Compression Algorithms for Flexible Video Decoding

Ngai-Man Cheung and Antonio Ortega

Signal and Image Processing Institute, Univ. of Southern California, Los Angeles, CA

ABSTRACT

We investigate compression techniques to support *flexible video decoding*. In these, encoders generate a *single* compressed bit-stream that can be decoded in *several* different ways, so that users or decoders can choose among several available decoding paths. Flexible decoding has several advantages, including improved accessibility of the compressed data for emerging applications (e.g., multiview video) and enhanced robustness for video communication. Flexible decoding, however, makes it difficult for compression algorithms to exploit temporal redundancy: when the decoder can choose among different decoding paths, the encoder no longer knows deterministically which previously reconstructed frames will be available for decoding the current frame. Therefore, to support flexible decoding, encoders need to operate under uncertainty on the decoder predictor status. This paper extends our previous work on video compression with decoder predictor uncertainty using distributed source coding (DSC). We present a thorough discussion of flexible decoding, including its theoretical performance. The main advantage of a DSC approach to flexible decoding is that the information communicated from the encoder to the decoder (namely, the parity bits) is independent of a specific predictor. By "decoupling" the compressed information from the predictor, we will demonstrate that, theoretically and experimentally, DSC can lead to a solution that compares favorably to one based on conventional "closed loop" prediction (CLP), where multiple prediction residues are sent, one for each possible predictor available at the decoder. The main novelties of the proposed algorithm are that it incorporates different macroblock modes and significance coding within the DSC framework. This, combined with a judicious exploitation of correlation statistics, allows us to achieve competitive coding performance. Experimental results using multiview video coding and forward/backward video playback suggest the proposed DSC-based solution can outperform flexible decoding techniques based on CLP coding.

Keywords: Flexible video decoding, distributed source coding, distributed video coding, multiview video, robust video transmission

1. INTRODUCTION

In this paper we investigate video compression algorithms to support *flexible decoding* for a number of emerging applications.[1] Flexibility in this context means that video frames can be decoded in several different *orders*, while still exploiting redundancy between successively *decoded* frames (e.g., temporal or cross-view redundancy)*. The decoding order is decided only at the time of decoding, so that a choice among several available decoding paths can be made depending on the users' preferences or the operating conditions. We focus on coding tools to generate a *single* compressed bit-stream that can be decoded in *several* different ways, i.e., we assume it is not possible to request at decoding time (via feedback) coded data matching the chosen decoding order.

Flexible decoding can be useful for several applications. Notably, it improves the *accessibility* of the compressed data, which is important for several emerging applications and for some novel imagery datasets.[2] For example, some emerging multiview video applications such as free viewpoint TV[3,4] aim to enable users to play back different views and to switch between different views during playback, and to facilitate these free viewpoint switchings, it is desirable for the compressed multiview video data to be decodable in several different orders, corresponding to different view switching scenarios[5] (Figure 1). As another example, emerging video applications which support forward and backward frame-by-frame playback can benefit from compression schemes that allow both forward and backward decoding[6] (Figure 2).

Further author information:

Ngai-Man Cheung, E-mail: ncheung@usc.edu

Antonio Ortega, E-mail: ortega@sipi.usc.edu

*A trivial approach to enable flexible decoding would be to encode every frame independently, as an Intra frame.

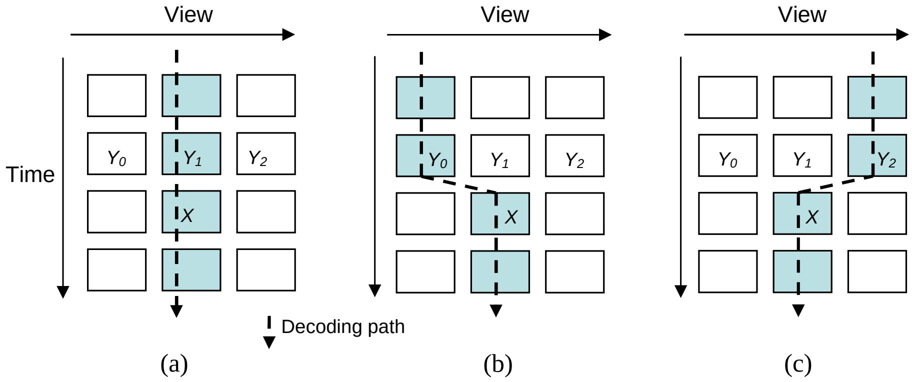


Figure 1. Multiview video applications - viewpoint switching may require a compression scheme to support several different decoding orders: (a) users stay in the same view during playback; (b), (c) users switch between adjacent views during playback.

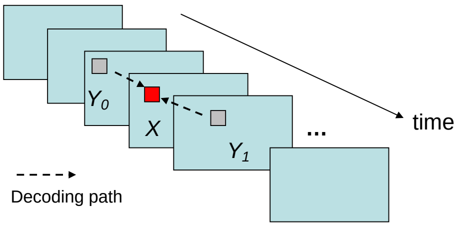


Figure 2. Forward and backward frame-by-frame playback.

Moreover, flexible decoding can be used to achieve more *robust* video communications, in applications where some reference frames may be corrupted during transmission. If a compression scheme can support multiple decoding paths decoder would be able to recover the current frame using any of the error-free references (Figure 3).[7]

State-of-the-art video coding algorithms exploit redundancy between neighboring frames to achieve compression. Flexible decoding makes it difficult to exploit this kind of interframe redundancy because decoders can choose different decoding paths, each leading to a different set of previously decoded frames. Thus at the time of encoding there will be uncertainty about which frames can be used to predict the current frame (as there is no guarantee that those same frames will be available at decoding time). For example, in multiview video applications, depending on whether the user continues requesting the same view as in Figure 1(a), or switches views as in Figure 1(b) or Figure 1(c), either the previous reconstructed frame of the same view (Y_1) or that of another view (Y_0 or Y_2) would be available as predictor for decoding the current frame X. However, since it is up to the users to choose among different decoding paths, the encoder would not know exactly which reconstructed frames will be available for decoding X. Similarly, in a forward/backward video playback application, either the "past" or the "future" reconstructed frame will be available at the decoder to serve as the predictor, depending

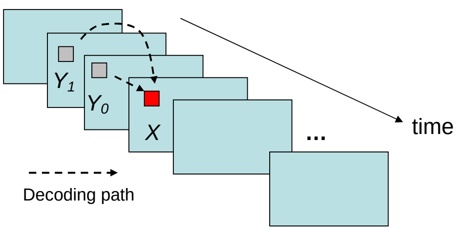


Figure 3. Robust video transmission using multiple decoding paths.

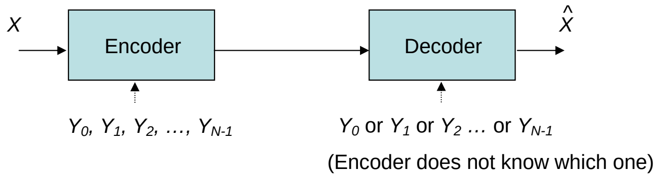


Figure 4. Problem formulation for flexible decoding. Either one of the candidate predictors $Y_0, Y_1, ..., Y_{N-1}$ will be present at the decoder, but encoder does not know which one.

on whether the data is being played back in the forward or backward direction (Figure 2). Since users can choose to play back in either direction, encoder would not know which reconstructed frame will be available at the decoder. Similar scenario can also arise in low delay video communication, where feedback could be infeasible. In these cases encoder may not have any information regarding which reference frames have arrived at the decoder error-free. Therefore encoder could not know exactly which reference frames will be available for decoding the current frame (Figure 3). In short, flexible decoding, while desirable, results in uncertainty on the predictor status at decoder.

Figure 4 depicts the general formulation of the flexible decoding problem. When compressing an input source X (the current video frame), encoder has access to a number of correlated sources $Y_0, Y_1, ..., Y_{N-1}$ (previously decoded video frames) to serve as predictors for encoding X. Here each Y_k is associated with a possible decoding path. However, of these predictor candidates, only *one* will be available at the decoder depending on the decoding path taken by the decoder. Crucially, since the encoder does not have any information regarding the chosen decoding path, it *does not* know which Y_k will be used at the decoder. Our goal is to investigate coding algorithms such that encoder can operate under this kind of uncertainty about predictor status at the decoder.

In order to support flexible decoding within a conventional closed-loop predictive (CLP) framework, e.g., motion-compensated predictive (MCP) video coding systems such as MPEG, H.26X, the encoder may send all the possible prediction residues $\{Z_i; i = 0 \text{ to } N-1\}$ to the decoder, where $Z_i = X - Y_i$ (following the notations in Figure 4), so that X can be recovered no matter which Y_i is available at the decoder. Each Z_i would correspond to a P-frame in these video coding standards. Note that it is indeed necessary for the encoder to communicate all the N possible prediction residues to the decoder. This is because, in CLP, a prediction residue would be "tied" to a specific predictor. For example, if Y_k is the available predictor at the decoder, then we can only use Z_k during the decoding process to recover X without causing significant mismatch. Therefore, in the cases of predictor uncertainty, the encoder would need to send multiple prediction residues. And there are two potential issues with the CLP approach, however. First, coding performance is degraded because multiple prediction residues are included in the bitstream. Specifically, the overhead to support flexible decoding would increase with the number of candidate predictors (or the number of possible decoding paths). Second, this approach may cause drifting. This is because, in practical video compression systems, the encoder would send the quantized versions of Z_i, $\hat{Z}_i$, to the decoder. Therefore, the reconstructed sources $\hat{X}_i = \hat{Z}_i + Y_i$ would be slightly different when different Y_i are used as predictors. Drifting may occur when $\hat{X}_i$ is used as reference for decoding future frames.

The H.264 video compression standard has defined SP- and SI-frames to support functionalities such as random access or error recovery that were originally supported by I-frames.[8] Essentially SP-frames follow the CLP coding approach we just discussed, but with modifications such that $\hat{X}_i$ can be identically reconstructed from different Y_i's using its corresponding Z_i (here Z_i corresponds to a primary or secondary SP-frame). This is achieved by using a different prediction loop from those in conventional P-frames (e.g., SP-frames compute the prediction residue in the transform domain whereas P-frames would compute it in the pixel domain[8]). However, this causes some penalty in coding performance, and the compression efficiency of SP-frames is in general worse than that of P-frames.[8] To support flexible decoding, different SP-frames bits (each corresponding to a different Y_i) need to be generated and sent to the decoder, similar to CLP coding, and therefore, H.264 SP-frames would incur a comparable amount of overhead as that in CLP coding. It should be noted that most H.264 SP-frame applications assume the availability of feedback from the decoder (e.g., Zhou et al.[9]), so that the encoder does

Table 1. Compare DSC-based low-complexity encoding and flexible decoding.

	DSC-based low-complexity encoding[13,14]	**DSC-based flexible decoding**
Key objective	Low complexity video encoding for mobile video, video sensors, etc.	Generate a single bitstream to support multiple decoding paths for forward/backward video playback, multiview video, video transmission, etc.
Encoder complexity	Most target applications require low-complexity, real-time encoding.	Not primary issue. Most target applications may use off-line encoding.
Encoder access to side information	SI not accessible by encoder due to complexity constraint.	Encoder has access to all the SI candidates. However, the exact one to be used at decoder is unknown to encoder.

know which predictor is available at the decoder and transmits only one of the Z_i. In short, H.264 SP-frames could be inefficient to support flexible decoding when there is no feedback.

In this paper, we extend our previous work[1] on flexible video decoding using a distributed source coding (DSC) approach.[10,11] In particular, this paper discusses the theoretically achievable performance using various approaches (intra coding, CLP and DSC) to address flexible decoding, and presents empirical data to illustrate how the performances may vary in several different applications. Our proposed DSC-based algorithm addresses the scenario where the encoder has access to all predictors, Y_k, which will play the role of side information (SI) at the decoder, but there is uncertainty as to which one will be used for decoding. One of the main challenges for DSC-based applications has proven to be achieving competitive compression efficiency.[12] To address this challenge, our proposed algorithm incorporates novel macroblock modes and significance coding into the DSC framework. This, along with careful exploitation of correlation statistics, allows to achieve significant performance improvements. Using multiview video and forward/backward video playback as examples, we demonstrate the proposed algorithm can outperform, in terms of coding efficiency, techniques based on CLP coding such as those as discussed above. Moreover, the proposed algorithm incurs only a small amount of drifting. In particular, DSC-coded macroblocks lead to the same reconstruction no matter which predictor candidate Y_k is used.

DSC has been studied extensively for enabling low-complexity video encoding, e.g., Puri et al.,[13] Aaron et al..[14] However, there are significant differences between low-complexity encoding and flexible decoding, as summarized in Table 1, which will lead us to a different solution. DSC has also been proposed to address compression of image-based rendering data/light fields to provide random access.[15,16] This prior work, however, assumes that the encoder has knowledge of predictor status at decoder, notably through using feedback, while in our case the encoder needs to operate with unknown predictor status. Recent work by Wang et al.[7] has proposed a DSC-based approach to address the problem of robust video transmission by allowing a video block to be decoded using more than one predictor blocks. While the general philosophy is similar to ours, different assumptions are made. In particular, this work assumes the encoder knows the probability that each predictor will be used, as determined by the packet erasure probability (whereas we assume all predictors are equally-likely to be used). This information is exploited to reduce the coding rate. In addition, the specific tools used are different from those proposed here. Recent work by Naman et al.[17] has proposed to enhance decoding flexibility and accessibility using intra coding and conditional replenishment. Our previous work[6] has also proposed to apply DSC to enable forward/backward video playback. The proposed algorithm in this paper is, however, considerably different and significantly more efficient. Among the key improvements are the introduction of macroblock modes and significance coding, a different approach to exploit the correlation between source and side-information, a different way to partition the input symbols and estimate the source bit's conditional probability, and a minimum MSE dequantization.

This paper is organized as follows. In Section 2 we discuss how DSC can address flexible decoding. A comparison of theoretically achievable performances is provided in Section 3. Section 4 we present the proposed compression algorithm. Section 5 presents the experimental results and Section 6 concludes the work.

2. FLEXIBLE DECODING BASED ON DSC: INTUITION

In conventional CLP coding, the encoder computes a prediction residual $Z = X - Y$, between source X and predictor Y, and communicates Z to the decoder (Figure 5(a)). DSC approaches the same compression problem taking a "virtual communication channel" perspective.[12, 18] Specifically, X is viewed as an input to a channel with *correlation noise* Z, and Y as the output of the channel (Figure 5(b)). Therefore, to recover X from Y, encoder would send *parity information* to the decoder. That is, in DSC, encoder would communicate X using parity information. Significantly, the parity information is independent of a specific Y being observed: the parity information is computed entirely from X taking into account the statistics of Z [†]. In particular, what matters in the DSC approach is, analogous to data communication, the *amount* of parity information corresponding to the statistics of Z, and decoder will be able to recover X as long as a sufficient amount of parity information has been communicated. In short, in DSC, the information communicated from encoder to decoder is independent of a specific Y, and this is in contrast to CLP, where encoder would communicate to decoder the prediction residue, which is tied to a specific Y.

To understand how DSC can tackle flexible decoding with N predictor candidates, consider N virtual channels, each corresponding to a predictor candidate Y_i (Figure 5(c)). Each channel is characterized by the correlation noise $Z_i = X - Y_i$. In order to recover X from any of these channels, the encoder would need to send an amount of parity sufficient for *all* the channels. In particular, the encoder would need to transmit enough parity information to allow decoding of the *worst-case* Z_i. Doing so, X can be recovered no matter which Y_i is available at the decoder. Note that encoder only needs to know the statistics of all the Z_i to determine the amount of parity information, and this is feasible since X and all Y_i are accessible at encoder in our problem formulation. In particular, the encoder does not need to know which Y_i is actually present at decoder. Since parity information is independent of a specific Y_i, the same parity information generated based on the worst case Z_i can be used to communicate X no matter which Y_i is available at the decoder.

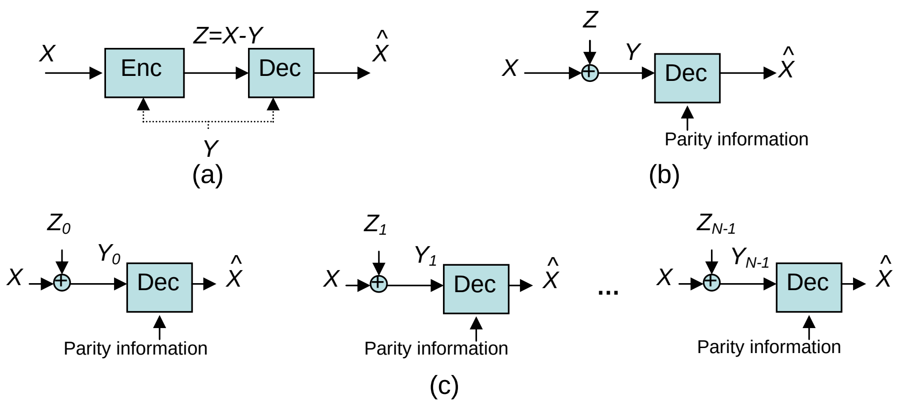


Figure 5. Compression of input source X: (a) CLP coding; (b) DSC from the virtual channel perspective; (c) DSC approach to the flexible decoding problem.

3. THEORETICAL PERFORMANCE

In this section we compare the theoretical performance of CLP and DSC in a flexible decoding scenario. We consider the compression of a discrete i.i.d. scalar source X under the scenario depicted in Figure 4. The predictor candidates Y_i, $i = 0$ to $N - 1$ are discrete i.i.d. scalar sources such that $X = Y_i + Z_i$, $Y_i \perp Z_i$, where Z_i are the discrete i.i.d. scalar prediction residues. As previously discussed, in CLP the overhead to address flexible decoding increases with the number of predictors N, while in DSC the overhead depends mainly on the

[†]As will be discussed in Section 4, we encode the bit-plane representation of X using DSC, and parity information is computed by XOR-ing a subset of bits in the bit-planes of X.

worst-case correlation noise. Specifically, in the CLP approach, all the residues $\{Z_i; i = 0 \text{ to } N-1\}$ would have to be sent to the decoder, which would require an information rate

$$R_{CLP} = \sum_{i=0}^{N-1} H(Z_i) \tag{1}$$

theoretically. This amount of information is necessary because, in CLP, the prediction residue is tied to a specific predictor, and therefore, under predictor uncertainty conditions, multiple residues each corresponding to a different predictor candidate would be required to be communicated to the decoder, so that severe mismatch and drifting can be avoided.

On the other hand, in the DSC approach, the information rate required to communicate X with Y_i at the decoder is $H(X|Y_i)$, and using $X = Y_i + Z_i$ and $Y_i \perp Z_i$, we have $H(X|Y_i) = H(Z_i)$. Under the scenario of side-information uncertainty, we would need to communicate X at a rate[‡]

$$R_{DSC} = \max_i H(X|Y_i) = \max_i H(Z_i). \tag{2}$$

It is clear that the encoder needs to communicate at least $\max_i H(X|Y_i)$ bits so that X can be recovered for whichever Y_i available at decoder. To show that $\max_i H(X|Y_i)$ is indeed achievable, we use the *source networks* approach proposed by Csiszar et al..[19, 20] A source network is a graphical abstraction of a multiterminal source coding problem involving information sources, encoders, and destinations located at its vertices. Each encoder operates on the input messages from the sources connected to it, and the resulting codeword is made available to all destinations connected to the encoder through noiseless communication channels. Each destination must be able to reproduce accurately the messages from certain specified sources based on the received codewords. In particular, the source networks in Csiszar et al.[19] focus on (i) discrete memoryless sources, (ii) graphs in which no edge joins two sources or two encoders or two destinations, and (iii) graphs in which the destinations are required to reproduce the messages of the specified sources with small probability of error (i.e., lossless data compression). Figure 6(a) shows an example of source network representing the Slepian-Wolf problem. For certain subclass of source network, Csiszar et al.[19] derived the exponential error bounds which are tight in a neighborhood of the boundary of the achievable rate region. In addition, these bounds were shown to be *universally attainable*, i.e., they are attainable by encoders and decoders not depending on the source statistics. Subsequently, Csiszar[20] further showed that the error exponents can be attained universally by linear code.

To model the predictor uncertainty in flexible decoding, we use N source networks each corresponding to a different predictor candidate as depicted in Figure 6(b). Following from the result in Csiszar et al.[19, 20] that the Slepian-Wolf's achievable rate region is universally attainable, the *same* codes can be used to communicate X in any of these source networks at an achievable rate $H(X|Y_i)$. Therefore, at a rate of $\max_i H(X|Y_i)$, the codes can be used to communicate X regardless of which Y_i is at the decoder. From (1) and (2), we have $R_{CLP} \geq R_{DSC}$. Therefore, the DSC approach can potentially achieve better coding performance. Figures 7 and 8 depict some empirical data of $H(X)$ (i.e., intra coding), $\sum_i H(Z_i)$ (i.e., R_{CLP}) and $\max_i H(Z_i)$ (i.e., R_{DSC}) in different applications and show how these quantities could vary with the number of predictors.

4. PROPOSED ALGORITHMS

Figure 9 depicts the proposed video encoding algorithms to address flexible decoding based on DSC.[1]

4.1. Motion estimation and macroblock classification

Each macroblock (MB) M in the current frame first undergoes standard motion estimation (and disparity estimation in the case of multiview video application) w.r.t. each candidate reference frame f_i, and the corresponding motion information (one per reference frame, f_i) is included in the bitstream, i.e., the encoder sends N motion vectors to the decoder. Denote A_i the best motion-compensated predictor for M obtained in f_i. If the difference between M and A_i is sufficiently small, M may be classified to be in skip mode w.r.t. f_i (Figure 9). In that

[‡]To be more precise, R_{CLP} and R_{DSC} are the best (minimum) achievable rates.

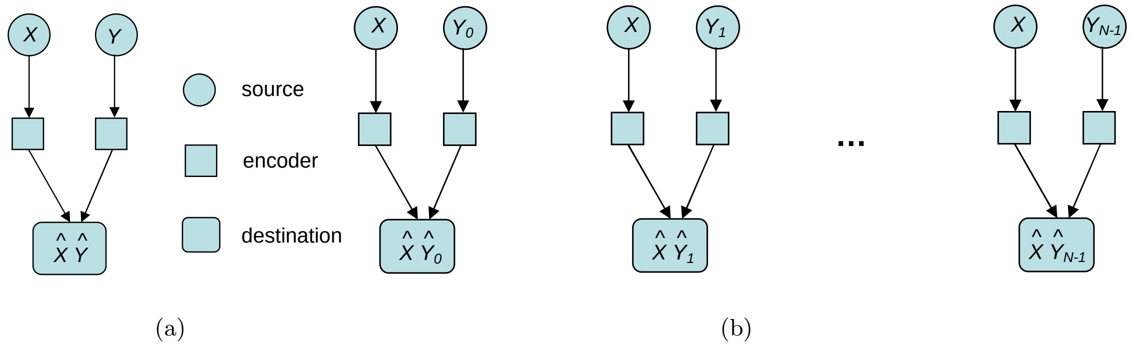


Figure 6. Source networks: (a) Source network of Slepian-Wolf.[10, 19] Csiszar et al.[19, 20] suggest an achievable rate $H(X|Y)$ for communicating X is universally attainable. (b) Source networks of flexible decoding with predictor uncertainty. The same universal codes can be used to communicate X in any of these networks at a rate $H(X|Y_i)$.

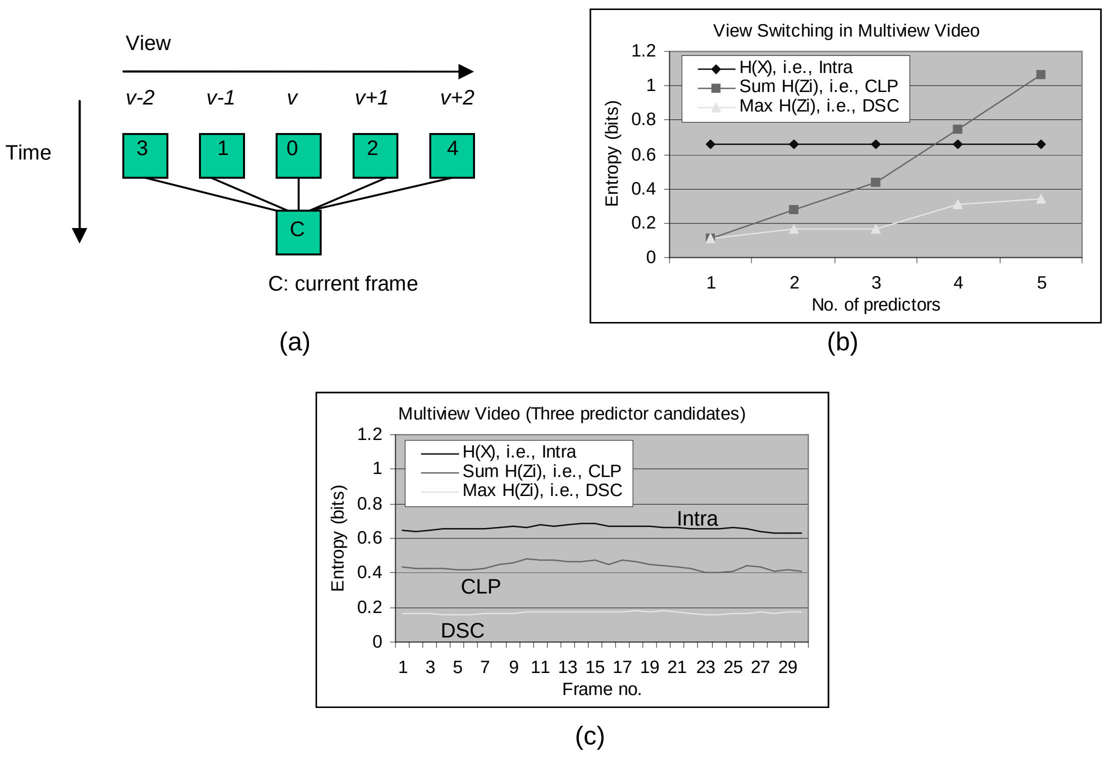


Figure 7. Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Multiview video coding as in Figure 1. (a) Previously reconstructed frames of neighboring views are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Akko&Kayo view 28-th. (c) Entropy of each frame in the case of three predictor candidates.

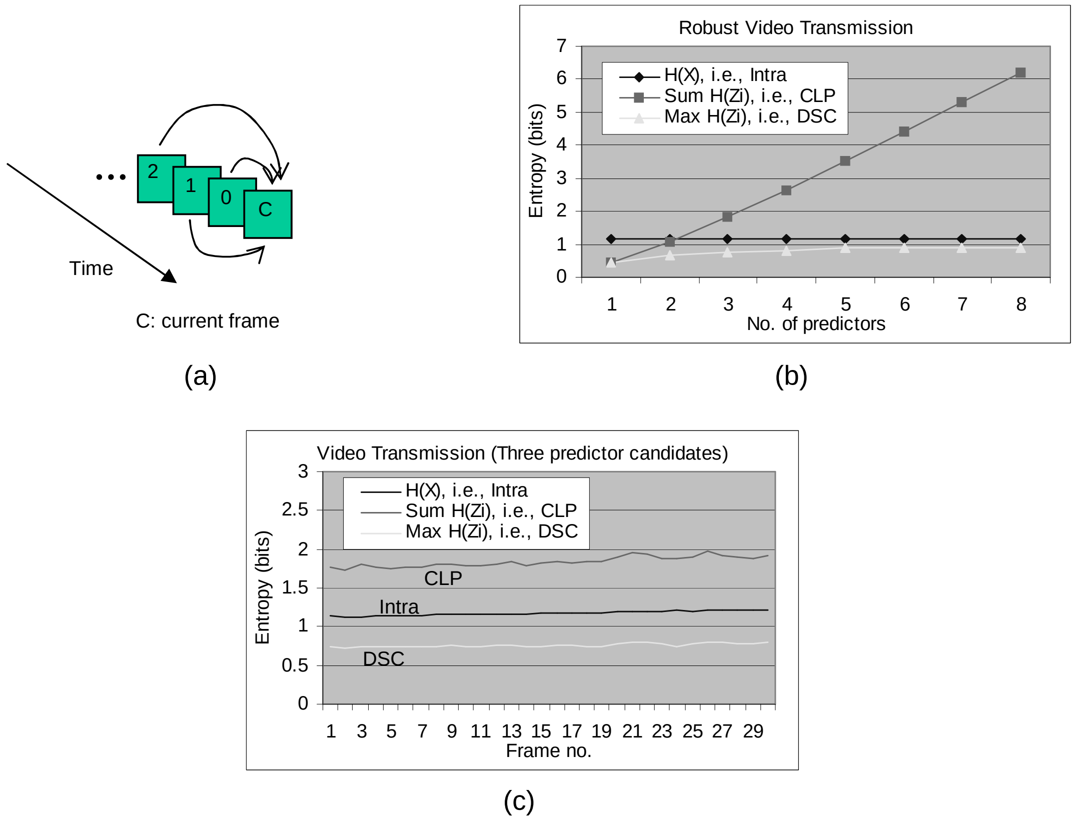


Figure 8. Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Robust video transmission as in Figure 3. (a) Past reconstructed frames are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Coastguard. (c) Entropy of each frame in the case of three predictor candidates.

case, since the encoder can skip some prediction residues and the encoder does not need to communicate all N residues, the overhead of including multiple prediction residues using CLP could be small. Specifically, in our current implementation, each macroblock M can be in either skip mode or a non-skip mode. A macroblock will be encoded using the skip mode when, out of all of the N residue blocks between M and A_i, $0 \leq i \leq N-1$, at least one of the residue blocks is a zero block after quantization (i.e., all the quantized transform coefficients in the block are zero). In skip mode M is encoded using conventional CLP coding (similar to standard H.26X algorithms) w.r.t. the candidate reference frames which do not have skipping. However, majority of the macroblocks will be classified into the non-skip mode and be encoded using DSC following the steps discussed in the next section.

Note that choosing between CLP and DSC for a given macroblock can be achieved using rate-distortion (RD) based mode selection (as in H.264): The RD costs of CLP and DSC are computed and the one achieving the minimum RD cost is selected. Such RD optimized mode decision algorithm can achieve a better coding performance, at the expense of requiring higher encoding complexity. In our comparison with H.263 (Section 5) we did not use this RD optimized mode decision. As will be discussed, we implemented our proposed algorithms mainly based on H.263 coding tools.

4.2. Direct coefficient coding (DCC)

For those macroblocks M to be encoded with DSC, we first apply standard 8×8 DCT to the pixel data to obtain the vector of transform coefficients X, and we then quantize X to obtain the quantization index W (Figure 10). This is similar to intra-frame coding in standard H.26X algorithms. Denote Y_i the DCT coefficient

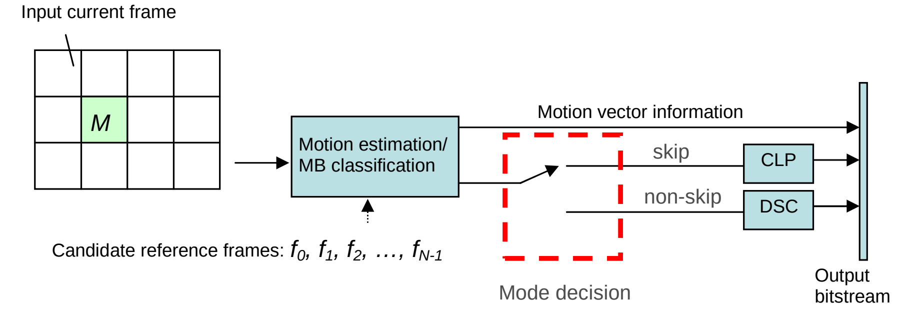


Figure 9. Proposed encoding algorithm to encode an macroblock M of the current frame.

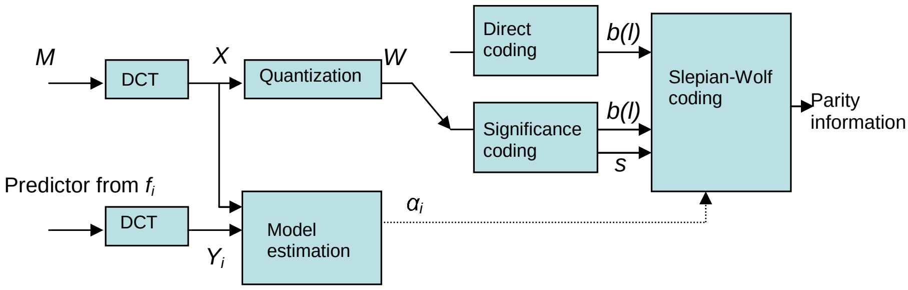


Figure 10. Encoding macroblock M using DSC.

in A_i corresponding to W (recall A_i is the best motion-compensated predictor from each f_i). We compress W by exploiting its correlation with the worst case Y_i, so that it can be recovered with any Y_i that may be present at the decoder. Specifically, based on a correlation model between W and Y_i (to be discussed in Section 4.5), the encoder can estimate the coding rates needed to communicate W when Y_i is available at the decoder. Then the encoder communicates W by sending an amount of parity information equal to the maximum of these estimated coding rates. Since both W and Y_i are available at the encoder in our problem, the correlation model can be readily estimated.

The quantized values of the K lowest frequency DCT coefficients (along a zig-zag scan order) are encoded with a *direct* coefficient coding (DCC), and for the rest we use a *significant* coefficient coding (SCC). In DCC, we form the k-th frequency coefficient vector by grouping together the k-th $(0 \leq k \leq K-1)$ frequency coefficients from all the 8×8 blocks in a frame (except those in skip modes). Then each of these vectors is converted into a bit-plane representation, and the bit-planes are passed to a Slepian-Wolf (SW) coder, where inter-frame correlation is exploited to compress the bit-planes losslessly.

4.3. Significant coefficient coding (SCC)

The quantized values of the k-th highest frequency coefficients, $k \geq K$, are encoded using SCC. Specifically, we first use a *significance bit* s to signal if the quantized value of a coefficient is zero $(s = 0)$ or not $(s = 1)$, so that only the value of a non-zero coefficient needs to be sent to the decoder. The significance bits for the k-th frequency coefficients from all the 8×8 blocks in a frame (except those in skip modes) are grouped together to form a significance bit-plane to be compressed by the SW coder. On the other hand, the non-zero coefficients are grouped together to form coefficient vectors where all the DCT frequencies are combined, as we found that the correlation statistics of non-zero coefficients are similar at different frequencies.

SCC is introduced as an alternative to DCC to reduce the number of source bits to be handled in SW coding. Specifically, assume DCC leads to L_k bitplanes for the the k-th frequency coefficient vector. Therefore, each k-th frequency coefficient contributes L_k source bits in DCC, regardless of whether the coefficient is zero or not. On the other hand, with SCC, a zero coefficient contributes one source bit (significance bit), while a non-zero coefficient contributes approximately $1 + L_k$ bits. If p_k is the probability that the k-th frequency coefficient will be zero, then the expected number of source bits using SCC is

$$1 \times p_k + (1 + L_k) \times (1 - p_k), \tag{3}$$

and SCC can lead to rate savings (compared with DCC) if the expected number of bits using SCC, i.e., (3), is less than that of DCC, i.e., L_k, or equivalently if

$$p_k > \frac{1}{L_k} \tag{4}$$

holds. Therefore, SCC can achieve rate savings when coefficients are likely to be zero. In the experiment, we use $K = 3$ (where SCC starts) determined using (4) and some statistics of the video sequences.

4.4. Bit-plane compression

Bit-planes extracted from the K coefficient vectors produced in DCC along with those produced in SCC are compressed by a SW coder, starting from the most significant bit-planes. Denote a bit in the bit-plane at l-th level of significance by a binary r.v. $b(l)$, where $l = 0$ corresponding to the least significant level. That is, $b(l)$ is the l-th significant bit in the quantization index W. A binary r.v. $b(l)$ is to be compressed using Y_i and decoded bits $b(l+1), b(l+2), ...$ as side information. Specifically, this is performed by a low density parity check (LDPC) based SW encoder, which computes the *syndrome* bits from the original bit-planes and sends them to the decoder.[21]

4.5. Model and conditional probability estimation

SW decoding needs the conditional probability $p(b(l)|Y_i, b(l+1), b(l+2), ...)$ estimated from SI to aid recovering $b(l)$. The probability can be estimated as follows. The encoder estimates the conditional p.d.f. $f_{X|Y_i}(x|y_i)$ for each coefficient vector and for each candidate predictor. Assuming a system model $X = Y_i + Z_i$, and under the assumption of independence of Y_i and Z_i, we have

$$f_{X|Y_i}(x|y_i) = f_{Z_i}(x - y_i) \tag{5}$$

We assume Z_i is Laplacian distributed, i.e., $f_{Z_i}(z_i) = \frac{1}{2}\alpha_i e^{-\alpha_i|z_i|}$, and estimate the model parameters α_i at the encoder using maximum likelihood estimation (MLE) and send to the decoder. Note that in flexible decoding problem, the encoder can access to all the candidate SIs. Therefore, the model parameters can be readily estimated. This is not the case in typical DSC applications, where there are constraints on accessing side-information at the encoder making model estimation a non-trivial problem.[22]

Given all the model parameters α_i, the decoder can estimate the conditional probability for any particular Y_i available at decoder using the following procedure (Figure 11). Denote $\tilde{W}$ the numerical value of the concatenation of the sequence of the decoded bits $b(l+1), b(l+2), ...$, i.e., $\tilde{W} = b(l+1) \times 2^0 + b(l+2) \times 2^1 + ...$ Given the decoded bits, the quantization index W can range only from $\tilde{W} \times 2^{l+1}$ to $\tilde{W} \times 2^{l+1} + 2^{l+1} - 1$. When $W \in [W_r, W_s]$, $b(l) = 0$, and when $W \in [W_t, W_u]$, $b(l) = 1$, where W_r, W_s, W_t, W_u are given by (in the cases when $\tilde{W} \geq 0$):

$$\begin{aligned} W_r &= \tilde{W} \times 2^{l+1}; \\ W_s &= \tilde{W} \times 2^{l+1} + 2^l - 1; \\ W_t &= \tilde{W} \times 2^{l+1} + 2^l; \\ W_u &= \tilde{W} \times 2^{l+1} + 2^{l+1} - 1. \end{aligned} \tag{6}$$

Equations for $\tilde{W} < 0$ are similar. Therefore, the decoder can estimate the probabilities that $b(l)$ will be zero and one by integrating $f_{X|Y_i}(x|y_i)$ over the intervals $[X_r, X_s]$ and $[X_t, X_u]$ respectively, where $[X_r, X_s]$ is the inverse quantization mapping of $[W_r, W_s]$, and $[X_t, X_u]$ is that of $[W_t, W_u]$.

Note that each Y_i exhibits different levels of correlation with respect to $b(l)$. To ensure that $b(l)$ can be recovered with any of the predictor candidates Y_i, the encoder sends R syndrome bits to the decoder, where $R = \max\ R_i$, and R_i is number of syndrome bits required to recover $b(l)$ when Y_i is used as predictor. By doing so, each bit-plane can be exactly recovered no matter which Y_i is available at the decoder, and therefore, W can be losslessly recovered and X reconstructed to the same value when any of the Y_i is used as predictor. This eliminates drifting in DSC-coded macroblock.

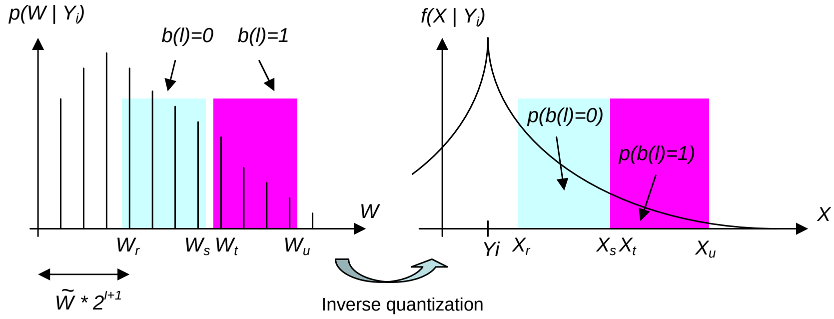


Figure 11. Estimate the conditional probability $p(b(l)|Y_i, b(l+1), b(l+2), ...)$.

5. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results. We first discuss our experiments on multiview video coding (MVC). Here we generate compressed multiview bit-streams that allow switching from the adjacent views as in Figure 1. Therefore, there are three predictor candidates. We compare the coding performance using the following algorithms to generate the bit-stream: (i) intra coding using H.263 I-frames; (ii) CLP approach with each of the three residues encoded using H.263 P-frames; (iii) proposed DSC-based algorithm, with H.263 half-pel motion estimation and quantization. Since we implement all the schemes using the same (H.263) coding tools (e.g., half-pixel accuracy motion estimation) the comparison is fair. We compare the schemes using MVC sequences Akko&Kayo and Ballroom, which are in 320×240 and encoded at 30fps and 25fps respectively. Figures 12 and 13 show the comparison results. As shown in the figures, the proposed algorithm outperforms CLP and intra coding, with about 1dB gain in the medium/high picture quality range (33-36dB). We also compare the approaches in terms of drifting by simulating a scenario where viewpoint switching from the $(v-1)$-th view to the v-th view occurs at frame number 2. Figure 14 compares the PSNR of the reconstructed frames within the GOP with that of the non-switching case, where the v-th view is being played back throughout the GOP. As shown in the figure, while CLP (using P-frame) may cause considerable amount of drifting, the proposed algorithm is almost drift-free, since the quantized coefficients in DSC coded macroblock are identically reconstructed.

We also experiment how the coding performance of the proposed system scales with the number of predictor candidates. In this experiment, the temporally/spatially adjacent reconstructed frames are used as predictor candidates following the order depicted in Figure 15. As shown in Figure 16, the bit-rate of DSC-based solution increases at a much slower rate compared with that of the CLP counterpart. It is because, with the DSC approach, an additional predictor candidate would cause a bit-rate increase (when coding a bit-plane) only if it has the worst correlation among all the predictor candidates (w.r.t. that bit-plane).

We then discuss our experiments on forward/backward playback application, where there are two predictor candidates (Figure 2). We compare our proposed algorithm with a CLP approach where both forward predicted H.263 P-frames and backward predicted H.263 P-frames are included. As discussed, such approach may incur drifting, since in general the reconstructed forward and backward predicted P-frames are not identical. We compare the schemes using sequences Coastguard and Stefan, which have considerable amounts of motion and picture details. Figures 17 and 18 show the comparison results. As shown in the figures, the proposed algorithm outperforms CLP and intra coding. We also show the results of "normal" H.263 inter-frame coding (i.e., including only forward prediction residue) with the same GOP sizes. Note that inter-frame coding cannot support flexible decoding. The results are shown here for reference only.

We also compare the approaches in terms of drifting with the following experiment: in forward decoding, a backward predicted frame is used for frame number 1 and as a reference for decoding the following frame. This is similar to what would happen when decoding direction switches from backward to forward. As shown in the results in Figure 19, the proposed algorithm incurs a negligible amount of drifting.

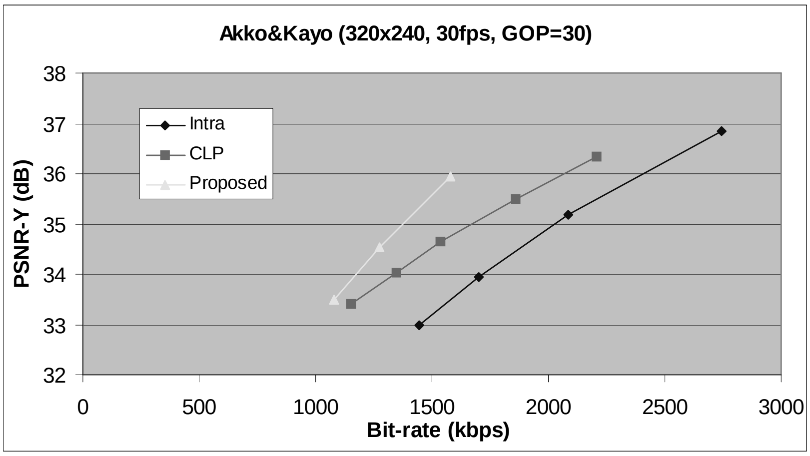


Figure 12. Simulation results of multiview video coding: Akko&Kayo. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 27th-29th) arbitrarily chosen from all the available views.

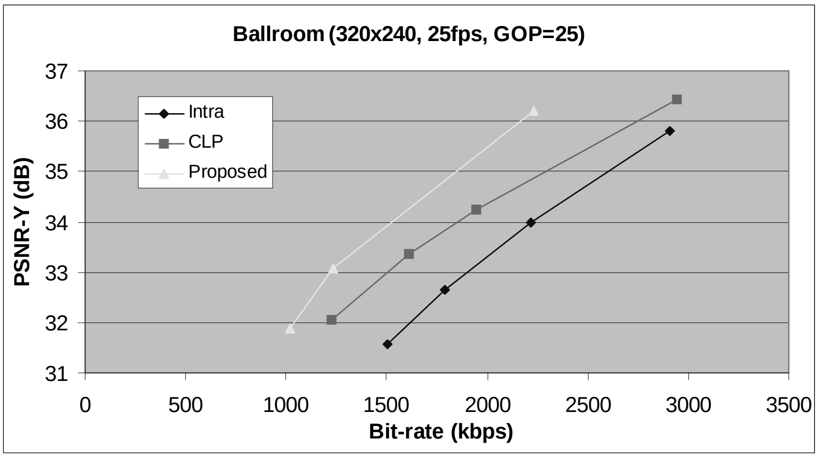


Figure 13. Simulation results of multiview video coding: Ballroom. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 3rd-5th) arbitrarily chosen from all the available views.

6. CONCLUSIONS AND FUTURE WORK

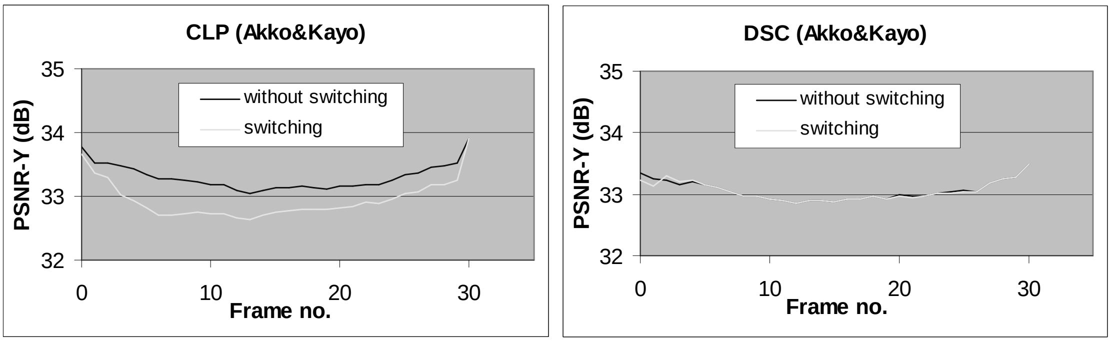
We have proposed a video compression algorithm to support flexible decoding, based on DSC. The proposed algorithm integrates macroblock mode and significance coding to improve coding performance. Simulation results using MVC and forward/backward video playback demonstrate the proposed DSC-based algorithm can outperform the CLP approach, while incurring only a negligible amount of drifting. Future work includes investigating improved model estimation methods.

ACKNOWLEDGEMENT

The work was supported in part by NASA-JPL. The authors would like to thank Matthew Klimesh of NASA-JPL for pointing out the theoretical results by Csiszar *et al.*.

REFERENCES

1. N.-M. Cheung and A. Ortega, "Flexible video decoding: A distributed source coding approach," in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2007.



(a) (b)

Figure 14. Drifting experiment using Akko&Kayo view 28th: (a) CLP; (b) DSC. GOP size is 30 frames. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.

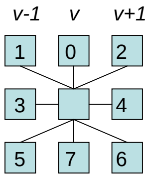


Figure 15. Scaling experiment - temporally and spatially adjacent reconstructed frames are used as predictor candidates following the depicted order.

2. A. Ortega, "Video coding: Predictions are hard to make, especially about the future," in *Proc. Image Media Processing Symposium*, 2007.
3. ISO/IEC JTC1/SC29/WG11, "Introduction to multiview video coding," Tech. Rep. N7328, MPEG document, 2005.
4. M. Tanimoto, "FTV (free viewpoint television) creating ray-based image engineering," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2005.
5. N.-M. Cheung and A. Ortega, "Distributed source coding application to low-delay free viewpoint switching in multiview video compression," in *Proc. Picture Coding Symposium (PCS)*, 2007.

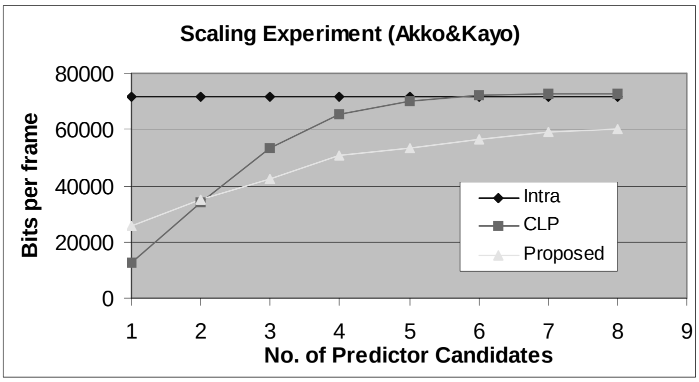


Figure 16. Scaling experiment using Akko&Kayo view 29th. The PSNR of the different schemes are comparable – Intra: 35.07dB, CLP: 34.78dB, DSC:34.79dB.

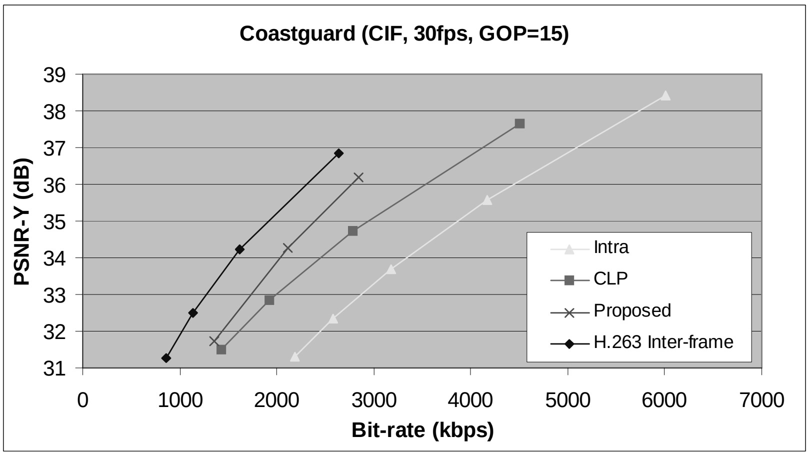


Figure 17. Simulation results of forward/backward video playback: Coastguard. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.

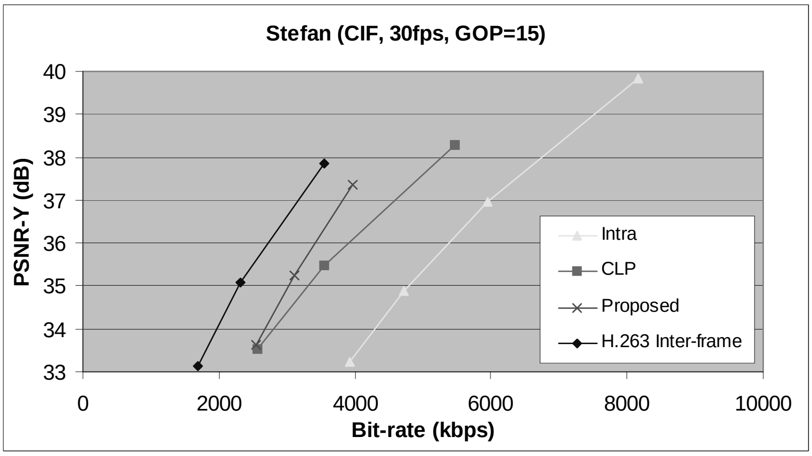
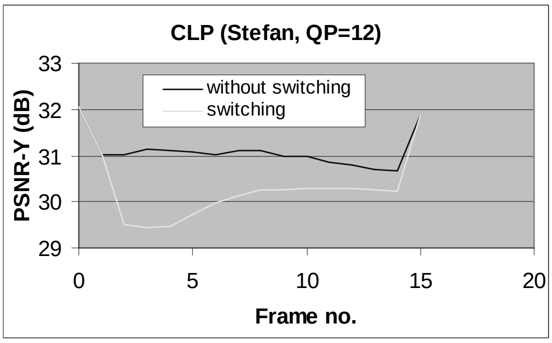
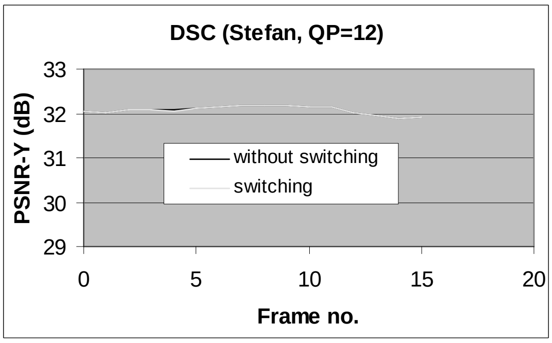


Figure 18. Simulation results of forward/backward video playback: Stefan. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.

6. N.-M. Cheung, H. Wang, and A. Ortega, "Video compression with flexible playback order based on distributed source coding," in *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
7. J. Wang, V. Prabhakaran, and K. Ramchandran, "Syndrome-based robust video transmission over networks with bursty losses," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2006.
8. M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology* **13**, July 2003.
9. X. Zhou, W.-Y. Kung, and C.-C. J. Kuo, "A robust H.264 video streaming scheme for portable devices," in *Proc. Int'l Symp. Circuits and Systems*, 2005.
10. D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Information Theory* **19**, pp. 471–480, July 1973.
11. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. Data Compression Conference (DCC)*, 1999.
12. B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery* **93**, pp. 71–83, Jan. 2005.



(a)



(b)

Figure 19. Drifting experiment using Stefan sequence: (a) CLP; (b) DSC. The figure shows the PSNR of the reconstructed frames in the first GOP. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.

13. R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conf. Communications, Control, and Computing*, Oct. 2002.
14. A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2002.
15. A. Jagmohan, A. Sehgal, and N. Ahuja, "Compressed of lightfield rendered images using coset codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.
16. A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," in *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2004.
17. A. Naman and D. Taubman, "A novel paradigm for optimized scalable video transmission based on JPEG2000 with motion," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2007.
18. Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine* **21**, pp. 80–94, Sept. 2004.
19. I. Csiszar and J. Korner, "Towards a general theory of source networks," *IEEE Trans. Information Theory* **26**, pp. 155–165, Mar. 1980.
20. I. Csiszar, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Information Theory* **28**, pp. 585–592, July 1982.
21. A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters* **6**, Oct. 2002.
22. N.-M. Cheung, H. Wang, and A. Ortega, "Correlation estimation for distributed source coding under information exchange constraints," in *Proc. Int'l Conf. Image Processing (ICIP)*, 2005.