

# A Framework for Adaptive Scalable Video Coding Using Wyner-Ziv Techniques

Huisheng Wang, Ngai-Man Cheung, and Antonio Ortega

*Integrated Media Systems Center and Department of Electrical Engineering, USC Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089-2564, USA*

Received 27 March 2005; Revised 31 August 2005; Accepted 12 September 2005

This paper proposes a practical video coding framework based on distributed source coding principles, with the goal to achieve efficient and low-complexity scalable coding. Starting from a standard predictive coder as base layer (such as MPEG-4 baseline video coder in our implementation), the proposed Wyner-Ziv scalable (WZS) coder can achieve higher coding efficiency, by selectively exploiting the high quality reconstruction of the previous frame in the enhancement layer coding of the current frame. This creates a multi-layer Wyner-Ziv prediction “link,” connecting the same bitplane level between successive frames, thus providing improved temporal prediction as compared to MPEG-4 FGS, while keeping complexity reasonable at the encoder. Since the temporal correlation varies in time and space, a block-based adaptive mode selection algorithm is designed for each bitplane, so that it is possible to switch between different coding modes. Experimental results show improvements in coding efficiency of 3–4.5 dB over MPEG-4 FGS for video sequences with high temporal correlation.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

## 1. INTRODUCTION

Scalable coding is well suited for video streaming and broadcast applications as it facilitates adapting to variations in network behavior, channel error characteristics, and computation power availability at the receiving terminal. Predictive coding, in which motion-compensated predictors are generated based on previously reconstructed frames, is an important technique to remove temporal redundancy among successive frames. It is well known that predictive techniques increase the difficulty of achieving efficient scalable coding because scalability leads to multiple possible reconstructions of each frame [1]. In this situation, either (i) the same predictor is used for all layers, which leads to either drift or coding inefficiency, or (ii) a different predictor is obtained for each reconstructed version and used for the corresponding layer of the current frame, which leads to added complexity. MPEG-2 SNR scalability with a single motion-compensated prediction loop and MPEG-4 FGS exemplify the first approach. MPEG-2 SNR scalability uses the enhancement-layer (EL) information in the prediction loop for both base and enhancement layers, which leads to drift if the EL is not received. MPEG-4 FGS provides flexibility in bandwidth adaptation and error recovery because the enhancement layers are coded in “intra-” mode, which results in low coding efficiency especially for sequences that exhibit high temporal correlation.

Rose and Regunathan [1] proposed a multiple motion-compensated prediction loop approach for general SNR scalability, in which each EL predictor is optimally estimated by considering all the available information from both base and enhancement layers. Several alternative multilayer techniques have also been proposed to exploit the temporal correlation in the EL inside the FGS framework [2–4]. They employ one or more additional motion-compensated prediction loops to code the EL, for which a certain number of FGS bitplanes are included in the EL prediction loop to improve the coding efficiency. Traditional closed-loop prediction (CLP) techniques have the disadvantage of requiring the encoder to generate all possible decoded versions for each frame, so that each of them can be used to generate a prediction residue. Thus, the complexity is high at the encoder, especially for multilayer coding scenarios. In addition, in order to avoid drift, the exact same predictor has to be used at both the encoder and decoder.

Distributed source coding techniques based on network information theory provide a different and interesting viewpoint to tackle these problems. Several video codecs using side information (SI) at the decoder [5–10] have been recently proposed within the Wyner-Ziv framework [11]. These can be thought of as an intermediate step between “closing the prediction loop” and coding each frame independently. In closed-loop prediction, in order for the encoder to generate a residue it needs to generate the same

predictor that will be available at the decoder. Instead, a Wyner-Ziv encoder only requires the *correlation structure* between the current signal and the predictor. Thus there is no need to generate the decoded signal at the encoder as long as the correlation structure is known or can be found.

Some recent work [12–15] has addressed the problem of scalable coding in the distributed source coding setting. Steinberg and Merhav [12] formulated the theoretical problem of successive refinement of information in the Wyner-Ziv setting, which serves as the theoretical background of our work. In our work, we target the application of these principles to actual video coding systems. The two most related recent algorithms are in the works by Xu and Xiong [13] and Sehgal et al. [14]. There are a number of important differences between our approach and those techniques. In [13], the authors presented a scheme similar to MPEG-4 FGS by building the bitplane ELs using Wyner-Ziv coding (WZC) with the current base and more significant ELs as SI, ignoring the EL information of the previous frames. In contrast, our approach explores the remaining temporal correlation between the successive frames in the EL using WZC to achieve improved performance over MPEG-4 FGS. In [14], multiple redundant Wyner-Ziv encodings are generated for each frame at different fidelities. An appropriate encoded version is selected for streaming, based on the encoder’s knowledge of the predictor available at the decoder. This scheme requires a feedback channel and additional delay and thus it is not well suited for broadcast or low-delay applications. In short, one method [13] ignores temporal redundancy in the design, while the other [14] creates separate and redundant enhancement layers rather than a single embedded enhancement layer. In addition to these approaches for SNR scalability, Tagliasacchi et al. [15] have proposed a spatial and temporal scalable codec using distributed source coding. They use the standards-conformant H.264/AVC to encode the base layer, and a syndrome-based approach similar to [6] to encode the spatial and temporal enhancement layers. Motion vectors from the base layer are used as coarse motion information so that the enhancement layers can obtain a better estimate of the temporal correlation. In contrast, our work focuses on SNR scalability.

We propose, extending our previous work [16, 17], an efficient solution to the problem of scalable predictive coding by recasting it as a Wyner-Ziv problem. Our proposed technique achieves scalability without feedback and exploits both spatial and temporal redundancy in the video signal. In [16], we introduced the basic concept on a first-order DPCM source model, and then presented a preliminary version of our approach in video applications in [17]. Our approach, Wyner-Ziv scalable coding (WZS), aims at applying in the context of Wyner-Ziv the CLP-based estimation-theoretic (ET) technique in [1]. Thus, in order to reduce the complexity, we do not explicitly construct multiple motion-compensation loops at the encoder, while, at the decoder, SI is constructed to combine spatial and temporal information in a manner that seeks to approximate the principles proposed in [1]. In particular, starting from a standard CLP base-layer (BL) video coder (such as MPEG-4 in our

implementation), we create a multilayer Wyner-Ziv prediction “link,” connecting the same bitplane level between successive frames. The decoder generates the enhancement-layer SI with either the estimation theoretic approach proposed in [1] or our proposed simplified switching algorithm to take into account all the available information to the EL. In order to design channel codes with appropriate rates, the encoder estimates the correlation between the current frame and its enhancement-layer SI available at the decoder. By exploiting the EL information from the previous frames, our approach can achieve significant gains in EL compression, as compared to MPEG-4 FGS, while keeping complexity reasonably low at the encoder.

A significant contribution of our work is to develop a framework for integrating WZC into a standard video codec to achieve efficient and low-complexity scalable coding. Our proposed framework is backward compatible with a standard base-layer video codec. Another main contribution of this work is to propose two simple and efficient algorithms to explicitly estimate at the encoder the parameters of a model to describe the correlation between the current frame and an optimized SI available only at the decoder. Our estimates closely match the actual correlation between the source and the decoder SI. The first algorithm is based on constructing an estimate of the reconstructed frame and directly measuring the required correlations from it. The second algorithm is based on an analytical model of the correlation structure, whose parameters the encoder can estimate.

The paper is organized as follows. In Section 2, we briefly review the theoretical background of successive refinement for the Wyner-Ziv problem. We then describe our proposed practical WZS framework and the correlation estimation algorithms in Sections 3 and 4, respectively. Section 5 describes the codec structure and implementation details. Simulation results are presented in Section 6, showing substantial improvement in video quality for sequences with high temporal correlation. Finally, conclusions and future work are provided in Section 7.

## 2. SUCCESSIVE REFINEMENT FOR THE WYNER-ZIV PROBLEM

Steinberg and Merhav [12] formulated the theoretical problem of successive refinement of information, originally proposed by Equitz and Cover [18], in a Wyner-Ziv setting (see Figure 1). A source  $X$  is to be encoded in two stages: at the coarse stage, using rate  $R_1$ , the decoder produces an approximation  $\hat{X}_1$  with distortion  $D_1$  based on SI  $Y_1$ . At the refinement stage, the encoder sends an additional  $\Delta R$  refinement bits so that the decoder can produce a more accurate reconstruction  $\hat{X}_2$  with a lower distortion  $D_2$  based on SI  $Y_2$ .  $Y_2$  is assumed to provide a better approximation to  $X$  than  $Y_1$  and to form a Markov chain  $X \rightarrow Y_2 \rightarrow Y_1$ . Let  $R_{X|Y}^*(D)$  be the Wyner-Ziv rate-distortion function for coding  $X$  with SI  $Y$ . A source  $X$  is successively refinable if [12]

$$R_1 = R_{X|Y_1}^*(D_1), \quad R_1 + \Delta R = R_{X|Y_2}^*(D_2). \quad (1)$$

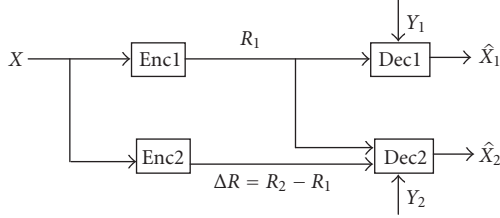


FIGURE 1: Two-stage successive refinement with different side information  $Y_1$  and  $Y_2$  at the decoders, where  $Y_2$  has better quality than  $Y_1$ , that is,  $X \rightarrow Y_2 \rightarrow Y_1$ .

Successive refinement is possible under a certain set of conditions. One of the conditions, as proved in [12], requires that the two SIs,  $Y_1$  and  $Y_2$ , be equivalent at the distortion level  $D_1$  in the coarse stage. To illustrate the concept of “equivalence,” we first consider the classical Wyner-Ziv problem (i.e., without successive refinement) as follows. Let  $Y$  be the SI available at the decoder only, for which a joint distribution with source  $X$  is known by the encoder. Wyner and Ziv [11] have shown that

$$R_{X|Y}^* = \min_U [I(X; U|Y)], \quad (2)$$

where  $U$  is an auxiliary random variable, and the minimization of mutual information between  $X$  and  $U$  given  $Y$  is over all possible  $U$  such that  $U \rightarrow X \rightarrow Y$  forms a Markov chain and  $E[d(X, f(U, Y))] \leq D$ . For the successive refinement problem,  $Y_2$  is said to be equivalent to  $Y_1$  at  $D_1$  if there exists a random variable  $U$  achieving (2) at  $D_1$  and satisfying  $I(U; Y_2|Y_1) = 0$  as well. In words, when  $Y_1$  is given,  $Y_2$  does not provide any more information about  $U$ .

It is important to note that this equivalence is unlikely to arise in scalable video coding. As an example, assume that  $Y_1$  and  $Y_2$  correspond to the BL and EL reconstruction of the previous frame, respectively. Then, the residual energy when the current frame is predicted based on  $Y_2$  will in general be lower than if  $Y_1$  is used. Thus, in general, this equivalence condition will not be met in the problem we consider and we should expect to observe a performance penalty with respect to a non-scalable system. Note that one special case where equivalence holds is that where identical SIs are used at all layers, that is,  $Y_1 = Y_2$ . For this case and for a Gaussian source with quadratic distortion measure, the successive refinement property holds [12]. Some practical coding techniques have been developed based on this equal SI property; for example, in the work of Xu and Xiong [13], where the BL of the current frame is regarded as the only SI at the decoder at both the coarse and refinement stages. However, as will be shown, constraining the decoder to use the same SI at all layers leads to suboptimal performance. In our work, the decoder will use the EL reconstruction of the previous frame as SI, outperforming an approach similar to that proposed in [13].

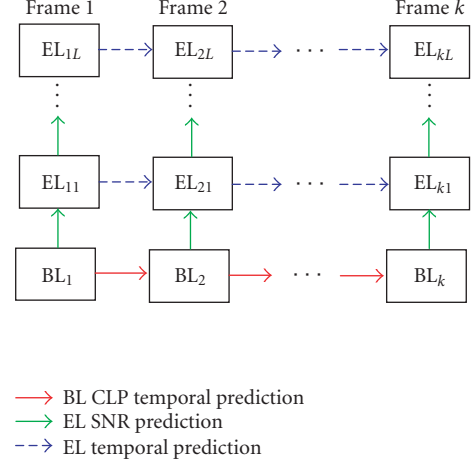


FIGURE 2: Proposed multilayer prediction problem.  $BL_i$ : the base layer of the  $i$ th frame.  $EL_{ij}$ : the  $j$ th EL of the  $i$ th frame, where the most significant EL bitplane is denoted by  $j = 1$ .

### 3. PROPOSED PREDICTION FRAMEWORK

In this section, we propose a practical framework to achieve the Wyner-Ziv scalability for video coding. Let video be encoded so that each frame  $i$  is represented by a base layer  $BL_i$ , and multiple enhancement layers  $EL_{i1}, EL_{i2}, \dots, EL_{iL}$ , as shown in Figure 2. We assume that in order to decode  $EL_{ij}$  and achieve the quality provided by the  $j$ th EL, the decoder will need to have access to (1) the previous frame decoded up to the  $j$ th EL,  $EL_{i-1,k}$ ,  $k \leq j$ , and (2) all information for the higher significance layers of the current frame,  $EL_{ik}$ ,  $k < j$ , including reconstruction, prediction mode, BL motion vector for each inter-mode macroblock, and the compressed residual. For simplicity, the BL motion vectors are reused by all EL bitplanes.

With the structure shown in Figure 2, a scalable coder based on WZC techniques would need to combine multiple SIs at the decoder. More specifically, when decoding the information corresponding to  $EL_{i,k}$ , the decoder can use as SI decoded data corresponding to  $EL_{i-1,k}$  and  $EL_{i,k-1}$ . In order to understand how several different SIs can be used together, we first review a well-known technique for combining multiple predictors in the context of closed-loop coding (Section 3.1 below). We then introduce an approach to formulate our problem as a one of source coding with side information at the decoder (Section 3.2).

#### 3.1. Brief review of ET approach [1]

The temporal evolution of DCT coefficients can be usually modelled by a first-order Markov process:

$$x_k = \rho x_{k-1} + z_k, \quad x_{k-1} \perp z_k, \quad (3)$$

where  $x_k$  is a DCT coefficient in the current frame and  $x_{k-1}$

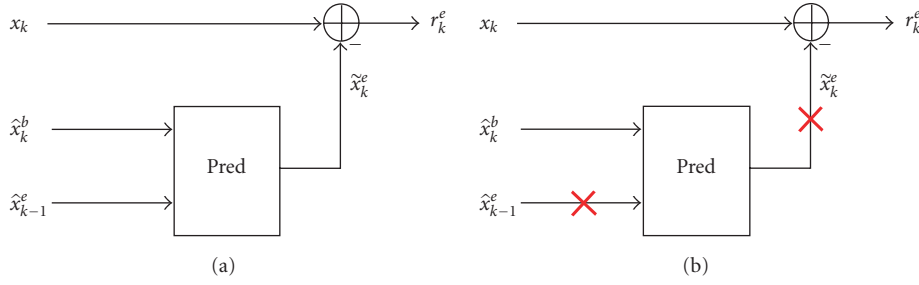


FIGURE 3: Basic difference at the encoder between the CLP techniques such as ET and our proposed problem: (a) CLP techniques, (b) our problem setting.

is the corresponding DCT coefficient in the previous frame after motion compensation. Let  $\hat{x}_k^b$  and  $\hat{x}_k^e$  be the base and enhancement-layer reconstruction of  $x_k$ , respectively. After the BL has been generated, we know that  $x_k \in (a, b)$ , where  $(a, b)$  is the quantization interval generated by the BL. In addition, assume that the EL encoder and decoder have access to the EL reconstructed DCT coefficient  $\hat{x}_{k-1}^e$  of the previous frame. Then the optimal EL predictor is given by

$$\begin{aligned} \tilde{x}_k^e &= E[x_k | \hat{x}_{k-1}^e, x_k \in (a, b)] \\ &\approx \rho \hat{x}_{k-1}^e + E[z_k | z_k \in (a - \rho \hat{x}_{k-1}^e, b - \rho \hat{x}_{k-1}^e)]. \end{aligned} \quad (4)$$

The EL encoder then quantizes the residual

$$r_k^e = x_k - \tilde{x}_k^e. \quad (5)$$

Let  $(c, d)$  be the quantization interval associated with  $r_k^e$ , that is,  $r_k^e \in (c, d)$ , and let  $e = \max(a, c + \tilde{x}_k^e)$  and  $f = \min(b, d + \tilde{x}_k^e)$ . The optimal EL reconstruction is given by

$$\hat{x}_k^e = E[x_k | \hat{x}_{k-1}^e, x_k \in (e, f)]. \quad (6)$$

The EL predictor in (4) can be simplified in the following two cases: (1)  $\tilde{x}_k^e \approx \hat{x}_k^b$  if the correlation is low,  $\rho \approx 0$ , or the total rate is approximately the same as the BL rate, that is,  $\hat{x}_{k-1}^e \approx \hat{x}_{k-1}^b$ ; and (2)  $\tilde{x}_k^e \approx \hat{x}_{k-1}^e$  for cases where temporal correlation is higher or such that the quality of the BL is much lower than that of the EL.

Note that in addition to optimal prediction and reconstruction, the ET method can lead to further performance gains if efficient context-based entropy coding strategies are used. For example, the two cases  $\tilde{x}_k^e \approx \hat{x}_k^b$  and  $\tilde{x}_k^e \approx \hat{x}_{k-1}^e$  could have different statistical properties. In general, with the predictor of (4), since the statistics of  $z_k$  tend to be different depending on the interval  $(a - \rho \hat{x}_{k-1}^e, b - \rho \hat{x}_{k-1}^e)$ , the encoder could use different entropy coding on different intervals [1]. Thus, a major goal in this paper is to design a system that can achieve some of the potential coding gains of conditional coding *in the context of a WZC technique*. To do so, we will

design a switching rule at the encoder that will lead to different coding for different types of source blocks.

### 3.2. Formulation as a distributed source coding problem

The main disadvantage of the ET approach for multilayer coding resides in its complexity, since multiple motion-compensated prediction loops are necessary for EL predictive coding. For example, in order to encode  $EL_{21}$  in Figure 2, the exact reproduction of  $EL_{11}$  must be available at the encoder. If the encoder complexity is limited, it may not be practical to generate all possible reconstructions of the reference frame at the encoder. In particular, in our work we assume that the encoder can generate *only* the reconstructed BL, and does not generate any EL reconstruction, that is, none of the  $EL_{ij}$  in Figure 2 are available at the encoder. Under this constraint, we seek efficient ways to exploit the temporal correlation between ELs of consecutive frames. In this paper, we propose to cast the EL prediction as a Wyner-Ziv problem, using Wyner-Ziv coding to replace the closed loop between the respective ELs of neighboring frames.

We first focus on the case of two-layer coders, which can be easily extended to multilayer coding scenarios. The basic difference at the encoder between CLP techniques, such as ET, and our problem formulation is illustrated in Figure 3. A CLP technique would compute an EL predictor:

$$\tilde{x}_k^e = f(\hat{x}_{k-1}^e, \hat{x}_k^b), \quad (7)$$

where  $f(\cdot)$  is a general prediction function (in the ET case,  $f(\cdot)$  would be defined as in (4)). Then, the EL encoder would quantize the residual  $r_k^e$  in (5) and send it to the decoder.

Instead, in our formulation, we assume that the encoder can only access  $\hat{x}_k^b$ , while the decoder has access to both  $\hat{x}_k^b$  and  $\hat{x}_{k-1}^e$ . Therefore, the encoder cannot generate the same predictor  $\tilde{x}_k^e$  as (7) and cannot explicitly generate  $r_k^e$ . Note, however, that  $\hat{x}_k^b$ , one of the components in (7), is in fact available at the encoder, and would exhibit some correlation with  $x_k$ . This suggests making use of  $\hat{x}_k^b$  at the encoder. First, we can rewrite  $r_k^e$  as

$$r_k^e = x_k - \tilde{x}_k^e = (x_k - \hat{x}_k^b) - (\tilde{x}_k^e - \hat{x}_k^b), \quad (8)$$

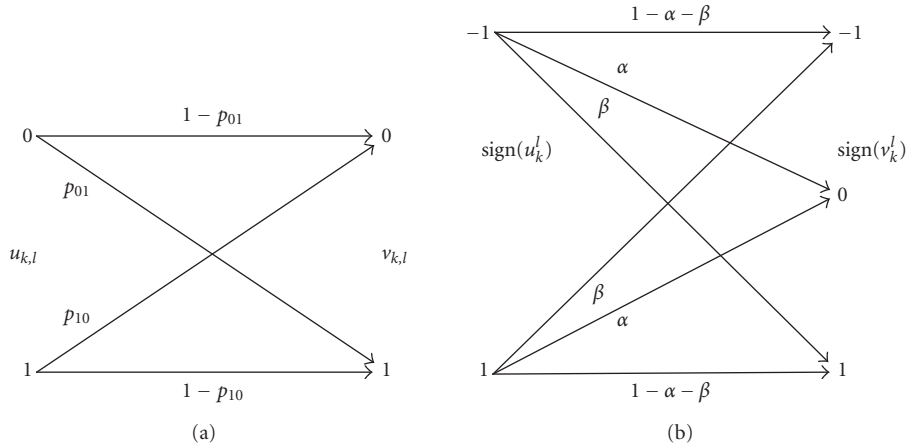


FIGURE 4: Discrete memoryless channel model for coding  $u_k$ : (a) binary channel for bitplanes corresponding to absolute values of frequency coefficients (i.e.,  $u_{k,l}$  at bitplane  $l$ ), (b) discrete memoryless channel with binary inputs (“-1” if  $u_k^l < 0$  and “1” if  $u_k^l > 0$ ) and three outputs (“-1” if  $v_k^l < 0$ , “1” if  $v_k^l > 0$ , and “0” if  $v_k^l = 0$ ) for sign bits.

and then to make explicit how this can be cast as a Wyner-Ziv coding problem, let  $u_k = x_k - \hat{x}_k^b$  and  $v_k = \hat{x}_k^e - \hat{x}_k^b$ . With this notation  $u_k$  plays the role of the input signal and  $v_k$  plays the role of SI available at the decoder only. We can view  $v_k$  as the output of a hypothetical communication channel with input  $u_k$  corrupted by correlation noise. Therefore, once the correlation between  $u_k$  and  $v_k$  has been estimated, the encoder can select an appropriate channel code and send the relevant coset information such that the decoder can obtain the correct  $u_k$  with SI  $v_k$ . Section 4 will present techniques to efficiently estimate the correlation parameters at the encoder.

In order to provide a representation with multiple layers coding, we generate the residue  $u_k$  for a frame and represent this information as a series of bitplanes. Each bitplane contains the bits at a given significance level obtained from the absolute values of all DCT coefficients in the residue frame (the difference between the base-layer reconstruction and the original frame). The sign bit of each DCT coefficient is coded once in the bitplane where that coefficient becomes significant (similar to what is done in standard bitplane-based wavelet image coders). Note that this would be the same information transmitted by an MPEG-4 FGS technique. However, differently from the intra-bitplane coding in MPEG-4 FGS, we create a multilayer Wyner-Ziv prediction link, connecting a given bitplane level in successive frames. In this way, we can exploit the temporal correlation between corresponding bitplanes of  $u_k$  and  $v_k$ , without reconstructing  $v_k$  explicitly at the encoder.

#### 4. PROPOSED CORRELATION ESTIMATION

Wyner-Ziv techniques are often advocated because of their reduced encoding complexity. It is important to note, however, that their compression performance depends greatly on the accuracy of the correlation parameters estimated at the

encoder. This correlation estimation can come at the expense of increased encoder complexity, thus potentially eliminating the complexity advantages of WZC techniques. In this section, we propose estimation techniques to achieve a good tradeoff between complexity and coding performance.

##### 4.1. Problem formulation

Our goal is to estimate the correlation statistics (e.g., the matrix of transition probabilities in a discrete memoryless channel) between bitplanes of same significance in  $u_k$  and  $v_k$ . To do so, we face two main difficulties. First, and most obvious,  $\hat{x}_{k-1}^e$ , and therefore  $v_k$ , are not generated at the encoder as shown in Figure 3. Second,  $v_k$  is generated at the decoder by using the predictor  $\hat{x}_k^e$  from (7), which combines  $\hat{x}_{k-1}^e$  and  $\hat{x}_k^b$ . In Section 4.2, we will discuss the effect of these combined predictors on the estimation problem, with a focus on our proposed mode-switching algorithm.

In what follows, the most significant bitplane is given the index “1,” the next most significant bitplane index “2,” and so on.  $u_{k,l}$  denotes the  $l$ th bitplane of absolute values of  $u_k$ , while  $u_k^l$  indicates the reconstruction of  $u_k$  (including the sign information) truncated to its  $l$  most significant bitplanes. The same notation will be used for other signals represented in terms of their bitplanes, such as  $v_k$ .

In this work, we assume the channel between the source  $u_k$  and the decoder SI  $v_k$  to be modeled as shown in Figure 4. With a binary source  $u_{k,l}$ , the corresponding bitplane of  $v_k$ ,  $v_{k,l}$ , is assumed to be generated by passing this binary source through a binary channel. In addition to the positive (symbol “1”) and negative (symbol “-1”) sign outputs, an additional output symbol “0” is introduced in the sign bit channel to represent the case when SI  $v_k = 0$ .

We propose two different methods to estimate crossover probabilities, namely, (1) a direct estimation (Section 4.3), which generates estimates of the bitplanes first, then directly measures the crossover probabilities for these estimated

bitplanes, and (2) a model-based estimation (Section 4.4), where a suitable model for the residue signal ( $u_k - v_k$ ) is obtained and used to estimate the crossover probabilities in the bitplanes. These two methods will be evaluated in terms of their computational requirements, as well as their estimation accuracy.

#### 4.2. Mode-switching prediction algorithm

As discussed in Section 3, the decoder has access to two SIs,  $\hat{x}_{k-1}^e$  and  $\hat{x}_k^b$ . Consider first the prediction function in (7) when both SIs are known. In the ET case,  $f(\cdot)$  is defined as an optimal prediction as in (4) based on a given statistical model of  $z_k$ . Alternatively, the optimal predictor  $\tilde{x}_k^e$  can be simplified to either  $\hat{x}_{k-1}^e$  or  $\hat{x}_k^b$  for a two-layer coder, depending on whether the temporal correlation is strong (choose  $\hat{x}_{k-1}^e$ ) or not (choose  $\hat{x}_k^b$ ).

Here we choose the switching approach due to its lower complexity, as compared to the optimal prediction, and also because it is amenable to an efficient use of “conditional” entropy coding. Thus, a different channel code could be used to code  $u_k$  when  $\tilde{x}_k^e \approx \hat{x}_k^b$  and when  $\tilde{x}_k^e \approx \hat{x}_{k-1}^e$ . In fact, if  $\tilde{x}_k^e = \hat{x}_k^b$ , then  $v_k = 0$ , and we can code  $u_k$  directly via entropy coding, rather than using channel coding. If  $\tilde{x}_k^e = \hat{x}_{k-1}^e$ , we apply WZC to  $u_k$  with the estimated correlation between  $u_k$  and  $v_k$ .

For a multilayer coder, the temporal correlation usually varies from bitplane to bitplane, and thus the correlation should be estimated at each bitplane level. Therefore, the switching rules we just described should be applied before each bitplane is transmitted. We allow a different prediction mode to be selected on a macroblock (MB) by macroblock basis (allowing adaptation of the prediction mode for smaller units, such as blocks or DCT coefficients, may be impractical). At bitplane  $l$ , the source  $u_k$  has two SIs available at the decoder:  $u_k^{l-1}$  (the reconstruction from its more significant bitplanes) and  $\hat{x}_{k-1}^e$  (the EL reconstruction from the previous frame). The correlation between  $u_k$  and each SI is estimated as the absolute sum of their difference. When both SIs are known, the following parameters are defined for each MB,

$$\begin{aligned} E_{\text{intra}} &= \sum_{\text{MB}_i} |u_k - u_k^{l-1}|, \\ E_{\text{inter}} &= \sum_{\text{MB}_i} |u_k - (\hat{x}_{k-1}^e - \hat{x}_k^b)| = \sum_{\text{MB}_i} |x_k - \hat{x}_{k-1}^e|, \end{aligned} \quad (9)$$

where only the luminance component is used in the computation. Thus, we can make the mode decision as follows: WZS-MB (coding of MB via WZS) mode is chosen if

$$E_{\text{inter}} < E_{\text{intra}}. \quad (10)$$

Otherwise, we code  $u_k$  directly via bitplane by bitplane refinement (FGS-MB) since it is more efficient to exploit spatial correlation through bitplane coding.

In general, mode-switching decisions can be made at either encoder or decoder. Making a mode decision at the decoder means deciding which SI should be used to decode WZC data sent by the encoder. The advantage of this approach is that all relevant SI is available. A disadvantage in this case is that the encoder has to estimate the correlation between  $u_k$  and  $v_k$  without exact knowledge of the mode decisions that will be made at the decoder. Thus, because it does not know which MBs will be decoded using each type of SI, the encoder has to encode all information under the assumption of a single “aggregate” correlation model for all blocks. This prevents the full use of conditional coding techniques discussed earlier.

Alternatively, making mode decisions at the encoder provides more flexibility as different coding techniques can be applied to each block. The main drawback of this approach is that the SI  $\hat{x}_{k-1}^e$  is not available at the encoder, which makes the mode decision difficult and possibly suboptimal. In this paper, we select to make mode decisions at the encoder, with mode switching decisions based on the estimated levels of temporal correlation. Thus  $E_{\text{inter}}$  cannot be computed exactly at the encoder as defined in (9), since  $\hat{x}_{k-1}^e$  is unknown; this will be further discussed once specific methods to approximate  $E_{\text{inter}}$  at the encoder have been introduced.

#### 4.3. Direct estimation

For the  $l$ th bitplane,  $1 \leq l \leq L$ , where  $L$  is the least significant bitplane level to be encoded, we need to estimate the correlation between  $u_{k,l}$  and  $v_k$  given all  $u_{k,j}$  ( $1 \leq j < l$ ) which have been sent to the decoder. While, in general, for decoding  $u_k$  all the information received by the decoder can be used, here, we estimate the correlation under the assumption that to decode bitplane  $l$ , we use only the  $l$  most significant bitplanes of the previous frame. The SI for bitplane  $l$  in this particular case is denoted by  $\check{v}_k(l)$ , which is unknown at the encoder.

We compute  $\bar{v}_k(l)$  at the encoder to approximate  $\check{v}_k(l)$ ,  $1 \leq l \leq L$ . Ideally we would like the following requirements to be satisfied: (1) the statistical correlation between each bitplane  $u_{k,l}$  and  $\check{v}_k(l)$ , given all  $u_{k,j}$  ( $1 \leq j < l$ ), can be well approximated by the corresponding correlation between  $u_{k,l}$  and  $\bar{v}_k(l)$ ; and (2)  $\bar{v}_k(l)$  can be obtained at the encoder in a simple way without much increased computational complexity. This can be achieved by processing the original reference frame  $x_{k-1}$  at the encoder. We first calculate the residual

$$s_k = x_{k-1} - \hat{x}_k^b \quad (11)$$

at the encoder, and then generate bitplanes  $s_k^l$  in the same way as the  $u_k^l$  are generated. Let  $\bar{v}_k(l) = s_k^l$  for  $1 \leq l \leq L$ . While  $\bar{v}_k(l)$  and  $\check{v}_k(l)$  are not equal, the correlation between  $\bar{v}_k(l)$  and  $u_{k,l}$  provides a good approximation to the correlation between  $\check{v}_k(l)$  and  $u_{k,l}$ , as seen in Figure 5, which shows the probability that  $u_k^l \neq s_k^l$  (i.e., the values of  $u_k$  and  $s_k$  do not fall into the same quantization bin), as well as the corresponding crossover probability between  $u_k$  and decoder SI  $\check{v}_k(l)$ . The crossover probability here is an indication of the correlation level.

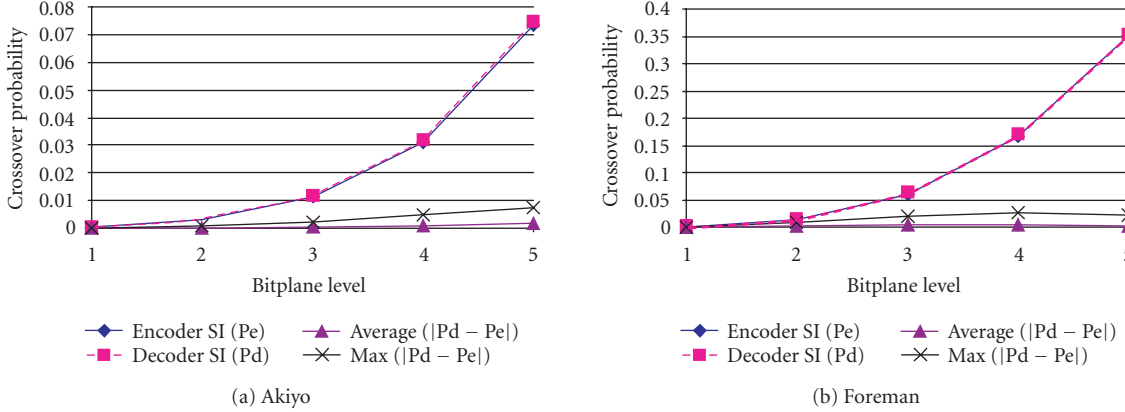


FIGURE 5: Measurement of approximation accuracy for *Akiyo* and *Foreman* sequences. The crossover probability is defined as the probability that the values of the source  $u_k$  and side information do not fall into the same quantization bin. The average and maximum absolute differences over all frames between the two crossover probabilities are also shown.

SI  $s_k^l$  can be used by the encoder to estimate the level of temporal correlation, which is again used to perform mode switching and determine the encoding rate of the channel codes applied to MBs in WZS-MB mode. Replacing the term  $(\hat{x}_{k-1}^e - \hat{x}_k^b)$  in (9) by  $s_k^l$ ,  $E_{\text{inter}}$  is redefined as

$$E_{\text{inter}} = \sum_{\text{MB}_i} |u_k - s_k^l|. \quad (12)$$

Clearly, the larger  $E_{\text{intra}}$ , the more bits will be required to refine the bitplane in FGS-MB mode. Similarly  $E_{\text{inter}}$  gives an indication of the correlation present in the  $i$ th MB between  $u_k^l$  and  $s_k^l$ , which are approximations of  $u_k$  and  $v_k$  at the  $l$ th bitplane, respectively. To code MBs in WZS-MB mode, we can further approximate the ET optimal predictor in (4) by taking into account both SIs,  $u_k^{l-1}$  and  $s_k^l$ , as follows: If  $s_k$  is within the quantization bin specified by  $u_k^{l-1}$ , the EL predictor is set to  $s_k^l$ ; however, if  $s_k$  is outside that quantization bin, the EL predictor is constructed by first clipping  $s_k$  to the closest value within the bin and then truncating this new value to its  $l$  most significant bitplanes. For simplicity, we still denote the improved EL predictor of the  $l$ th bitplane as  $s_k^l$  in the following discussion.

At bitplane  $l$ , the rate of the channel code used to code  $u_{k,l}$  (or the sign bits that correspond to that bitplane) for MBs in WZS-MB mode is determined by the encoder based on the estimated conditional entropy  $H(u_{k,l} | s_{k,l})$  (or  $H(\text{sign}(u_k^l) | \text{sign}(s_k^l))$ ). For discrete random variables  $X$  and  $Y$ ,  $H(X | Y)$  can be written as

$$H(X | Y) = \sum_{y_i} \Pr(Y = y_i) H(X | Y = y_i), \quad (13)$$

where both  $\Pr(Y = y_i)$  and  $H(X | Y = y_i)$  can be easily calculated once the a priori probability of  $X$  and the transition probability matrix are known. The crossover probability, for example  $p_{01}$  in Figure 4(a), is derived by counting

TABLE 1: Channel parameters and the a priori probabilities for the 3rd bitplane of frame 3 of *Akiyo* CIF sequence when BL quantization parameter is 20 (with the same symbol notation as Figure 4).

$\Pr(u_{k,l} = 1)$	$p_{01}$	$p_{10}$	$\Pr(\text{sign}(u_k^l) = 1)$	$\alpha$	$\beta$
0.13	0.019	0.14	0.49	0.13	0.001

the number of coefficients such that  $u_{k,l} = 0$  and  $u_{k,l} \neq s_{k,l}$ . Table 1 shows an example of those parameters for both  $u_{k,l}$  and the sign bits. Note that the crossover probabilities between  $u_{k,l}$  and  $s_{k,l}$  are very different for source symbols 0 and 1, and therefore an asymmetric binary channel model will be needed to code  $u_{k,l}$  as shown in Figure 4(a). However, the sign bit has almost the same transitional probabilities whenever the input is  $-1$  or  $1$ , and is thus modelled as a symmetric discrete memoryless channel in Figure 4(b).

In terms of complexity, note that there are two major steps in this estimation method: (i) bitplane extraction from  $s_k$  and (ii) conditional entropy calculation (including the counting to estimate the crossover probabilities). Bitplanes need to be extracted only once per frame and this is done with a simple shifting operation on the original frame. Conditional entropy will be calculated for each bitplane based on the crossover probabilities estimated by simple counting. In Section 5, we will compare the complexity of the proposed WZS approach and the ET approach.

#### 4.4. Model-based estimation

In this section, we introduce a model-based method for correlation estimation that has lower computational complexity, at the expense of a small penalty in coding efficiency. The basic idea is to estimate first the probability density functions (pdf) of the DCT residuals ( $u_k, v_k, z_k = v_k - u_k$ ), and then use the estimated pdf to derive the crossover probabilities for each bitplane.

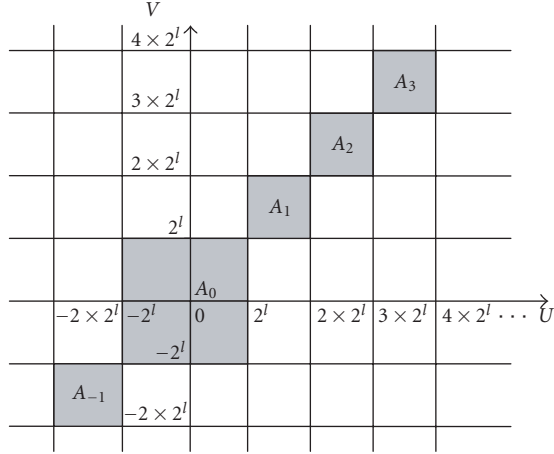


FIGURE 6: Crossover probability estimation. The shaded square regions  $A_i$  correspond to the event where crossover does not occur at bitplane  $l$ .

Assume that  $u_k, v_k, z_k$  are independent realizations of the random variables  $U, V$ , and  $Z$ , respectively. Furthermore, assume that  $V = U + Z$ , with  $U$  and  $Z$ , independent. We start by estimating the pdf's  $f_U(u)$  and  $f_Z(z)$ . This can be done by choosing appropriate models for the data samples, and estimating the model parameters using one of the standard parameter estimation techniques, for example, maximum-likelihood estimation, expectation maximization (EM), and so forth. Note that since the  $v_k$  are not available in our encoder, we use  $s_k$  to approximate  $v_k$  in the model parameter estimation.

Once we have estimated  $f_U(u)$  and  $f_Z(z)$ , we can derive the crossover probabilities at each bitplane as follows. Recall that we consider there is no crossover when  $u_k, v_k$  fall into the same quantization bin. This corresponds to the event denoted by the shaded square regions in Figure 6. Hence we can find the estimate of the crossover probability at bitplane  $l$  (denoted as  $\hat{p}(l)$ ) by

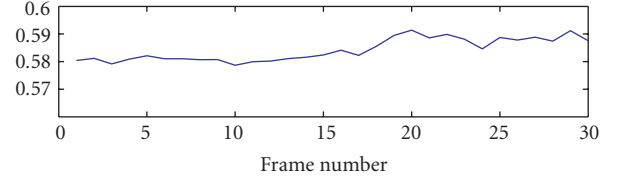
$$\hat{p}(l) = 1 - I(l), \quad (14)$$

where  $I(l)$  is given by

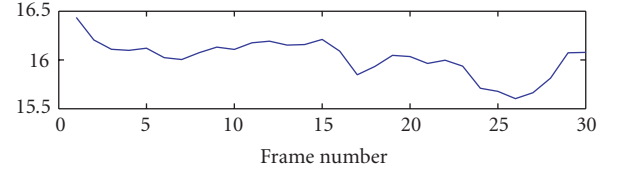
$$\begin{aligned} I(l) &= \sum_i \iint_{A_i} f_{UV}(u, v) du dv \\ &= \sum_i \iint_{A_i} f_U(u) f_{V|U}(v | u) du dv. \end{aligned} \quad (15)$$

$I(l)$  is simply the probability that  $U, V$  fall into the same quantization bin. The conditional pdf  $f_{V|U}(v | u)$  can be obtained as

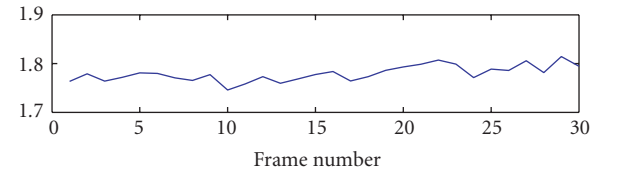
$$f_{V|U}(v | u) = f_Z(v - u), \quad (16)$$



(a) Mixing probability



(b) Standard deviation of the 1st Laplacian



(c) Standard deviation of the 2nd Laplacian

FIGURE 7: Model parameters of  $u_k$  estimated by EM using the video frames from *Akiyo*.

and the integral in (15) can be readily evaluated for a variety of densities. In practice, we only need to sum over a few regions,  $A_i$ , where the integrals are nonzero.

We found that  $U$  and  $Z$  can be well modeled by mixtures of two zero-mean Laplacians with different variances. We use the EM algorithm to obtain the maximum-likelihood estimation of the model parameters, and use (15) and (16) to compute the estimates of the crossover probabilities.

The main advantage of this model-based estimation approach as compared with the direct estimation is that it incurs less complexity and requires less frame data to be measured. In our experiment, the EM was operating on only 25% of the frame samples. Moreover, since the model parameters do not vary very much between consecutive frames (Figure 7), it is viable to use the previous estimates to initialize the current estimation and this can usually lead to convergence within a few iterations. Once we have found the model parameters, computing the crossover probability of each bitplane from the model parameters requires only negligible complexity since this can be done using closed-form expressions obtained from the integrals in (15). However, the approach suffers some loss in compression efficiency due to the inaccuracy in the estimation. We can assess the compression efficiency by evaluating the entropy function on the estimates of the crossover probabilities (which gives the theoretical limit in compressing the bitplanes given the estimates [19]), and compare to that of the direct estimation. Experiments using video frames from the *Akiyo* sequence show that with base layer quantization parameter (QP) set to 31



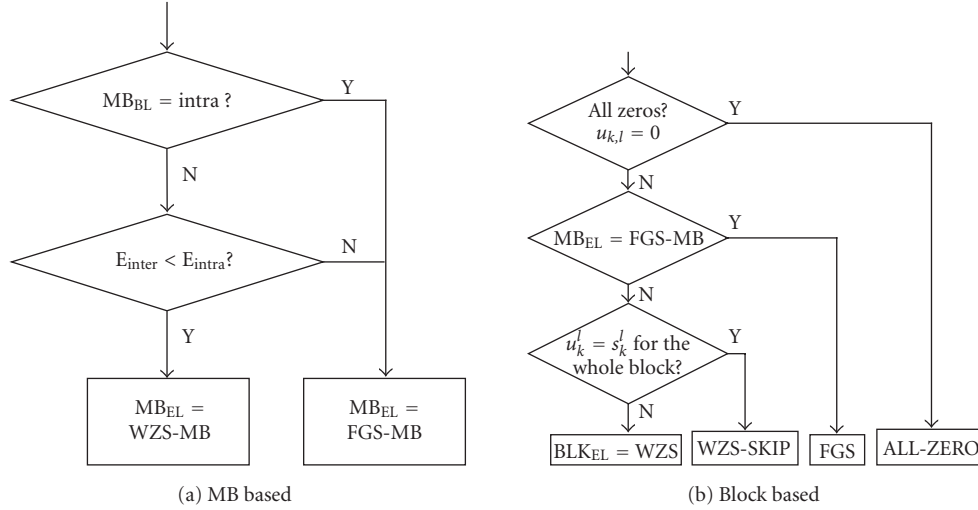


FIGURE 9: The block diagram of mode selection algorithm.

The encoder SI  $s_k$  is constructed in a similar way as (11), while taking into account the motion compensation and DCT transform as

$$s_k = T(MC_k[X_{k-1}] - \hat{X}_k^b). \quad (19)$$

Both  $u_k$  and  $s_k$  are converted into bitplanes.

Based on the switching rule given in Section 4.2, we define our mode selection algorithm as shown in Figure 9. At each bitplane, we first decide the coding mode on the MB-basis as in Figure 9(a), and then in each MB, we will decide the corresponding modes at the DCT block level to include the two special cases ALL-ZERO and WZS-SKIP (see Figure 9(b)). In either ALL-ZERO or WZS-SKIP modes, no additional information is sent to refine the block. The ALL-ZERO mode already exists in the current MPEG-4 FGS syntax. For a block coded in WZS-SKIP, the decoder just copies the corresponding block of the reference frame.<sup>1</sup> All blocks in FGS mode are coded directly using MPEG-4 FGS bitplane coding.

For blocks in WZS mode, we apply channel codes to exploit the temporal correlation between neighboring frames. Here, we choose low-density parity check (LDPC) codes [19, 20] for their low probability of undetectable decoding errors and near-capacity coding performance. A  $(n, k)$  LDPC code is defined by its parity-check matrix  $H$  with size  $n \times (n - k)$ . Given  $H$ , to encode an arbitrary binary input sequence  $c$  with length  $n$ , we multiply  $c$  with  $H$  and output the corresponding syndrome  $z$  with length  $(n - k)$  [19]. In a practical implementation, this involves only a few binary

additions due to the low-density property of LDPC codes. At bitplane  $l$ , we first code the binary number  $u_{k,l}$  for all coefficients in the WZS blocks, using LDPC codes to generate syndrome bits at a rate determined by the conditional entropy in (13). We leave a margin of about 0.1 bits above the Slepian-Wolf limit (i.e., the conditional entropy) to ensure that the decoding error is negligible. Then, for those coefficients that become significant in the current bitplane (i.e., coefficients that were 0 in all the more significant bitplanes and become 1 in the current bitplane), their sign bits are coded in a similar way using the sign bits of the corresponding  $s_k$  as SI.

The adaptivity of our scalable coder comes at the cost of an extra coding overhead. It includes: (1) the prediction modes for MBs and DCT blocks, (2) the a priori probability for  $u_{k,l}$  (based on our experiments, we assume a uniform distribution for sign bits) and channel parameters, and (3) encoding rate  $(1 - k/n)$ . A 1-bit syntax element is used to indicate the prediction mode for each MB at each bitplane. The MPEG-4 FGS defines the most significant bitplane level for each frame, which is found by first computing the residue with respect to the corresponding base layer for the frame and then determining what is the minimum number of bits needed to represent the largest DCT coefficient in the residue. Clearly, this most significant bitplane level varies from frame to frame. Note that representation of many DCT blocks in a given frame is likely to require fewer bitplanes than the maximum number of bitplanes for the frame. Thus, for these blocks, the first few most significant bitplanes to be coded are likely to be ALL-ZERO (for these blocks, the residual energy after interpolation using the base layer is low, so that most DCT coefficients will be relatively small). To take advantage of this, the MB prediction mode for a given bitplane is not sent if all its six DCT blocks are ALL-ZERO. Note also that the number of bits needed to represent the MB mode is negligible for the

<sup>1</sup> The WZS-SKIP mode may introduce some small errors due to the difference between the SI at the encoder and decoder.

least significant bitplanes, as compared to the number of bits needed to code the bitplanes. It is also worth pointing out that this mode selection overhead is required as well for a closed-loop coder that attempts to exploit temporal correlation through the mode-switching algorithm. For an MB in WZS-MB mode, the block mode (either WZS or WZS-SKIP) is signaled by an additional 1-bit syntax. This overhead depends on the number of MBs in WZS-MB mode, and a good entropy coding can be applied to reduce the overhead, since we have observed in our experiments that the two different modes have biased probabilities (see Figure 11). The encoding rate of syndrome codes varies from 1/64 to 63/64 in incremental steps of size 1/64, and thus 6 bits are used to code the selected encoding rate. We use a fixed-point 10 bit representation for the different kinds of probabilities to be sent to the decoder. An example of the total overhead percentage at each bitplane, which is calculated as the ratio between the number of overhead bits and the number of total bits to code this bitplane, is given in Table 2 for *News* sequence.

### 5.2. Decoding algorithm

Decoding of the EL bitplanes of  $X_k$  proceeds by using the EL reconstruction of the previous frame  $\hat{X}_{k-1}^e$  to form the SI for each bitplane. The syndrome bits received are used to decode the blocks in WZS mode. The procedure is the same as at the encoder, except that the original frame  $X_{k-1}$  is now replaced by the high quality reconstruction  $\hat{X}_{k-1}^e$  to generate SI:

$$v_k = T(MC_k[\hat{X}_{k-1}^e] - \hat{X}_k^b). \quad (20)$$

The corresponding SI at each bitplane is formed by converting  $v_k$  into bitplanes. The decoder performs sequential decoding since decoding a particular bitplane can only be done after more significant bitplanes have been decoded.

We modified the conventional LDPC software [20, 21] for the Slepian-Wolf approach by taking the syndrome information into account during the decoding process based on probability propagation. We follow a method similar to that described in [19, 22] to force the search of the most probable codeword in a specified coset determined by the syndrome bits. One main difference is that the a priori probability of the source bits  $u_{k,l}$  ( $p_0 = \Pr(u_{k,l} = 0)$  and  $p_1 = 1 - p_0$ ) is also considered in the decoding process. The likelihood ratio for each variable node at bitplane  $l$  is given by

$$\begin{aligned} \text{LLR} &= \log \frac{\Pr(u_{k,l} = 1 | v_{k,l})}{\Pr(u_{k,l} = 0 | v_{k,l})} \\ &= \begin{cases} \log \frac{p_{10}}{1 - p_{01}} + \log \frac{p_1}{p_0}, & \text{if } v_{k,l} = 0, \\ \log \frac{1 - p_{10}}{p_{01}} + \log \frac{p_1}{p_0}, & \text{if } v_{k,l} = 1, \end{cases} \quad (21) \end{aligned}$$

TABLE 2: Coding overhead for *News* sequence.

Bitplane	1	2	3	4
Overhead percentage (%)	19.8	9.6	7.5	4.6

where  $p_{ij}$  is the crossover probability defined in Figure 4(a). The syndrome information is considered in the same way as in [19] when calculating the likelihood ratio at the check node.

### 5.3. Complexity analysis

In our approach, the base-layer structure is the same as in an MPEG-4 FGS system. An additional set of frame memory, motion compensation (MC) and DCT modules, is introduced for the EL coding at both the encoder and decoder. The MC and DCT operations are only done once per frame even for multilayer coding. In comparison, the ET approach requires multiple motion-compensation prediction loops, each of which needs a separate set of frame memory, MC and DCT modules, as well as additional dequantization and IDCT modules to obtain each EL reconstruction. More importantly, for each EL, the ET approach needs to repeat all the operations such as reconstruction and prediction. Though our proposed approach requires correlation estimation at the encoder as discussed in Section 4, the additional complexity involved is very limited, including simple shifting, comparison, and  $+/-$  operations. Therefore, the proposed approach can be implemented in a lower complexity even for multiple layers.

It should be noted that the complexity associated with reconstructing the enhancement layers can be a significant portion of the overall encoding complexity in a closed-loop scalable encoder. While it is true that *full search* motion estimation (ME) (in base layer) may require a large amount of computational power, practical encoders will employ some form of fast ME, and the complexity of ME module can be substantially reduced. For example, [23] reports that ME (full-pel and sub-pel) takes only around 50% of the overall complexity in a practical nonscalable video encoder employing fast ME. As a result, the complexity of closing the loop (motion compensation, forward and inverse transforms, quantization, and inverse quantization) becomes a significant fraction of the overall codec complexity. Moreover, we need to perform these operations in every enhancement layer in a closed-loop scalable system (while usually we perform ME only in base layer). In addition to computational complexity reduction, our system does not need to allocate the frame buffers to store the reconstructions in each enhancement layer. This can lead to considerable savings in memory usage, which may be important for embedded applications.

## 6. EXPERIMENTAL RESULTS

Several experiments have been conducted to test the performance of the proposed WZS approach. We implemented a

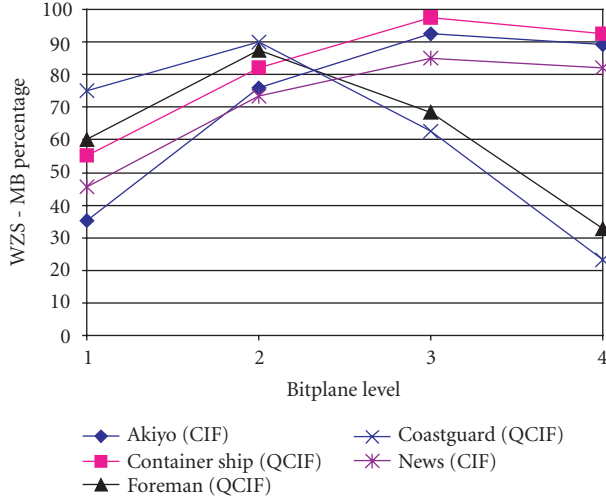


FIGURE 10: WZS-MB percentage for sequences in CIF and QCIF formats (BL quantization parameter= 20, frame rate= 30 Hz).

WZS video codec based on the MPEG-4 FGS reference software. In the experiments, we used the direct correlation estimation method, as it can lead to better compression efficiency as compared the model-based approach.

### 6.1. Prediction mode analysis

In this section, we analyze the block prediction modes at each bitplane for various video sequences. Figure 10 shows that the percentage of MBs in WZS-MB mode exceeds 50% for most video sequences (in some cases surpassing 90%, as in bitplane 3 for *Akiyo* and *Container Ship*). Therefore there is potentially a large coding gain over MPEG-4 FGS with our proposed approach. The percentage of MBs in WZS-MB is on average higher for low-motion sequences (such as *Akiyo*) than for high-motion sequences (such as *Coastguard*), especially for lower significance bitplanes. Moreover, this percentage varies from bitplane to bitplane. For the most significant bitplanes, the FGS-MB mode tends to be dominant for some sequences (such as *Akiyo* and *News*), due to the low quality of the EL reconstruction of the previous frame. When the reconstruction quality improves, as more bitplanes are decoded, the temporal correlation is higher and the WZS-MB mode becomes dominant, for example, for bitplanes 2 and 3 in Figure 10. However, the WZS-MB percentage starts to drop for even lower significance bitplanes. This is because the temporal correlation decreases for these bitplanes which tend to be increasingly “noise-like.”

The DCT block mode distribution in Figure 11 illustrates how the motion characteristics of the source sequence affect the relative frequency of occurrence of each block mode. The *Akiyo* sequence has a much larger WZS-SKIP percentage, and a larger percentage of WZ coded blocks, than *Coastguard*; thus *Akiyo* sees more significant reductions in coding rate when WZS is introduced. In contrast, for *Coastguard*, the percentage of blocks in WZS mode is less than

that in FGS mode starting at bitplane 4, thus showing that as motion in the video sequence increases, the potential benefits of exploiting temporal correlation in the manner proposed in this paper decreases. Note that neither Figure 10 nor Figure 11 include the least two significant bitplanes since the PSNR ranges for these bitplanes are not of practical interest.

## 6.2. Rate-distortion performance

### 6.2.1. Coding efficiency of WZS

In this section we evaluate the coding efficiency of the proposed WZS approach. Simulation results are given for a series of test sequences in CIF (352×288) and QCIF (176×144) resolutions with frame rate 30 Hz. *Akiyo* and *Container Ship* sequences have limited motion and low spatial detail, while the *Coastguard* and *Foreman* sequences have higher motion and more spatial detail. *News* sequence is similar to *Akiyo*, but with more background motion.

In addition to the MPEG-4 FGS and non-scalable (single layer) coding, we also compare our proposed approach with a multilayer closed-loop (MCLP) system that exploits EL temporal correlation through multiple motion-compensation loops at the encoder. The same MPEG-4 baseline video coder is used for all the experimental systems (note that the proposed WZS framework does not inherently require the use of a specific BL video coder). The first video frame is intra-coded and all the subsequent frames are coded as P-frame (i.e., IPPP...). The BL quantization parameter (QP) is set to 20. Prior to reporting the simulation results, we give a brief description of our proposed system together with the MCLP system.

#### Proposed WZS system

The DCT blocks are coded in four different modes as described in Section 6.1. An LDPC code is used to code those blocks in WZS mode at each bitplane to exploit the EL correlation between adjacent frames. The encoding rate is determined by the correlation estimated at the encoder without constructing multiple motion-compensation loops. To limit the error introduced by WZS-SKIP mode due to the small difference between the encoder and decoder SI, we disable WZS-SKIP mode once every 10 frames in our implementation.

#### Multiple closed-loop (MCLP) system

This system is an approximation to the ET approach discussed in Section 3.1 through the mode-switching algorithm. We describe the coding procedure for each enhancement layer as follows. To code an EL which corresponds to the same quality achieved by bitplane  $l$  in MPEG-4 FGS, the encoder goes through the following steps. (i) Generate the EL reconstruction of the previous frame up to this bitplane level, which we denote  $\hat{x}_{k-1}^l$ . (ii) Follow a switching rule similar to that proposed for the WZS system to determine the

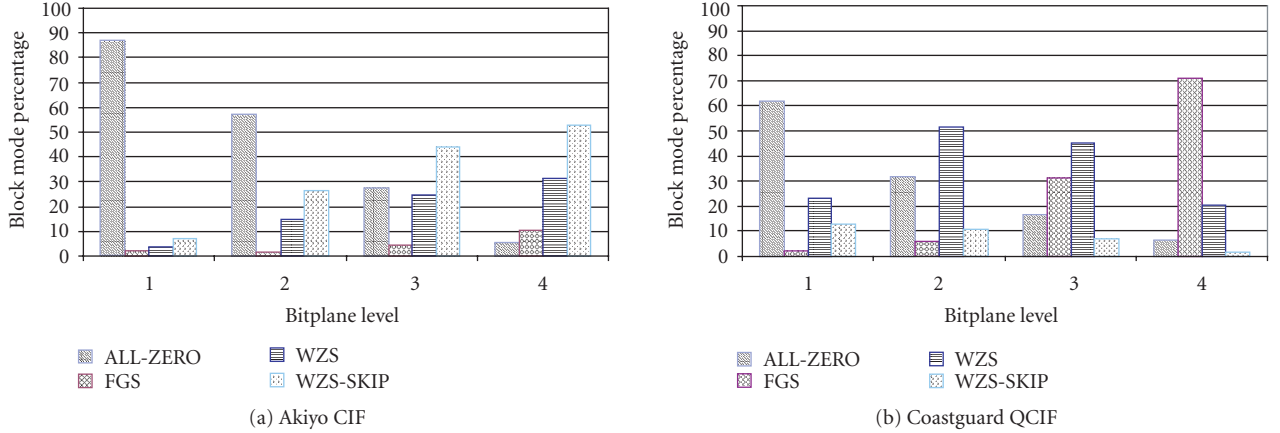


FIGURE 11: Percentages of different block modes for *Akiyo* and *Coastguard* sequences (BL quantization parameter = 20, frame rate = 30 Hz).

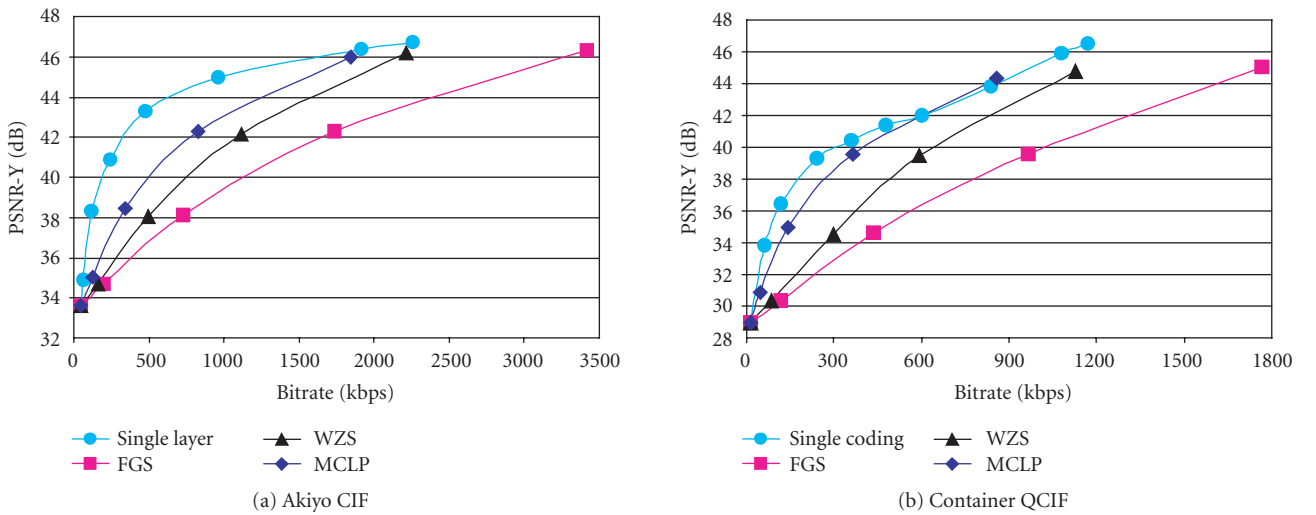


FIGURE 12: Comparison between WZS, nonscalable coding, MPEG-4 FGS, and MCLP for *Akiyo* and *Container Ship* sequences.

prediction mode of each MB, that is, inter-mode is chosen if  $E_{\text{inter}} < E_{\text{intra}}$ , and the FGS mode is chosen otherwise. Since the EL reconstruction is known at the encoder, it can calculate  $E_{\text{inter}}$  directly using the expression of (9). (iii) Calculate the EL residual  $r_k^e$  following (5) by using  $\hat{x}_{k-1}^l$  as the predictor for inter-mode, and the reconstruction of the current frame with more significant ELs  $\hat{x}_{k-1}^{l-1}$  as the predictor for FGS mode. (iv) Convert  $r_k^e$  to bitplanes, and code those bitplanes that are at least as significant as bitplane  $l$  (i.e., quantize to the  $l$ th bitplane) to generate the compressed bitstream.

Figures 12–14 provide a comparison between the proposed WZS, nonscalable coder, MPEG-4 FGS, and the MCLP coder. The PSNR gain obtained by the proposed WZS approach over MPEG-4 FGS depends greatly on the temporal correlation degree of the video sequence. For sequences with higher temporal correlation, such as *Akiyo* and *Container Ship*, the PSNR gain of WZS is greater than that for

lower temporal correlation sequences, such as *Foreman*, for example, 3–4.5 dB PSNR gain for the former, as compared to 0.5–1 dB gain for the latter.

To demonstrate the efficiency of Wyner-Ziv coding for WZS blocks, we compare the proposed coder to a simplified version that uses only the ALL-ZERO, FGS, and WZS-SKIP modes (which we call the “WZS-SKIP only” coder). The “WZS-SKIP only” coder codes the WZS blocks in FGS instead. Figure 15 shows that, for both *Akiyo* and *Coastguard* sequences, there is a significant improvement by adding the WZS mode. Note that the PSNR values for a given bitplane level are exactly the same for the two coders. The only difference is the number of bits used to code those blocks that are coded in WZS mode. Thus the coding gain of Wyner-Ziv coding (exploiting temporal correlation) over the original bitplane coding (that does not exploit temporal correlation) can be quantified as a percentage reduction in rate. We present this in two different ways as shown in

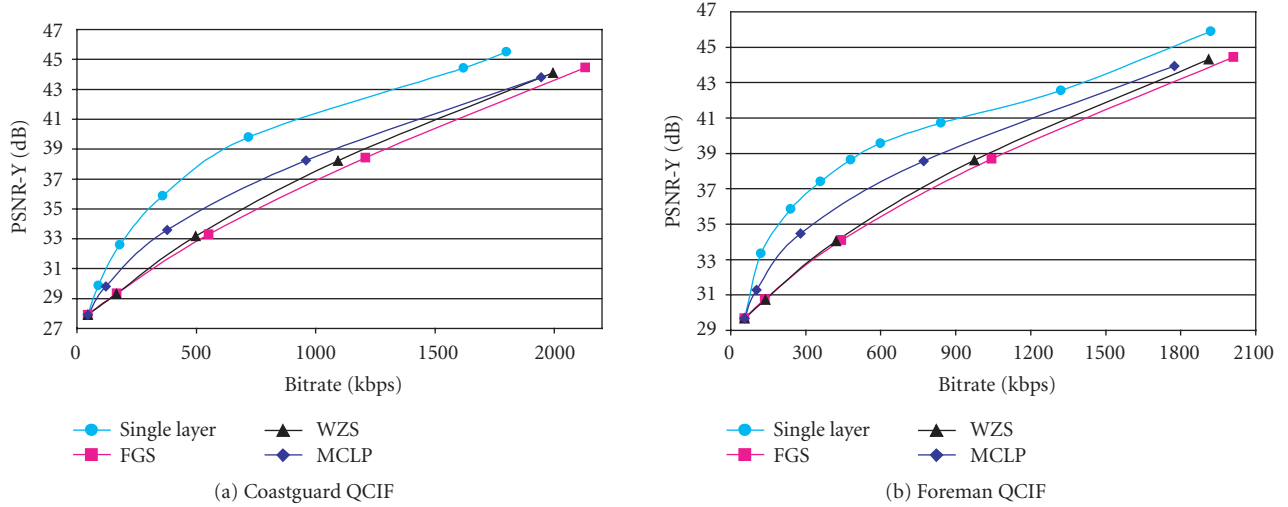


FIGURE 13: Comparison between WZS, nonscalable coding, MPEG-4 FGS, and MCLP for *Coastguard* and *Foreman* sequences.

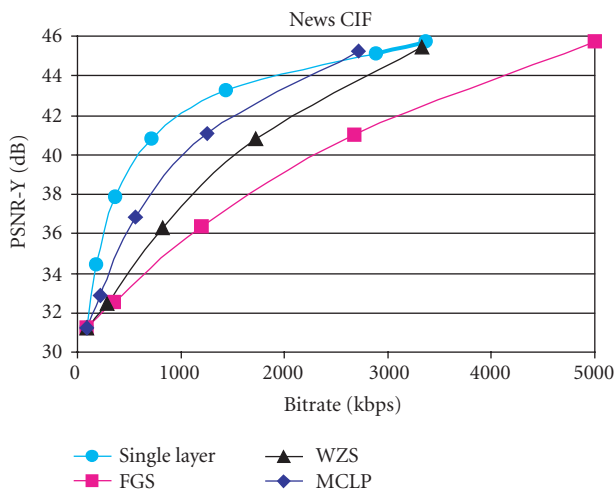


FIGURE 14: Comparison between WZS, nonscalable coding, MPEG-4 FGS, and MCLP for *News* sequence.

Tables 3 and 4.<sup>2</sup> Table 3 provides the rate savings for only those blocks in WZS mode. It can be seen that *Akiyo* achieves larger coding gain than *Coastguard* due to higher temporal correlation. Table 4 provides the overall rate savings (i.e., based on the total rate needed to code the sequence). These rate savings reflect not only the efficiency of coding each WZS block by Wyner-Ziv coding but also the percentage of blocks that are coded in WZS mode.

<sup>2</sup> It is usually required for an LDPC coder to have a large code length to achieve good performance. If the number of WZS blocks is not enough for the required code length, we force all blocks in the bitplane to be coded in FGS mode instead. This happens, for example, for the most significant bitplane of most sequences. Thus, only the results for bitplanes 2–4 are shown in these tables.

TABLE 3: Rate savings due to WZS for WZS blocks only (percentage reduction in rate with respect to using FGS instead for those blocks).

Bitplane level	2	3	4
Akiyo	24.66	31.20	26.71
Coastguard	19.98	22.91	19.19

TABLE 4: Overall rate savings due to WZS (percentage reduction in overall rate as compared to FGS).

Bitplane level	2	3	4
Akiyo	10.98	16.61	16.11
Coastguard	8.38	8.68	6.08

As seen from Figures 12–14, there is still a performance gap between WZS and MCLP. We compare the main features of these two approaches that affect the coding performance in Table 5. It should be clarified that occasionally, at very high rate for low-motion sequences, the MCLP approach can achieve similar (or even better) coding performance than the nonscalable coder. That is because bitplane coding is more efficient than nonscalable entropy coding when compressing the high-frequency DCT coefficients. We believe that the performance gap between WZS and MCLP is mainly due to the relative inefficiency of the current channel coding techniques as compared to bitplane coding. We expect that large rate savings with respect to our present WZS implementation can be achieved if better channel codes are used, that can perform closer to the Slepian-Wolf limit, or more advanced channel coding techniques are designed for more complex channels, which can take advantage of the existence of correlation among channel errors.

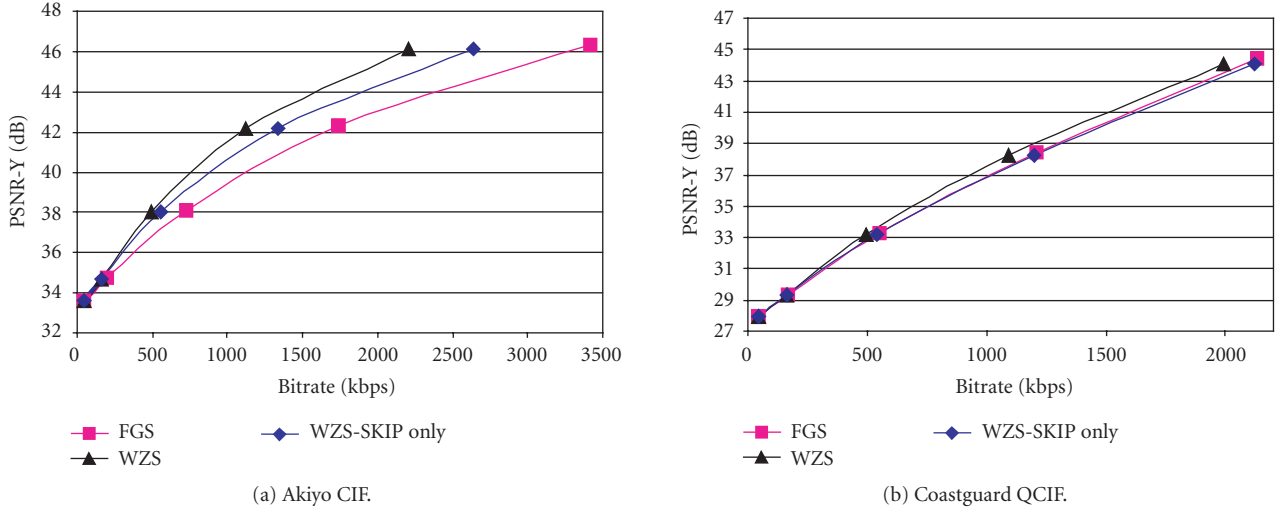
FIGURE 15: Comparison between WZS and “WZS-SKIP only” for *Akiyo* and *Coastguard* sequences.

TABLE 5: Comparisons between MCLP and WZS.

<i>Multiple closed-loop approach</i>	<i>Wyner-Ziv scalability approach</i>
(1) Exploits temporal correlation through closed-loop predictive coding.	(1) Exploits temporal correlation through Wyner-Ziv coding.
(2) Efficient bitplane coding (run, end-of-plane) of the EL residual in (5) to exploit the correlation between consecutive zeros in the same bitplane.	(2) Channel coding techniques designed for memoryless channels cannot exploit correlation between source symbols.
(3) The residual error between the source and EL reference from the previous frame may increase the dynamic range of the difference and thus cause fluctuations in the magnitude of residue coefficients (as the number of refinement iterations grows, the magnitude of residues in some coefficients can actually increase, even if the overall residual energy decreases).	(3) The source information to be coded is exactly the same as the EL bitplanes in MPEG-4 FGS, and therefore there are no fluctuations in magnitude and no additional sign bits are needed.
(4) Each EL has to code its own sign map, and therefore for some coefficients, the sign bits are coded more than once.	(4) An extra encoding rate margin is added to compensate for the small mismatch between encoder and decoder SI as well as for the practical channel coders which cannot achieve the Slepian-Wolf limit exactly.

### 6.2.2. Rate-distortion performance versus base layer quality

It is interesting to consider the effects of the base-layer quality on the EL performance of the WZS approach. We use *Akiyo*, *Container Ship*, and *News* sequences in the experiment. Table 6 shows the base-layer PSNR (for luminance component only) for several sequences under different base-layer quantization parameter (QP) values. The PSNR gains obtained by the proposed WZS approach over MPEG-4 FGS are plotted in Figure 16. The coding gain achieved by WZS decreases if a higher quality base-layer is used, as seen from Figure 16 when the base layer QP decreases to 8. That is because the temporal correlation between the successive frames

is already well exploited by a high-quality base layer. This observation is in agreement with the analysis in Section 3.1.

### 6.2.3. Comparisons with progressive fine granularity scalable (PFGS) coding

The PFGS scheme proposed by Wu et al. [2] improves the coding efficiency of FGS, by employing an additional motion-compensation loop to code the EL, for which several FGS bitplanes are included in the loop to exploit EL temporal correlation. Figure 17 compares the coding performance between WZS and PFGS for *Foreman* sequence. WZS performs worse than PFGS. In addition to the limitation of current techniques for Wyner-Ziv coding, the performance gap

TABLE 6: The base-layer PSNR (dB) for different QP.

Base-layer QP	31	20	8
Akiyo	32.03	33.61	38.05
Container Ship	26.97	28.93	34.11
News	29.17	31.23	36.09

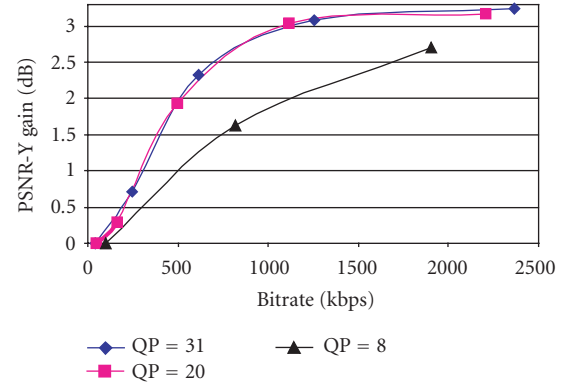
may come from the difference of the prediction link structure between these two approaches. WZS creates a multilayer Wyner-Ziv prediction link to connect the same bitplane level in successive frames. However, in PFGS, usually at least two or three FGS bitplanes are used in the EL prediction for all the bitplanes. Thus, this structure is beneficial to the most significant bitplanes (e.g., the 1st or 2nd bitplane) as they have higher-quality reference than what they would in WZS.

On the other hand, our proposed WZS techniques can be easily combined with a PFGS coder such that the more significant bitplanes can be encoded in a closed-loop manner by PFGS techniques, while the least significant bitplanes are predicted through Wyner-Ziv links to exploit the remaining temporal correlation. Figure 10 shows that for some sequences (especially those with low motion), the temporal correlation for some lower significance bitplanes (e.g., bitplane 4) is still high, so that WZS-MB mode is chosen for a considerable percentage of MBs. Thus, we expect that further gain would be achieved with our techniques over what is achievable with PFGS.

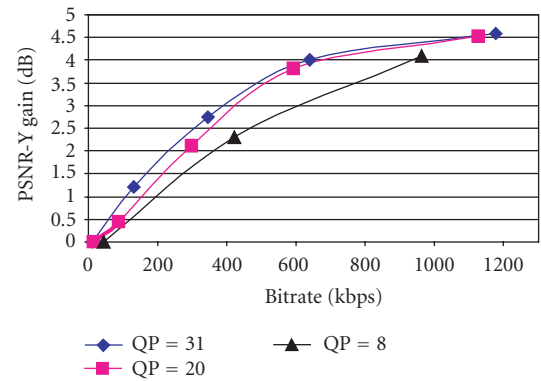
## 7. CONCLUSIONS AND FUTURE WORK

We have presented a new practical Wyner-Ziv scalable coding structure to achieve high coding efficiency. By using principles from distributed source coding, the proposed coder is able to exploit the enhancement-layer correlation between adjacent frames without explicitly constructing multiple motion-compensation loops, and thus reduce the encoder complexity. In addition, it has the advantage of backward compatibility with standard video codecs by using a standard CLP video coder as base layer. Two efficient methods are proposed for correlation estimation based on different tradeoff between the complexity and accuracy at the encoder even when the exact reconstruction value of the previous frame is unknown. Simulation results show much better performance over MPEG-4 FGS for sequences with high temporal correlation and limited improvement for high-motion sequences. Though we implemented the proposed Wyner-Ziv scalable framework in the MPEG-4 FGS software as bitplanes, it can be integrated with other SNR scalable coding techniques.

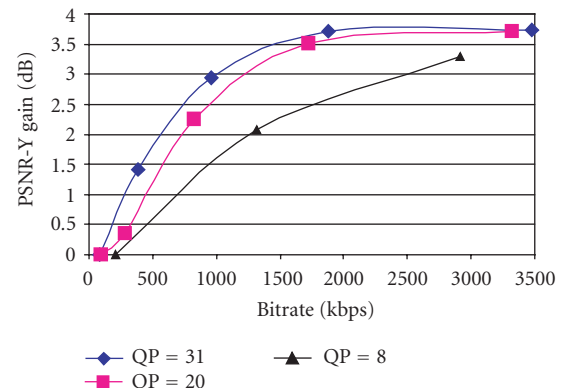
Further work is needed within the proposed framework to improve coding efficiency and provide flexible bandwidth adaptation and robustness. In particular, the selection of efficient channel coding techniques well suited for distributed source coding deserves additional investigation. Another possible reason for the gap between our proposed coder and a non-scalable coder is due to less accurate motion



(a) Akiyo CIF



(b) Container ship QCIF



(c) News CIF

FIGURE 16: The PSNR gain obtained by WZS over MPEG-4 FGS for different base-layer qualities.

compensation prediction in the enhancement layer when sharing motion vectors with the base layer. This can be improved by exploring the flexibility at the decoder, an important benefit of Wyner-Ziv coding, to refine the enhancement-layer motion vectors by taking into account the received enhancement-layer information from the previous frame. In terms of bandwidth adaptation, the current coder cannot fully achieve fine granularity scalability, given that the LDPC coder can only decode the whole block at the bitplane

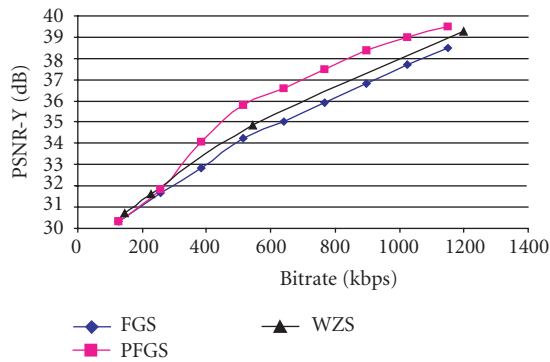


FIGURE 17: Compare WZS with MPEG-4 PFGS for *Foreman* CIF sequence (base layer QP = 19, frame rate = 10 Hz). The PFGS results are provided by Wu et al. from [24].

boundary. There is recent interest on punctured LDPC codes [25], and the possibility of using this code for bandwidth adaptation is under investigation. In addition, it is also interesting to evaluate the error resilience performance of the proposed coder. In principle, the Wyner-Ziv coding has more tolerance on noise introduced to the side information.

## ACKNOWLEDGMENTS

This research has been funded in part by the Integrated Media Systems Center, National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. The authors also wish to thank the anonymous reviewers for their comments which helped improve this paper significantly.

## REFERENCES

- [1] K. Rose and S. L. Regunathan, "Toward optimality in scalable predictive coding," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 965–976, 2001.
- [2] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 332–344, 2001.
- [3] M. van der Schaar and H. Radha, "Adaptive motion-compensation fine-granularity-scalability (AMC-FGS) for wireless video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 360–371, 2002.
- [4] H.-C. Huang, C.-N. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 372–385, 2002.
- [5] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proceedings of 40th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, Ill, USA, October 2002.
- [6] R. Puri and K. Ramchandran, "PRISM: a video coding architecture based on distributed compression principles," Tech. Rep. UCB/ERL M03/6, EECS Department, University of California, Berkeley, Calif, USA, March 2003.
- [7] A. M. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proceedings of 36th Asilomar Conference on Signals, Systems and Computers (ACSSC '02)*, vol. 1, pp. 240–244, Pacific Grove, Calif, USA, November 2002.
- [8] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005, Special issue on advances in video coding and delivery.
- [9] A. Sehgal, A. Jagmohan, and N. Ahuja, "A state-free causal video encoding paradigm," in *Proceedings of IEEE International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 605–608, Barcelona, Spain, September 2003.
- [10] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249–258, 2004.
- [11] 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, 1976.
- [12] Y. Steinberg and N. Merhav, "On successive refinement for the Wyner-Ziv problem," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1636–1654, 2004.
- [13] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Visual Communications and Image Processing (VCIP '04)*, vol. 5308 of *Proceedings of SPIE*, pp. 83–91, San Jose, Calif, USA, January 2004.
- [14] A. Sehgal, A. Jagmohan, and N. Ahuja, "Scalable video coding using Wyner-Ziv codes," in *Proceedings of Picture Coding Symposium (PCS '04)*, San Francisco, Calif, USA, December 2004.
- [15] M. Tagliasacchi, A. Majumdar, and K. Ramchandran, "A distributed-source-coding based robust spatio-temporal scalable video codec," in *Proceedings of Picture Coding Symposium (PCS '04)*, San Francisco, Calif, USA, December 2004.
- [16] H. Wang and A. Ortega, "Scalable predictive coding by nested quantization with layered side information," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 3, pp. 1755–1758, Singapore, October 2004.
- [17] H. Wang, N.-M. Cheung, and A. Ortega, "WZS: Wyner-Ziv scalable predictive video coding," in *Proceedings of Picture Coding Symposium (PCS '04)*, San Francisco, Calif, USA, December 2004.
- [18] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, 1991.
- [19] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.
- [20] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, 1996, reprinted with printing errors corrected in vol. 33, no. 6, pp. 457–458, 1997.
- [21] R. M. Neal, "Software for Low Density Parity Check (LDPC) codes," <http://www.cs.toronto.edu/~radford/ldpc.software.html>.
- [22] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.
- [23] H.-Y. Cheong, A. M. Tourapis, and P. Topiwala, "Fast motion estimation within the JVT codec," Tech. Rep. JVT-E023, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG 5th

Meeting, Geneva, Switzerland, October 2002, [http://ftp3.itu.ch/av-arch/jvt-site/2002\\_10\\_Geneva/JVT-E023.doc](http://ftp3.itu.ch/av-arch/jvt-site/2002_10_Geneva/JVT-E023.doc).

- [24] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, "H.26L-based fine granularity scalable video coding," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 4, pp. 548–551, Phoenix-Scottsdale, Ariz, USA, May 2002.
- [25] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.

**Huisheng Wang** received the B.Eng. degree from Xi'an Jiaotong University, China, in 1995 and the M. Engineering degree from Nanyang Technological University, Singapore, in 1998, both in electrical engineering. She is currently pursuing her Ph.D. degree in the Department of Electrical Engineering-Systems at the University of Southern California, Los Angeles. From 1997 to 2000, she worked in Creative Technology Ltd., Singapore, as an R & D Software Engineer. She was also a Research Intern at La Jolla lab, ST Microelectronics, San Diego, Calif, and at HP Labs, Palo Alto, Calif. Her research interests include signal processing, multimedia compression, networking, and communications.



**Ngai-Man Cheung** is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, University of Southern California, Los Angeles. He was a Research Engineer with Texas Instruments R & D Center, Tsukuba, Japan. He was also a Research Intern with IBM T. J. Watson Research Center, Yorktown, NY, and Nokia Research Center, Irving, Tex. His research interests include multimedia compression, signal processing, and communications.



**Antonio Ortega** received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid, Madrid, Spain, in 1989, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, in 1994, where he was supported by a Fulbright scholarship. In 1994, he joined the Electrical Engineering-Systems Department at the University of Southern California, where he is currently a Professor and an Associate Chair of the Department. He is a Senior Member of IEEE, and a Member of ACM. He has been Chair of the Image and Multidimensional Signal Processing (IMDSP) technical committee and a Member of the Board of Governors of the IEEE SPS (2002). He was the technical program Cochair of ICME 2002, and has served as the Associate Editor for the IEEE Transactions on Image Processing and the IEEE Signal Processing Letters. He received the NSF CAREER Award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, and the IEEE Signal Processing Society 1999 Magazine Award. His research interests are in the areas of multimedia compression and communications. His recent work is focusing on distributed compression, multiview coding, compression for recognition and classification applications, and wireless sensor networks.

