PRISM: AN UPLINK-FRIENDLY MULTIMEDIA CODING PARADIGM

Rohit Puri

UC Berkeley,
EECS Dept.,
{rpuri}@eecs.berkeley.edu

Kannan Ramchandran

UC Berkeley,
EECS Dept.,
{kannanr}@eecs.berkeley.edu

ABSTRACT

In this work, we present PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding), a new video coding paradigm based on the principles of distributed source coding. Incurring the low encoding complexity of intra-frame (still-image) video coding, PRISM approaches the high compression efficiency of full-motion inter-frame video coding, while simultaneously offering natural robustness to the drift problem and channel loss in direct contrast to conventional video coding architectures (MPEG and H.26L). These traits make it well-matched to uplink-rich applications such as multimedia over wireless networks, wireless video and sensor cameras, etc.

1. INTRODUCTION

We are at the dawn of a new era where traditional views of video transmission (primarily television broadcast models) are being challenged. With the expected proliferation of digital camera equipped cellular phones as well as the emergence of low-power surveillance and sensor networks, the days of typecasting media transmission as a "downlink" experience (e.g., TV broadcast) are over. Under the existing video codec architectures, the video encoder is the computational workhorse of the video codec, with its computational complexity dominated by the motion compensated prediction operation. The conventional video decoder on the other hand is a relatively lightweight device operating in a "slave" mode to the encoder. Such a model is obviously at complete odds with the emerging class of "uplink" rich media applications such as video transmission over wireless networks (e.g., cellular or 802.11) and low-power video sensor networks (e.g., surveillance or security applications). The architectural requirements here include:

1. **low-power and computational complexity at the mobile/sensor node for both encoding and decoding of video:** this is critical to prolonging battery life of these low-power devices;
2. **high compression efficiency:** both bandwidth and transmission power are at a premium, calling for maximal compression efficiency;
3. **robustness to channel loss:** transmission losses in the wireless medium can lead to packet drops or even frame drops which can lead to drift between the encoder and the decoder.

Current video coding paradigms fail to simultaneously address these demanding requirements satisfactorily. The predictive or inter-frame video coding mode achieves high compression efficiency, but it is computationally heavy at the encoder while also being very fragile to packet losses. Alternatively, intra-frame video coding methods have low computational complexity and are relatively robust to packet drops but achieve poor compression efficiency. This raises the interesting question of whether it is possible to architect a new video coding paradigm that is driven to attain all these requirements possibly at the expense of shifting the computational burden from the encoder to the decoder (see Figure 1).

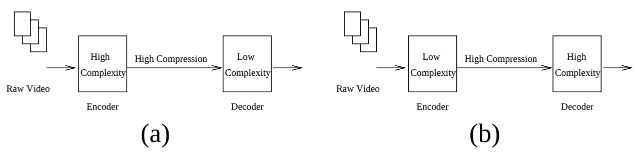


Fig. 1. (a) Conventional video encoding architecture comprises of a high complexity encoder and a low complexity decoder. (b) Proposed video coding paradigm (PRISM) comprising of a low complexity encoder achieving the compression performance of the conventional framework.

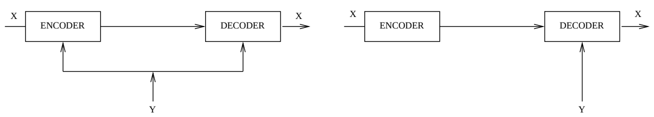
Motivated by this, in this work, we present PRISM [1] (Power-efficient, Robust, hIgh-compression Syndrome based Multimedia coding), a novel video encoding paradigm that represents a significant departure from the traditional video coding methods. Leveraging the power of distributed compression methods [2, 3], PRISM incurs the low encoding complexity of still image compression methods, approaches the compression performance of conventional video coding techniques and additionally offers the feature of robustness. A typical network configuration involving the PRISM codec consists of a low-complexity PRISM encoder at the transmit node and a conventional low-complexity decoder at the receive node that are interfaced through a base station that has a “trans-coding proxy” that efficiently converts the PRISM bit-stream into a standard bit-stream (e.g., MPEG/H.26L). Under this architecture, the entire computational burden has been absorbed into the network device.

In this paper, in addition to introducing the PRISM paradigm, we will present a specific implementation of the PRISM framework that is inspired by staying close to current video standards involving block motion-compensation and DCT’s. The scope of applicability of the framework extends to beyond this narrow instantiation however.

2. BASIC CONCEPTS

To get insights into the PRISM coding approach, we first examine an instructive toy example that was first presented in [4] (See Figure 2).

2.1. Example for Coding with Side Information



(a) (b)

Fig. 2. X and Y are correlated, length 3-bit binary data equally likely taking each of the 8 possible values, individually. The Hamming distance between the codeword for X and that for Y is at most 1. (a) Both encoder and decoder use the side information Y which is correlated to X. Here X can be encoded with 2 bits. (b) Only decoder accesses Y. Here too, X can be encoded using 2 bits.

Let X and Y be length 3-bit binary data that can equally likely take on each of the 8 possible binary 3-tuples. However, X and Y are correlated such that the Hamming distance between X and Y is at most 1. The goal is to efficiently encode X in the two scenarios shown in Figure 2 so that it can be perfectly reconstructed at the decoder.

Scenario 1: In the first scenario (see Figure 2 (a)), Y is present both at the encoder and at the decoder. Here X can be predicted from Y. The residue ($X \oplus Y$) or the error pattern of X with respect to Y takes 4 distinct values and hence can be encoded with 2 bits. X, Y and the correlation between X and Y are respectively analogous to the current video block, the predictor from the frame memory, and the temporal correlation between successive video frames. Hence this mode of encoding is similar to **predictive coding**.

Scenario 2: Here, unlike the first scenario (see Figure 2 (b)) the encoder for X does not have access to Y. However, it does know the correlation structure between them and also knows that the decoder has access to Y. Surprisingly even in this seemingly worse scenario X can be encoded with 2 bits!

This can be done using the following approach. The space of codewords of X is partitioned into 4 sets each containing 2 codewords, namely, **Coset1** ([0 0 0] and [1 1 1]), **Coset2** ([0 0 1] and [1 1 0]), **Coset3** ([0 1 0] and [1 0 1]) and **Coset4** ([1 0 0] and [0 1 1]). The encoder for X identifies the set containing the codeword for X and sends the index for the set instead of the individual codeword. This can be done in 2 bits. The decoder, in turn, on the reception of the coset index, uses Y to disambiguate the correct X from the set by declaring the codeword that is closest to to Y as the answer. Note that the distance between X and Y is at most 1, and the distance between the 2 codewords in any set is 3. Hence, decoding can be done perfectly. This mode of encoding where the decoder has access to correlated side information is known as **side information coding** [2, 3].

1. The partitioning of the source codeword space, index labeling of the resulting cosets (index labels for cosets are also called syndromes) and mapping from the source codeword space to the space of coset indices can be done in a *computationally efficient* way through the framework of coset codes [5] resulting in **low encoding complexity**.
2. **Coset1** is a repetition channel code [6] of distance 3 and the other sets are cosets [5] of this code in the codeword space of X. We have used a channel code that is “matched” to the correlation distance (equivalently, noise) between X and Y to partition the source codeword space of X resulting in **high compression** performance.
3. This partitioning of X is also *universal*. The same partitioning of X works for all Y regardless of the value of Y as long as both X and Y satisfy the correlation structure. e.g., if X is [0 1 0], then the same encoding for X (index of **Coset 3**) will be applicable to all cases of Y i.e., [0 1 0], [1 1 0], [0 0 0] and [0 1 1] thus providing **robustness** w.r.t the value of Y.

2.2. The PRISM approach

We consider the video coding problem now. Let $\mathbf{X}$ denote the current macro-block to be encoded (e.g., $\mathbf{X}$ is a vector of size 256 if macroblocks of size 16×16 are chosen) . Let $\mathbf{Y}$ denote the best (motion-compensated) predictor for $\mathbf{X}$ in the previous frame and let $\mathbf{Y} = \mathbf{X} + \mathbf{N}$ (We model $\mathbf{X}$, $\mathbf{N}$ as independent Laplacian random vectors.). We first encode $\mathbf{X}$ in the intra-coding mode to come up with the quantized codeword for $\mathbf{X}$. Now, using the insight from the above example, we find a channel code that is matched to the "correlation noise" $\mathbf{N}$, and use that to partition the quantized codeword space of $\mathbf{X}$. We can thus expect to approach the compression performance of predictive coding incurring only the complexity of intra-coding at the encoder. This is the main intuition behind the PRISM approach.

Note that, unlike the example presented above, in the video case we are dealing with real-valued sources (in the transform domain) and potentially unbounded correlation noises. Thus while perfect decoding was possible in the example (zero decoding error probability), there is, in general, a non-zero probability of decoding error in our case. This can be addressed by a combination of detection and concealment strategies.

3. PRISM: IMPLEMENTATION

3.1. Encoding

In this section we present a block-based implementation of the PRISM approach. The video frame to be encoded is first divided into non-overlapping spatial blocks (we choose blocks of size 16×16 or 8×8.).

1. **Transform Coding:** Each block is first transformed using the two-dimensional DCT. This is done so as to more easily exploit spatial correlations.

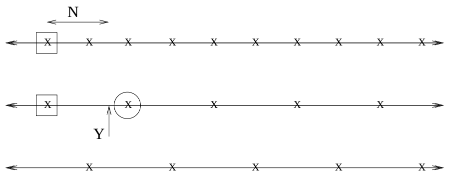


Fig. 3. The top line shows the quantized codewords for X. The bottom two lines show the two partitions of the quantized codeword space of X. The box shows the observed codeword which lies in the first partition. Since the magnitude of N is large the decoder decodes the circled codeword and makes a decoding error.

2. **Scalar Quantization:** The DCT coefficients are quantized with a step size proportional to standard deviation of $\mathbf{N}$. If a very fine step size is chosen to encode $\mathbf{X}$, then there can be decoding errors, since the codewords will be too "close" so that the side information $\mathbf{Y}$ cannot disambiguate them correctly [1].

3. **Syndrome Encoding:** Now the space of quantized codewords which has been appropriately generated using the statistics of $\mathbf{N}$ can be partitioned using a Euclidean space trellis channel code [7] analogous to the repetition channel code used to partition the source codeword space in the example in Section 2.1. In our particular implementation, we use a memory-7 rate-$1/2$ trellis code from [5].

4. **Refinement Quantization:** A target reconstruction quality corresponds to a particular quantization step size. When the coefficients that are syndrome encoded are quantized, the choice of the base quantization step size is limited by $\mathbf{N}$. This is done so as to minimize the probability of decoding error. To attain the target quantization step size, the coefficients need to be re-quantized further. This is accomplished in the refinement quantization stage.

[1]This is illustrated through the example in Figure 3. Here the top line shows the quantized codeword set for $\mathbf{X}$, and the two bottom lines show the partition of the space of quantized codewords. The rectangular box shows the observed codeword which lies in the first partition. Since the magnitude of $\mathbf{N}$ is more than the quantization step size, the decoder uses the side information $\mathbf{Y}$ to decode the incorrect (circled) codeword.

3.2. Decoding

1. **Syndrome Decoding:** All the sequences that are labeled by the received syndrome can be represented on a trellis. The Viterbi algorithm [5] can be used on the 128-state rate-$1/2$ trellis to identify the correct sequence from the from the set of candidate sequences that is "nearest" to the candidate predictor. The PRISM framework allows for flexibility in the method of choice of the candidate predictor.

2. **Estimation and Reconstruction:** Once the quantized codeword sequence is recovered, it is used along with the predictor to obtain the best reconstruction of the source. Any of the sophisticated signal processing algorithms (e.g., spatio-temporal interpolation) or post processing mechanisms can be deployed in this framework and these can only serve to improve the overall performance.

3. **Inverse Transform:** The transformed coefficients are then inverted using the inverse transform so as to give reconstructed pixels.

4. SIMULATION RESULTS

In this section, we present some preliminary simulation results that illustrate the various features of PRISM. The current implementation of our coder operates well in the high quality (PSNR of the order of 30 dB) regime. The extension to lower bit rates is a bit more involved, and is a part of the ongoing work.

We tested our implementation of the PRISM codec for sequences such as Football (352x240), Euronews (320x240), Foreman (176x144) and Carphone (176x144). The first two sequences are associated with high motion content. The reference system is an implementation of the H.263+ [8] video coder obtained from University of British Columbia, Vancouver. The first frame in all cases is encoded in the intra mode and the remaining frames are encoded in the non-intra mode. From a compression standpoint, subjectively the PRISM reconstructed quality is visually indistinguishable from that of the standards-based reference system, and objectively, it performs within 3-3.5 dB (in the current preliminary implementation) in PSNR of the standard. One reason for this inconsistency between PSNR and visual quality might be the fact that the bulk of PRISM's loss of performance can be attributed to decoding failure in a visually undetectable small fraction of the image blocks that can be easily concealed but which nonetheless incur a significant PSNR cost.

We also conducted preliminary tests on the robustness of the proposed PRISM framework. For both PRISM and the reference system, we introduced a frame loss by removing the second frame in the video sequence from the frame memory. This while the third frame is encoded off the second frame, it is decoded off the first frame. This leads to drift which accumulates and propagates in the H.263+ case. In contrast, the decoded quality is only moderately affected in PRISM and drift does not occur. Figure 4 compares the decoded visual quality for the Football sequence using PRISM and H.263+. Figures 4 (a) and (c) show respectively the decoded third and the fourteenth frames for the PRISM paradigm and Figures 4 (b) and (d) for the H.263+ coder. There is practically no drop in quality for PRISM while in the case of H.263+ the drop in quality is very significant leading to glaring visual artifacts (see Figures 4 (b) and 4 (d)). Similar observations were made when we conducted this robustness test on other sequences. These experiments clearly illustrate the inherent robustness of PRISM.

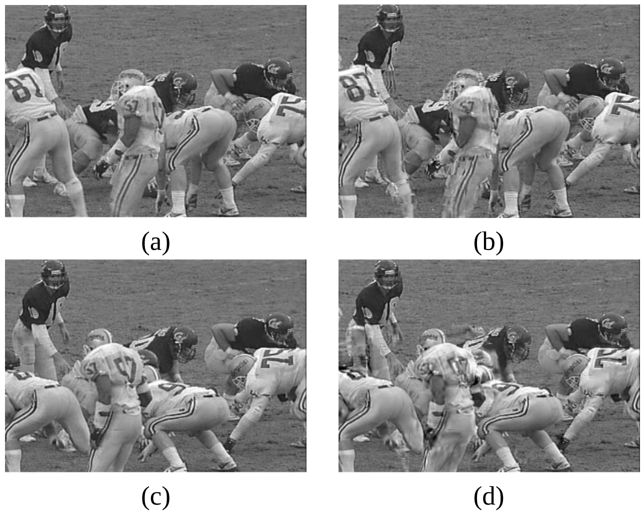


Fig. 4. Performance of PRISM and H.263+ coder in the case of frame loss. Fifteen frames of the football video sequence were encoded in both cases and the second decoded frame was removed from the frame memory in both cases. The third frame was decoded using the first frame as side information for the proposed paradigm and a predictor for H.263+. Figures 4 (a) and (c) show respectively the decoded third and the fourteenth frames for PRISM. Figures 4 (b) and (d) show the same for the H.263+ coder. We see in Figure 4 (b) that displeasing visual artifacts arise because of the drift and Figure 4 (d) shows that they propagate for the remainder of the sequence. In particular, the jersey number of the football player with jersey 57 cannot be seen in Figure 4 (d) while it is fairly clear in Figure 4 (c).

5. CONCLUSIONS AND FURTHER WORK

We have introduced PRISM – a novel, low encoding complexity, high performance and robust video coding paradigm. Under this paradigm, the encoding and the decoding complexities are roughly swapped with respect to the conventional paradigm resulting in a "light" encoder "heavy" decoder architecture.

Our present implementation of the framework, although promising, is far from complete and can be substantially enriched.Part of our ongoing work includes extending the PRISM paradigm to lower bit-rates/qualities and also reducing the gap between the compression performance of conventional video codecs and the our implementation of the PRISM framework.

6. REFERENCES

[1] R. Puri and K. Ramchandran, "PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles," *40th Allerton Conference on Communication, Control and Computing*, October, Allerton, IL, 2002.

[2] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471–480, July 1973.

[3] A. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, January 1976.

[4] S. S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction," *Proceedings of the Data Compression Conference (DCC)*, March, Snowbird, UT, 1999.

[5] G. D. Forney, "Coset Codes-Part I: Introduction and Geometrical Classification," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1123–1151, September 1988.

[6] F. J. Macwilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, Elseiver-North-Holland, 1977.

[7] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, pp. 55–67, January 1982.

[8] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–66, November 1998.