coding and scalable coding of predictively encoded video can also be formulated as decoder side-information problems, and thus can benefit from the principles presented here.

It has recently been brought to our attention that [25] has also proposed a video codec based on the principles delineated in [26]. In contrast to the algorithm presented in this paper, the primary goal of their work is the design of a low-complexity encoder.

Finally, some of our other work exploring the relationship between predictive coding and the decoder side-information problem can be found in [1], [9], [26]–[30].

Acknowledgment

The authors thank the Associate Editor, Dr. P. Frossard, and the anonymous reviewers whose meticulous comments vastly improved the content and the presentation of this paper. They also thank Dr. P. A. Chou, Microsoft Research, for his help during the preliminary stages of this work, Prof. D. MacKay and Prof. R. Neal for putting the C code for LDPC codes on

their webpage, and Dr. O. Verscheure, IBM Research, for his guidance during the latter part of this work.

REFERENCES

[1] A. Sehgal, A. Jagmohan, and N. Ahuja. Wyner-Ziv Coding of Video: An Error Resilient Compression Framework. Beckman Inst., Univ. Illinois, Urbana. [Online]. Available: http://vision.ai.uiuc.edu/anshul/SehgalJA03.pdf

[2] B. Kurceren and J. W. Modestino, "A joint source-channel coding approach to network transport on digital video," in *Proc. IEEE INFOCOM*, vol. 2, 2000, pp. 717–726.

[3] P. A. Chou and Z. Miao, "Rate-distortion optimized sender-driven streaming over best-effort networks," in *IEEE Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 587–592.

[4] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," in *IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 440–449.

[5] W.-T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 373–386, Mar. 2001.

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

[7] R. Zamir and S. Shamai, "Nested linear/lattice codes for wyner-ziv encoding," in *Information Theory Workshop*, Killarney, Ireland, June 1998, pp. 92–93.

[8] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–1276, June 2002.

[9] A. Sehgal, A. Jagmohan, and N. Ahuja, "Layered predictive coding based on the Wyner-Ziv problem," in *Proc. IEEE Int. Conf. Communication Systems*, Singapore, Nov. 2002.

[10] T. Wiegand, "H.26L test model long-term number 9 (tml-9) draft0," ITU-T Video Coding Experts Group, Doc. VCEG-N83d1, Dec. 2001.

[11] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," in *Proc. IEEE Data Compression Conf.*, 1999, pp. 158–167.

[12] Y. Zhao and J. Garca-Fras, "Data compression of correlated nonbinary sources using punctured turbo codes," in *Proc. DCC'02*, Snowbird, UT, Apr. 2002.

[13] A. Liveris, Z. Xiong, and C. Georghiadis, "Compression of binary sources with side-information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2002.

[14] J. Garca-Fras and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, pp. 417–419, Oct. 2001.

[15] D. J. C. Mackay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, p. 1645, Aug. 1996.

[16] K. Rose and S. Regunathan, "Toward optimality in scalable predictive coding," *IEEE Trans. Image Processing*, vol. 10, pp. 965–976, July 2001.

[17] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, "Multiple-description video coding using motion-compensated temporal prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 193–204, Mar. 2002.

[18] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," in *IEEE Workshop on Mumtimedia Signal Processing*, 2001, pp. 441–446.

[19] T. Stockhammer, T. Wiegand, and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks," in *Proc. Int. Conf. Image Processing*, vol. 2, Sept. 2002, pp. 173–176.

[20] P. Frossard and O. Verscheure, "MPEG-2 video over lossy packet networks: Adaptive MPEG-2 information structuring," in *SPIE Int. Symp. Voice, Video, and Data Communications*, vol. 3528, Nov. 1998, pp. 113–123.

[21] S. John and V. Vaishampayan, "Interframe multiple description video coder," Columbia Univ., New York, Packet Video Workshop, 1998.

[22] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri, "Multiple description coding for video using motion compensated prediction," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999.

[23] S. S. Hemami and R. M. Gray, "Subband coded image reconstruction for lossy packet networks," *IEEE Trans. Image Processing*, vol. 6, pp. 523–539, Apr. 1997.

[24] D. S. Turaga and T. Chen, "Model-based error concealment for wireless video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 483–495, June 2002.

[25] R. Puri and K. Ramchandran, "PRISM: A video coding architecture based on distributed compression principles," ERL, Berkeley, CA, Tech. Rep..

[26] A. Jagmohan, A. Sehgal, and N. Ahuja, "Predictive coding using coset codes," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2002, pp. 29–32.

[27] A. Sehgal and N. Ahuja, "Robust predictive coding and the Wyner-Ziv problem," in *Proc. IEEE Data Compression Conf.*, 2003, pp. 103–112.

[28] A. Jagmohan and N. Ahuja, "Wyner-Ziv encoded predictive multiple descriptions," in *IEEE Data Compression Conf.*, 2003, pp. 213–222.

[29] A. Jagmohanm, A. Sehgal, and N. Ahuja, "WYZE-PMD based multiple description video codec," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, 2003, pp. 569–572.

[30] A. Sehgal, A. Jagmohan, and N. Ahuja, "A state-free causal video encoding paradigm," in *Proc. IEEE Int. Conf. Image Processing* , vol. 1, Sept. 2003, pp. 605–608.

Anshul Sehgal received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1999. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Illinois, Urbana–Champaign.

Ashish Jagmohan received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, India, in May 1999. He is currently pursuing the Ph.D. degree in the Electrical Engineering Department, University of Illinois, Urbana–Champaign.

His research interests include image and video processing, signal processing, and communications.

Narendra Ahuja (S'79–M'79–SM'85–F'92) received the B.E. degree (Hons.) in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree (with distinction) in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975, he was a Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979, he was with the Computer Vision Laboratory, University of Maryland. Since 1979 he has been with the University of Illinois, Urbana–Champaign, where he is currently the Donald Biggar Willet Professor in the Department of Electrical and Computer Engineering, the Beckman Institute, and the Coordinated Science Laboratory. He has co-authored the books Pattern Models (New York: Wiley, 1983), Motion and Structure from Image Sequences (New York: Springer-Verlag, 1992), and Face and Gesture Recognition (Norwell, MA: Kluwer, 2001), and co-edited the book Advances in Image Understanding (New York: IEEE Press, 1996). His current research emphasizes integrated use of multiple image sources of scene information to construct three-dimenstional and other descriptions of scenes; the use of integrated image analysis for realistic image synthesis; sensors for computer vision, extraction and representation of spatial structure, e.g., in images and video; and use of the results of image analysis for a variety of applications including visual communication, image manipulation, robotics, and scene navigation.

Dr. Ahuja is on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Computer Vision, Graphics, and Image Processing*, the *Journal of Mathematical Imaging and Vision*, the *Journal of Pattern Analysis and Applications*, the *International Journal of Imaging Systems and Technology*, and the *Journal of Information Science and Technology*. He was a Guest Co-Editor of the *Artificial Intillegence* special issue on vision. He received the 1999 Emanuel R. Piore Award of the IEEE and the 1998 Technology Achievement Award of the International Society for Optical Engineering. He was selected as Associate (1998–1999) and Beckman Associate (1990–1991) in the University of Illinois Center for Advanced Study. He received University Scholar Award (1985), Presidential Young Investigator Award (1984), National Scholarship (1967–1972), and President's Merit Award (1966). He is a Fellow of the American Association for Artificial Intelligence, International Association for Pattern Recognition, Association for Computing Machinery, the American Association for the Advancement of Science, and the International Society for Optical Engineering.