Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 22 (2007) 389–402
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Drift controlled scalable wavelet based video coding in the overcomplete discrete wavelet transform domain

Vidhya Seran, Lisimachos P. Kondi*

332 Bonner Hall, Department of Electrical Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA

Received 26 August 2004; received in revised form 23 January 2007; accepted 30 January 2007

Abstract

Traditional video coders use the previous frame to perform motion estimation and compensation. Though they are less complex and have minimum coding delays, these coders lose their efficiency when subjected to scalability requirements. Recent 3D wavelet coders using lifting schemes offer high compression efficiency and scalability without significant loss in performance. The main drawback of 3D coders is that they process several frames at a time. This introduces additional delay, which makes them less suitable for real time applications.

In this work, we propose a novel scheme to minimize drift in scalable wavelet based video coding, which gives a balanced performance between compression efficiency and reconstructed quality with less drift. Our drift control mechanism maintains two frame buffers in the encoder and decoder; one that is based on the base layer and one that is based on the base plus enhancement layers. Drift control is achieved by switching between these two buffers for motion estimation and compensation. Our prediction is initially based on the base plus enhancement layers buffer, which inherently introduces drift in the system if a part of the enhancement layer is not available at the receiver. A measure of drift is computed based on the channel information and a threshold is set. When the measure exceeds the threshold, i.e., when drift becomes significant, we switch the prediction to be based on the base layer buffer, which is always available to the receiver. We also developed an adaptive scheme with additional computation overhead at the encoder to decide the switching instance. The performance of the threshold case that needs fewer computations is comparable with the adaptive scheme. Our coder offers high compression efficiency and sustained video quality for variable bit rate wireless channels. This proves that we need not completely eliminate drift and decrease compression efficiency to get better received video quality.

Keywords: Wavelet-based video coding; Scalable video coding; Wireless video transmission; Drift control

1. Introduction

The popularity of multimedia applications demands support for different receivers that operate at different bit rates, resolution and complexity. This mandates the need for a scalable video coder with high compression efficiency. Wavelet based image coding has the very best coding efficiency and provides SNR scalability, besides resolution scalability. This has kindled the minds of many researchers to explore the possibilities of using wavelets in video coding. Initial research in this area had to face challenges like aliasing caused by

*Corresponding author. Tel.: +1 716 6452422x1147; fax: +1 716 6453656.
E-mail addresses: vseran@eng.buffalo.edu (V. Seran), lkondi@eng.buffalo.edu (L.P. Kondi).

0923-5965/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2007.01.003

decimation in the wavelet decomposition. This led to the use of the overcomplete wavelet decomposition to overcome the aliasing problem. Several works have been recently proposed for motion estimation and compensation in the overcomplete wavelet domain [1,6,13,15].

Layered coding together with error protection techniques offers high error resilience to channel induced errors [3,11,12]. In traditional coders, motion estimation/motion compensation (ME/MC) is only based on the base layer and it ignores all the enhancement layer information. This is done because the enhancement layers are not always available at the receiver. By neglecting the enhancement layers in the prediction, traditional coders will lose in compression efficiency and there will be a loss of 0.5–1.5 dB for every layer [17,18]. Including the enhancement layer in the ME/MC loop introduces drift, defined as the propagation of errors due to partial reception of enhancement information. Base layer prediction using enhancement layers is of particular importance because it offers very good compression efficiency though it suffers from drift in a lossy network. Recent works have shown that drift need not be completely eliminated, but it can be controlled in discrete cosine transform (DCT) based video coders [4,17,18].

Another way to totally eliminate drift is to use 3D subband coding methods. In 3D subband coding, a group of frames is processed at a time to compress the video sequence [10,14,30]. The encoder and decoder must buffer the required number of frames before they can apply wavelet transform. MC in the temporal domain (either 2D + t or t + 2D) and with lifting techniques offers high compression and scalability [2,5,7,16,23,24,29]. However, motion compensated temporal filtering requires the availability of future frames for ME/MC. This introduces a delay, which makes such codecs less suitable for real time video applications like teleconferencing.

Our work is an attempt to control drift in wavelet based video coders where enhancement layers are used to predict the base layer information. The proposed coder eliminates the need to transmit an intra frame at regular intervals to completely eliminate drift. The focus of this work is to manage drift in a wireless transmission system with unequal error protection (UEP)/variable bit rate channels that have a known loss rate. Our coder is optimized for known channel conditions. We also extended the proposed scheme for heterogeneous network conditions. Some preliminary results have appeared in [25,26].

The rest of the paper is organized as follows: in Section 2, we discuss the ME/MC in overcomplete wavelet domain. In Section 3, we explain our coder architecture and drift control mechanism. In Section 4, we deal with the channel modeling for a lossy packet network and a heterogeneous network. In Section 5, we present the simulation results for different channel conditions. Finally in Section 6, we present our conclusions.

2. Wavelet based video coding

Wavelet transform of a signal provides a multiresolution/multifrequency representation in both the spatial and frequency domains. Wavelet based image coding has achieved tremendous success both in coding efficiency and in scalability. Many researchers are trying to exploit the advantages of wavelet transforms in video coding. When discrete wavelet transform (DWT) is applied, the original signal is transformed into low and high resolution components or subbands by the successive application of filters. The main difficulty in wavelet based video coding is to couple the DWT with ME/MC. One approach is to replace the DCT with DWT, i.e., the ME/MC takes place in the spatial domain and the DWT is applied to the residual image. This approach suffers from blocking artifacts, as the DWT is applied to the whole residue image and not block-by-block. Another approach is to have ME/MC in the wavelet domain. The decimation and expansion operation in DWT are shift variant, which hinders ME/MC and produces large error coefficients. Such high error coefficients decrease the coding efficiency.

To overcome the shift variant property of DWT, the overcomplete DWT (ODWT) is used [15]. The ODWT eliminates the downsampling operation. In other words, the critically sampled DWT does not consider all phase information due to decimation. As an example, in 1D one level wavelet transform, there are two phases, odd and even. The DWT uses either the odd phase or the even phase. In ODWT, both the odd phase and even phase coefficients are retained. ODWT is also called undecimated DWT, redundant DWT or the algorithme a trous.

In this work, the ME/MC is done using the low band shift method [15]. An input frame is decomposed in the critically sampled DWT domain and the reference frame is transformed using ODWT.

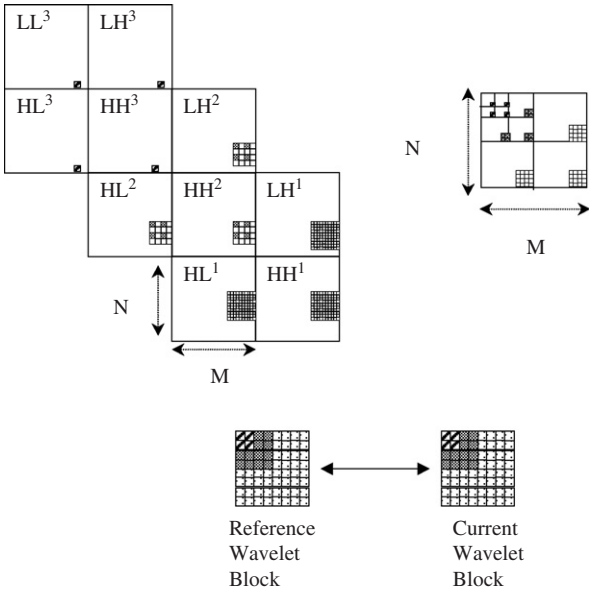

Fig. 1. Overcomplete wavelet transform (ODWT) vs discrete wavelet transform (DWT) is shown for a frame of size $N \times M$. The wavelet block formation is explained using the patched lines.

The wavelet coefficients are rearranged to form wavelet blocks such that the related coefficients in all scales and orientations are included in each wavelet block. This can be pictorially represented as in Fig. 1. The video frames of size $N \times M$ are subjected to a three-level decomposition. So, for each coefficient at level L in DWT domain, there are 4^L coefficients in ODWT. Thus, there are 4^L phases in level L.

Motion estimation is done using the block matching technique. Thus, the wavelet block of the reference frame is matched with the wavelet blocks of the current frame in a search window W, and the reference wavelet block is selected by minimizing the mean absolute difference (MAD). The MAD for the kth wavelet block is calculated as follows:

$$\begin{aligned}
&\mathrm{MAD}_k(\mathrm{d}x,\mathrm{d}y)\\
&=\sum_{i=1}^{3}\sum_{x_i=x_{i,k}}^{x_{i,k}+M/2^i-1}\sum_{y_i=y_{i,k}}^{y_{i,k}+N/2^i-1}\Bigg\{\mathrm{HL}_{\mathrm{cur}}^{(i)}(x_i,y_i)\\
&\quad-\mathrm{HL}_{\mathrm{ref}}^{(i)}\left(\mathrm{d}x\%2^i,x^i+\left\lfloor\frac{\mathrm{d}x}{2^i}\right\rfloor,y_i+\left\lfloor\frac{\mathrm{d}y}{2^i}\right\rfloor\right)\\
&\quad+\mathrm{LH}_{\mathrm{cur}}^{(i)}(x_i,y_i)-\mathrm{LH}_{\mathrm{ref}}^{(i)}\\
&\quad\times\left(\mathrm{d}x\%2^i,x^i+\left\lfloor\frac{\mathrm{d}x}{2^i}\right\rfloor,y_i+\left\lfloor\frac{\mathrm{d}y}{2^i}\right\rfloor\right)\\
&\quad+\mathrm{HH}_{\mathrm{cur}}^{(i)}(x_i,y_i)-\mathrm{HH}_{\mathrm{ref}}^{(i)}\\
&\quad\times\left(\mathrm{d}x\%2^i,x^i+\left\lfloor\frac{\mathrm{d}x}{2^i}\right\rfloor,y_i+\left\lfloor\frac{\mathrm{d}y}{2^i}\right\rfloor\right)\Bigg\}\\
&\quad+\sum_{x_3=x_{3,k}}^{x_{3,k}+M/2^3-1}\sum_{y_3=y_{3,k}}^{y_{3,k}+N/2^3-1}\Bigg\{\mathrm{LL}_{\mathrm{cur}}^{(3)}(x_3,y_3)\\
&\quad-\mathrm{LL}_{\mathrm{ref}}^{(3)}\left(\mathrm{d}x\%2^3,x_3+\left\lfloor\frac{\mathrm{d}x}{2^3}\right\rfloor,y_3+\left\lfloor\frac{\mathrm{d}y}{2^3}\right\rfloor\right)\Bigg\},\quad(1)
\end{aligned}$$

where $(\mathrm{d}x,\mathrm{d}y)$ is the displacement vector, $x\%y$ is the modulo operation and operator $\lfloor x\rfloor$ denotes the largest integer value less that or equal to x. $\mathrm{HL}^{(i)}$, $\mathrm{LH}^{(i)}$, $\mathrm{HH}^{(i)}$ represent the high–low, low–high, high–high subband of the current and reference frames at level i, respectively. The $x_{i,k}$ and $y_{i,k}$ are the initial point in the ith level subband of the kth wavelet block. The motion vector is the displacement vector that provides the minimum MAD.

$$(\mathrm{d}x,\mathrm{d}y)_{\min}=\min_{(\mathrm{d}x,\mathrm{d}y)\in W}\mathrm{MAD}_k(\mathrm{d}x,\mathrm{d}y).\quad(2)$$

This $(\mathrm{d}x,\mathrm{d}y)_{\min}$ motion vector retains all motion and phase information.

The residual image is in the downsampled wavelet domain and can be coded using any still image coders like the embedded zerotree wavelet (EZW) [27] or the set partitioning in hierarchical trees (SPIHT) coder [22]. We use SPIHT for encoding and decoding the residues as SPIHT gives a better compression efficiency compared to EZW.

The motion compensated current frame will be in critically sampled DWT, which serves as the reference for the next frame. The inverse DWT is performed first on the reference frame to get the overcomplete wavelet coefficients. In the resolution scalable scenario, the coarsest resolution is processed independently and every higher resolution is incrementally processed. In such case, only an approximated ODWT of the reference frame can be calculated, as the higher resolution subbands might not be available. This approximation will introduce additional drift. To avoid such drift in resolution scalability, the method using prediction filters [1] can be used, where the ODWT is calculated from the critically sampled DWT for every level.

3. Drift controlled coder

In a motion compensated multilayer encoder, prediction can be based on either the base layer or the base and one or more enhancement layers. The inclusion of enhancement layers in ME/MC

improves the reconstructed frame quality significantly by decreasing the residual errors. During transmission over a lossy channel, the enhancement layer may or may not be received completely at the receiver. Without loss of generality, we can assume that the base layer is always available at the decoder. At the decoder, the base layer and the partially received enhancement layer are used to reconstruct the next frame, $F_{k_{\text{dec}}}$. But the $F_{k_{\text{dec}}}$ will not be the same as the reconstructed frame at the encoder $F_{k_{\text{enc}}}$, if the enhancement layer suffers some loss. While $F_{k_{\text{enc}}}$ was used in the encoder to predict the next frame $F_{(k+1)_{\text{enc}}}$, the same frame $F_{k_{\text{enc}}}$ is not available at the decoder. This mismatch introduces drift in the decoder output and the decoder loses synchronization with the encoded sequence. Even if the enhancement layer of a subsequent frame is fully available at the decoder, a drift that has been previously introduced will not be eliminated. Forcing the decoder to synchronize with the encoder can eliminate drift.

The proposed encoder and decoder are shown in Figs. 2 and 3, respectively. The first frame is treated as an intra frame and transformed in the discrete wavelet domain. The DWT coefficients are coded using the SPIHT coder to produce a bitstream. The bitstream is partitioned into blocks of data such that there is one block per layer per frame. A block of the embedded bitstream that contains the most significant information forms the base layer. The remaining blocks are used to form one or more enhancement layers. The SPIHT decoded intra frame will be the reference frame for ME/MC of the current (second) frame. The reference frame is transformed in the overcomplete wavelet domain and the current frame is transformed using DWT. Motion vectors are estimated using the low band shift method as explained in Section 2. The prediction error is encoded using the SPIHT coder to generate a bitstream.

The encoder maintains two frame buffers: base buffer (Enc_BB) for base layer only prediction and

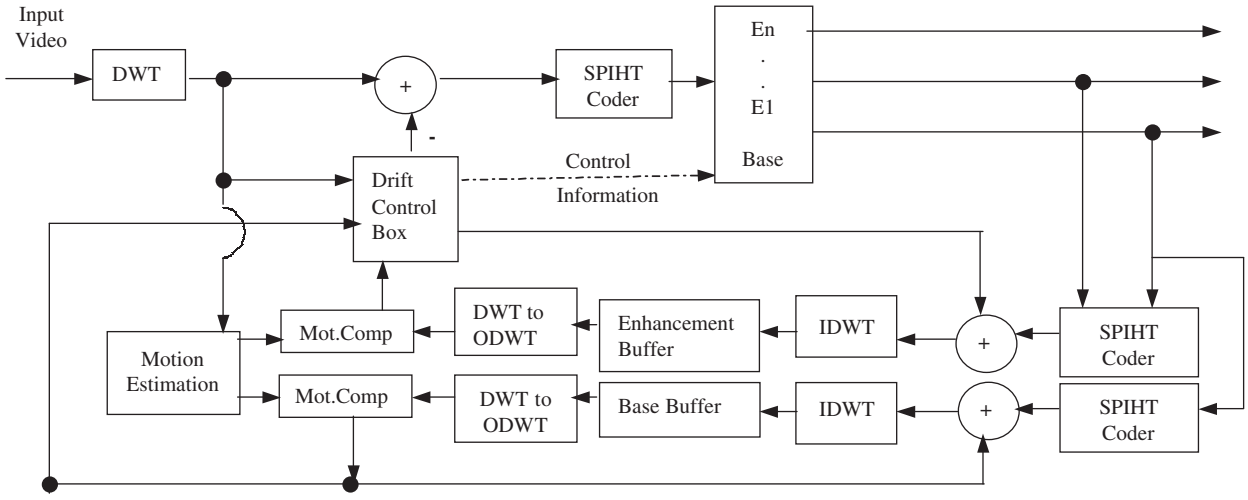

Fig. 2. Proposed encoder block diagram.

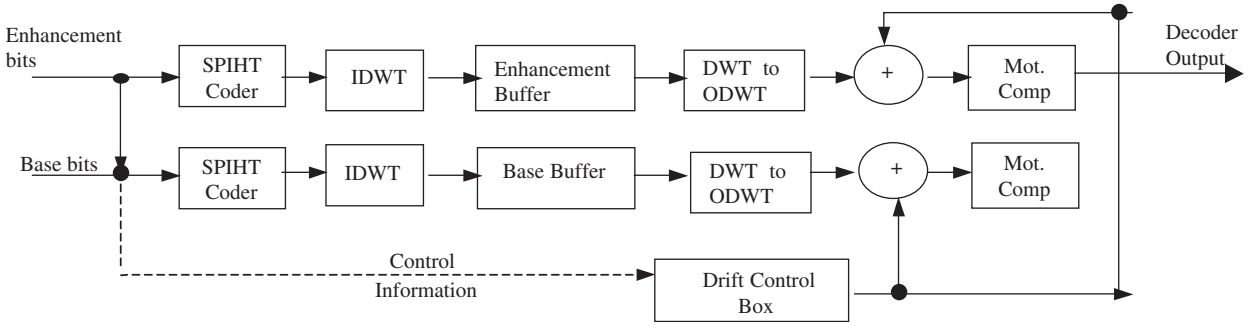

Fig. 3. Proposed decoder block diagram.

enhancement buffer (Enc_EB) for prediction based on both base and enhancement layers. The decoder also maintains two buffers: decoder base buffer (Dec_BB) and decoder enhancement buffer (Dec_EB). The Enc_BB and Dec_BB will maintain a predicted frame using only the base layer, though the ME/MC is done with the Enc_EB information. Since the base layer is always received in full, Enc_BB and Dec_BB have identical data and hence can be used to control drift in the Dec_EB. Given the approximate channel conditions and the rate, the encoder computes a measure of drift (MD) for the Enc_EB output. The MD is compared with a threshold, which we call as enhancement threshold (ET) and if MD exceeds the preset ET, significant drift will be induced at the decoder in the case of partial reception of the enhancement layer. Switching the prediction to be based on the Enc_BB for the next frame eliminates the drift introduced in the system. For subsequent frames prediction is again based on Enc_EB. An adaptive scheme to determine the switching instance is also proposed and explained in the following section.

It is to be noted that the use of Enc_EB motion estimation information in Enc_BB for motion compensation will progressively decrease the quality of the Enc_BB. Though of lesser quality, the base buffers in the encoder and decoder are identical. When switching to the base buffer is done, a poor quality Enc_BB would yield poor prediction. Hence, it is a precautionary measure to arrest any considerable drop in the quality of Enc_BB. Initiating a switching action when the difference between the quality of Enc_BB and Enc_EB (DB) rises above a base threshold (BT) will maintain the Enc_EB quality.

The switching decisions are made in the drift control box based on the adaptive method or the threshold method and the MD. The switching instances are conveyed to the decoder as control information. The drift control box in the decoder examines the received control information and does the switching between the buffers at exactly the same instance as in the encoder.

3.1. Drift estimation and drift control

The proposed encoder is employed for two channel conditions: a packet loss network and a heterogeneous network. The coder designed achieves maximum efficiency for a given transport mechanism. In a typical wireless video transmission scenario, the video packets are subjected to losses that are random in nature, but the probability of successful packet reception can be obtained. If the encoder is optimized to perform for a specific packet success rate, the decoder can reconstruct the video with acceptable quality and controlled drift. In the case of heterogeneous networks, different users operate at different bitrates and the encoded video stream should be able to efficiently operate over the expected range of bitrates. In [19,25,31], different optimization criteria are used to minimize distortion for different channel conditions. We consider the channel model explained in Section 4. The embedded coder generates a bitstream with $\mathrm{Renc_f}$ bits per frame. The encoder assumes that all bits are received by the decoder for given $\mathrm{Renc_f}$ and computes three peak to signal noise ratio (PSNR) values to calculate the MD and difference between the quality of Enc_EB and Enc_BB buffers (DB) at frame instance k. The three values are PSNR of Enc_EB at rate $\mathrm{Renc_f}$ (P_{EB}), PSNR of Enc_BB at base rate R_{b} (P_{BB}) and the PSNR at the average rate of $\mathrm{Renc_f}$ and R_{b} (P_{Av}). Then MD_k and DB_k are calculated as follows:

$$\mathrm{MD}_k = P_{k_{\mathrm{EB}}} - P_{k_{\mathrm{AV}}}, \tag{3}$$

$$\mathrm{DB}_k = P_{k_{\mathrm{EB}}} - P_{k_{\mathrm{BB}}}. \tag{4}$$

The selection of the buffer to be used for predicting the next frame $(k+1)$ can be done in two ways: using the threshold method and using the adaptive method.

3.1.1. Threshold method

The switching between Enc_EB and Enc_BB occurs when the following conditions are met:

$$\mathrm{MD}_k > \mathrm{ET} \text{ or } \mathrm{DB}_k > \mathrm{BT}. \tag{5}$$

If this switching is done very often or less frequently, it affects the compression efficiency. Therefore, the selection of a threshold value is very important. The thresholds are selected for a set of training sequences and channel conditions and can be applied to other sequences that do not belong to the training sets for the same channel conditions.

The quality of the decoder output without any loss and with drift for a given loss rate can be estimated. The difference between these values will provide an approximate range for the ET settings. Similarly, the difference in quality between the base and enhancement buffers will yield a range of BT

values. An optimum value is selected for ET and BT from a range for a packet loss networks and is explained in Section 5.1.3.

3.1.2. Adaptive method

There are two buffer options for ME/MC for frame $(k+1)$. In the threshold method, the buffer to be used for the $(k+1)$th frame is selected based on the thresholds and the buffer qualities at frame instance k. Instead, we can get two predictions separately for $(k+1)$ using the two buffers (one using Enc_EB and the other using Enc_BB). Thus, two sets of estimated PSNR values can be calculated as in the threshold case: one set represents the prediction using the Enc_BB buffer and the other set from the Enc_EB buffer. The estimated values are used to decide which buffer should be used to predict the $(k+1)$th frame in order to minimize the drift and maximize the output PSNR.

The superscript in the notations denotes the buffer used for prediction. For prediction based on Enc_EB, we get $P^{\mathrm{EB}}_{(k+1)_{\mathrm{EB}}}$, $P^{\mathrm{EB}}_{(k+1)_{\mathrm{BB}}}$ and $P^{\mathrm{EB}}_{(k+1)_{\mathrm{AV}}}$. For prediction based on Enc_BB, we calculate $P^{\mathrm{BB}}_{(k+1)_{\mathrm{EB}}}$, $P^{\mathrm{BB}}_{(k+1)_{\mathrm{BB}}}$ and $P^{\mathrm{BB}}_{(k+1)_{\mathrm{AV}}}$ values. The measure of drift for both predictions is calculated as

$$\mathrm{MD}^{\mathrm{EB}}_{(k+1)} = P^{\mathrm{EB}}_{(k+1)_{\mathrm{EB}}} - P^{\mathrm{EB}}_{(k+1)_{\mathrm{AV}}},$$
$$\mathrm{MD}^{\mathrm{BB}}_{(k+1)} = P^{\mathrm{BB}}_{(k+1)_{\mathrm{EB}}} - P^{\mathrm{BB}}_{(k+1)_{\mathrm{AV}}}. \quad (6)$$

The buffer selection is based on two requirements: one minimizing the drift and the other maximizing the output PSNR when there is drift. The adaptive algorithm for selection is given below:

(1) Find minimum drift case:

$$\min\{\mathrm{MD}^{\mathrm{EB}}_{(k+1)}, \mathrm{MD}^{\mathrm{BB}}_{(k+1)}\}. \quad (7)$$

(2) Find minimum distortion case:

$$\max\{P^{\mathrm{EB}}_{(k+1)_{\mathrm{AV}}}, P^{\mathrm{BB}}_{(k+1)_{\mathrm{AV}}}\}. \quad (8)$$

(3) Use the buffer that meets both Steps 1 and 2.
(4) Else use the enhancement buffer (Enc_EB).

When the buffer satisfies both the minimum distortion and minimum drift conditions, it is used for predicting the next frame. When there is a mismatch between the two conditions, then the Enc_EB buffer is used. In the adaptive method, we do not have to explicitly check for the based buffer quality as a separate ME/MC is also performed for the base layer. The proposed adaptive method does not require any threshold settings, but this method uses an additional ME/MC for each frame compared to the threshold case. The proposed adaptive algorithm uses local sequence information combined with the channel characteristics to decide on the switching instance.

4. Channel modelling

The video transmission system consists of an encoder, a channel and a decoder. Two different transport channel mechanisms are considered in this section: a typical wireless packet loss channel and a heterogeneous network case. Typically, video transmission channels are designed with a leaky bucket model [20]. There are different approaches to design a leaky bucket model [8,9,21,28]. The variable channel conditions are overcome at the receiver by choosing any of those models, such that the encoded bitstream is received without any loss. In scalable video, the decoding can be performed even with a partially received bitstream. In this work, our focus is on the determination of the number of bits received per frame during a frame interval. Transmission over a wireless channel is subject to loss and hence the received bits will be less than the encoded bits. Decoding of the frame starts immediately at a fixed time interval (frame interval) with the available bits. Thus, we do not consider buffer fullness and underflow conditions.

4.1. Lossy packet model

We assume a wireless medium with a known a priori probabilistic model, which has a feedback channel for error detection and re-transmission. We use the model proposed in [9,28]. This model is simple but meaningful and is used here in order to illustrate the proposed drift compensation scheme. The encoded video frame is partitioned into packets of fixed size of C bits. Packets are indexed sequentially and transmitted at regular time intervals τ_l. These link layer packets are transmitted with a certain success probability p. Packet losses are statistically independent and the lost packet is retransmitted at the next time instant.

Let the frame rate be $1/t_{\mathrm{f}}$ frames/s. Then, the number of packets per frame, N is

$$N = t_{\mathrm{f}}/\tau_l. \quad (9)$$

Let X_i be a random variable where $i = 1, 2, 3, \ldots, N$. $X_i = 1$ denotes a successful packet transmission with probability p and $X_i = 0$ denotes a lost packet with probability $1 - p$. The random variable T_i defines the number of successfully received packets after i transmission attempts,

$$T_i = \sum_{j=1}^{i} X_j \quad \text{where } i \leqslant N. \tag{10}$$

T_i is binomially distributed since we assumed statistically independent packets.

$$\Pr(T_i = j) = \binom{i}{j} p^i (1-p)^{i-j}. \tag{11}$$

Therefore, the received number of bits per frame is given by,

$$R_f = CT_i. \tag{12}$$

At the encoder, the bits per frame will be always equal to, $\text{Renc}_f = CN$.

For example, we consider the case as in [9,28], we set $C = 320$ bits, $p = 0.9$ and $\tau_l = 5$ ms. Let t_f be 0.1 ($1/t_f = 10$ frames/s). Then, the number of packets per frame is

$$N = t_f/\tau_l = 20. \tag{13}$$

Under these conditions, the peak transmission rate will be 64 000 bits/s.

4.1.1. Selection of base rate

In layered coding, the base layer contains vital information and the channel should at least guarantee the lossless delivery of the base layer to the receiver. In [25] the base layer bit rate was arbitrarily chosen and assumed to be always available at the decoder. With the knowledge of success probability and the number of fixed size packets required per frame, we can estimate the base rate for lossless delivery. For our channel model discussed in Section 4.1, the base rate is selected as $R_b = Cj^*_{\text{base}}$, where j^*_{base} is the minimum value of j_{base} that satisfies the inequality,

$$\sum_{i=j_{\text{base}}}^{N} \binom{N}{i} p^i (1-p)^{N-i} \geqslant \Pr_{\text{base}}, \tag{14}$$

where $\Pr_{\text{base}}$ is the probability of successfully receiving the base layer packets.

In the ideal case, $\Pr_{\text{base}}$ would be equal to 1, i.e., absolutely no base layer packet loss. However, when $\Pr_{\text{base}}$ is equal to 1, it results in a zero value for the base rate, which is undesirable. In practice, $\Pr_{\text{base}}$ can be selected to be less than but very close to 1. It is very important to select an optimum $\Pr_{\text{base}}$ value, that will yield a base rate which produces acceptable quality at the receiver. The selection of $\Pr_{\text{base}}$ is discussed in Section 5.1.2.

4.2. Heterogeneous network

In a heterogeneous network, the encoder will generate one bitstream that can be decoded by users operating at different bitrates. The encoder and decoder proposed in Section 3.1.2 can be adapted for a heterogeneous network with some additional considerations discussed below. Let us assume that the rates at which the users are operating is known prior to encoding and let the minimum channel rate be $R_{\min}$ and the maximum rate be $R_{\max}$. With the known range of rate values, we consider the following rate constraints: $R_{\text{base}} \leqslant R_{\min}$ and $R_{\text{base+enh}} \leqslant R_{\max}$, where R_{base} is the base layer bit rate and $R_{\text{base+enh}}$ is the base plus all enhancement layers bit rate at which the encoding is done. The encoder produces a base layer and one or more enhancement layers. The encoder can be optimized for the most expected user rate with agreeable performance for the rest of the users. At the decoder, the base layer is received as it is and the enhancement layer is truncated to match the user rate.

In the adaptive method described in Section 3.1.2, the switching occurs between the Enc_EB and Enc_BB buffers based on the drift measured at the average channel rate. Since we are considering a wider range of channel rates for a heterogeneous network, if the encoding is done using either the Enc_EB or the Enc_BB buffer, we will not get an optimum performance at the intermediate rates. To obtain a better video quality at the intermediate rates, we introduce a third buffer called the intermediate buffer (Enc_IB). The Enc_IB buffer maintains predicted frames based on enhancement layers that match the average channel rate. Now the switching is done among three buffers and the adaptive algorithm is modified as below:

For prediction based on Enc_IB, we calculate $P^{\text{IB}}_{(k+1)_{\text{EB}}}$, $P^{\text{IB}}_{(k+1)_{\text{BB}}}$ and $P^{\text{IB}}_{(k+1)_{\text{AV}}}$ values.

$$\begin{aligned}
\text{MD}^{\text{EB}}_{(k+1)} &= P^{\text{EB}}_{(k+1)_{\text{EB}}} - P^{\text{EB}}_{(k+1)_{\text{AV}}},\\
\text{MD}^{\text{BB}}_{(k+1)} &= P^{\text{BB}}_{(k+1)_{\text{EB}}} - P^{\text{BB}}_{(k+1)_{\text{AV}}},\\
\text{MD}^{\text{IB}}_{(k+1)} &= P^{\text{IB}}_{(k+1)_{\text{EB}}} - P^{\text{IB}}_{(k+1)_{\text{AV}}}.
\end{aligned} \tag{15}$$

The adaptive algorithm for selecting the buffer is given below:

(1) Find minimum drift case:

$$\min\{\mathrm{MD}^{\mathrm{EB}}_{(k+1)}, \mathrm{MD}^{\mathrm{BB}}_{(k+1)}, \mathrm{MD}^{\mathrm{IB}}_{(k+1)}\}. \quad (16)$$

(2) Find minimum distortion case:

$$\max\{P^{\mathrm{EB}}_{(k+1)_{\mathrm{AV}}}, P^{\mathrm{BB}}_{(k+1)_{\mathrm{AV}}}, P^{\mathrm{IB}}_{(k+1)_{\mathrm{AV}}}\}. \quad (17)$$

(3) Use the buffer that meets both Steps 1 and 2.
(4) Else use the enhancement buffer (Enc_EB).

It is not mandatory that the intermediate buffer rate be fixed at the average channel rate. It can also be set closer to a rate at which the majority of users will be operating.

5. Experimental results

A wavelet based video coder is implemented using the low band shift method as explained in Section 2. A Daubechies [11,18] filter with a three-level decomposition is used to compute the wavelet coefficients. The motion estimation is performed in the overcomplete domain using the block matching technique. A 16×16 wavelet block is matched in a search window of $[-16, 16]$. The residues are encoded using the SPIHT coder. In [25], we used the EZW coder to produce the compressed bitstream. Since SPIHT outperformed EZW coder in compression efficiency, SPIHT is the preferred coder in this implementation. We use the "Foreman", "Susie", "News", "Salesman", "Akiyo", "Container" and "Carphone" video sequences to analyze the performance of the proposed coder. A frame rate of 10 frames/s is maintained for all sequences and only one intra frame is used in all the simulations. The results for the two discussed channel models are presented.

5.1. Lossy packet model

Channel rates of 64 kbps and 128 kbps are used to check the performance of the proposed coder. As discussed in Section 4.1, the bitstream is broken down into link layer packets of length $C = 320$ bits. The time interval between two consecutive packets is set to $\tau_l = 5$ ms and $\tau_l = 2.5$ ms for 64 kbps and 128 kbps, respectively. The probability of successful reception p of the link layer packets at the receiver in the proper sequence is a varying factor. Simulations are performed for five different values of $p = 0.85, 0.87, 0.9, 0.93$ and 0.95. The number of packets and the bits per frame for the 64 kbps case are calculated as

$$N = t_{\mathrm{f}}/\tau_l = 20, \quad (18)$$

$$\mathrm{Renc_f} = CN = 6400 \text{ bits}. \quad (19)$$

The encoder assumes that the transmission channel is capable of delivering 6400 bits per frame under lossless conditions. The data per frame include control information, motion vectors, base layer and enhancement layers. In our implementation, we have one base layer and one enhancement layer. The partitioning of the two layers is done according to base layer rate selection. The results presented for each experiment were averaged over 50 simulations.

5.1.1. Drift, ideal and base cases

Drift is introduced in a single layer coder due to prevailing variations in the channel rate. To gauge the performance of our proposed coder, we compared with three different encoder setups.

Case 1: The video sequence is encoded at 6400 bits per frame and all the bits are used to for prediction. The decoder decodes at variable bit rate conditions for each p case. Due to lossy channel condition, the drift is introduced and this is referred as "drift case" in the following discussions.

Case 2: When the encoder has the capability to exactly predict the channel conditions as perceived by the decoder, we obtain the highest efficiency, i.e., the prediction is based on the bitstream that is actually received by the decoder. Though this is not practically achievable, this would serve as the upper bound for the proposed coder. To simulate this condition, we assumed that each frame is encoded using the number of bits that are actually received by the decoder. We refer to this as the "ideal case".

Case 3: We also evaluated the traditional layered coding concept, where only the base layer information is used for prediction and enhancement layers are added only to improve the overall quality. We named this as the "base case" in our result comparisons. The base layer rate is the same rate used by the proposed coder cases.

5.1.2. Base layer selection

As explained in Section 4.1.1, the base layer is a function of $\mathrm{Pr_{base}}$. Hence, we ran simulations for

different values of $Pr_{base} = \{0.9, 0.99, 0.999, 0.9999, 0.99999\}$ that would yield different base layer rates. For each Pr_{base}, we calculate a base layer rate for different p and is shown in Table 1. The results from our simulations for the "Carphone" and "Susie" sequences to identify the base rate are plotted in Figs. 4 and 5. For any p, it is observed that when $Pr_{base} = 0.999$, we get the best quality. When $Pr_{base} > 0.999$, it results in a lower value for base rate, which will reduce the base quality. So when switching is performed, it reduces the overall quality of the decoder output. When $Pr_{base} < 0.999$, the base rates will be higher, but the base layer is also subjected to higher loss probability. Hence we obtain a relatively better performance when $Pr_{base} = 0.999$. Based on these experimental results, $Pr_{base} = 0.999$ is used in the following experiments. The corresponding base rates for $Pr_{base} = 0.999$ are highlighted in Table 1.

5.1.3. Threshold setting

Threshold setting is a crucial parameter and decides the switching instances. The range of ET is approximately identified by calculating the qualities (PSNR) of the received frames for $p = 1$, $p = 0.95$ and $p = 0.85$ in the "Drift case". The lower and upper limits are set from the difference between $p = 1$ and $p = 0.95$ and $p = 1$ and $p = 0.85$, respectively. From our experimental results for different sequences, we found these limits to be between 1.1 and 2.3 dB. BT is used to monitor the Enc_BB quality, which is obviously lesser than the quality of Enc_EB. For this reason, BT should be greater than ET. Our experiments were performed with five different sets of ET and BT, $\{TS_1 = (2, 2.5), TS_2 = (1.6, 1.9), TS_3 = (1.2, 1.5), TS_4 = (0.8, 1.0), TS_5 = (0.4, 0.6)\}$ in dB. Figs. 6 and 7 show PSNR as a function of p for different threshold sets that operate using the optimum base layer rate chosen from Table 1.

From our results, we infer that the threshold set TS_3 gives the best performance. An aggressive approach to contain the output quality with threshold settings TS_4 and TS_5 would result in frequent switching. With frequent switching, prediction is mostly based on Enc_BB and this is very close to

Table 1
Base layer rates for different Pr_{base} and p values

Pr_{base}	0.999999	0.9999	0.999	0.99	0.99
$p = 0.85$	360	400	**440**	520	600
$p = 0.87$	400	440	**480**	520	600
$p = 0.90$	440	480	**520**	560	600
$p = 0.93$	480	520	**560**	600	680
$p = 0.95$	520	560	**600**	640	720

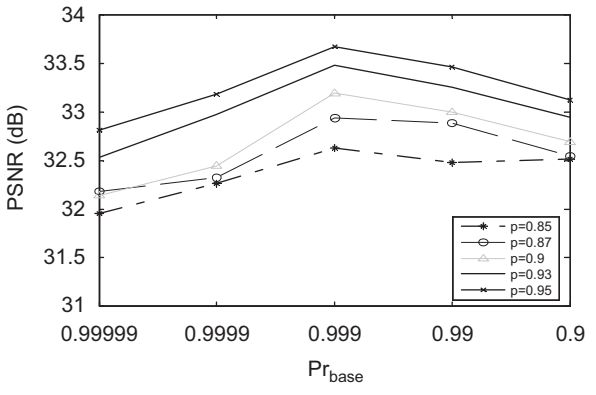

Fig. 4. Base rate selection for "Carphone" sequence in PSNR (dB) vs base rate probabilities Pr_{base} for five p values.

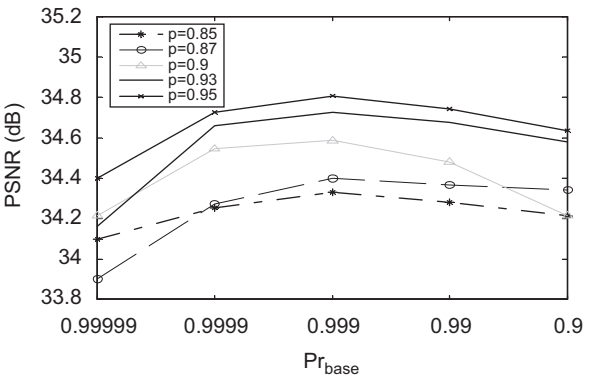

Fig. 5. Base rate selection for "Susie" sequence in PSNR vs base rates probabilities Pr_{base} for five p values.

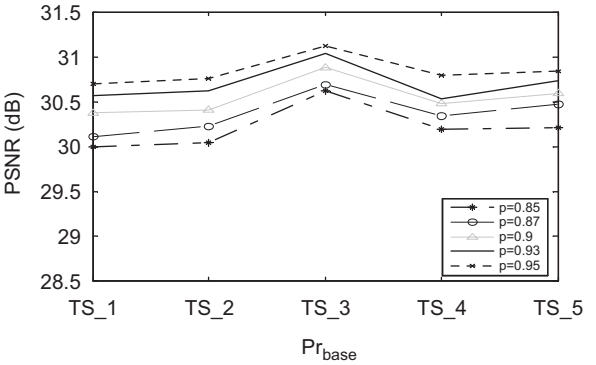

Fig. 6. "Foreman" sequence threshold selection: PSNR vs five threshold sets for different p values.

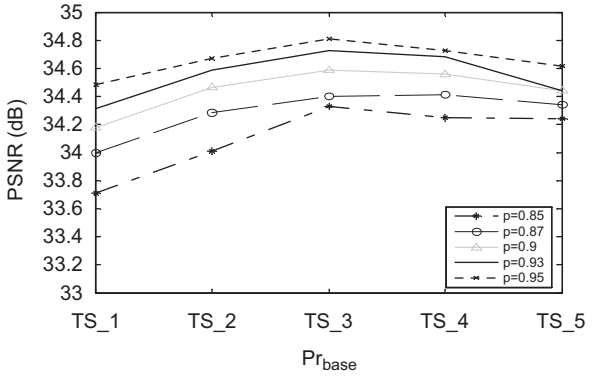

Fig. 7. "Susie" sequence threshold selection: PSNR vs five threshold sets for different p values.

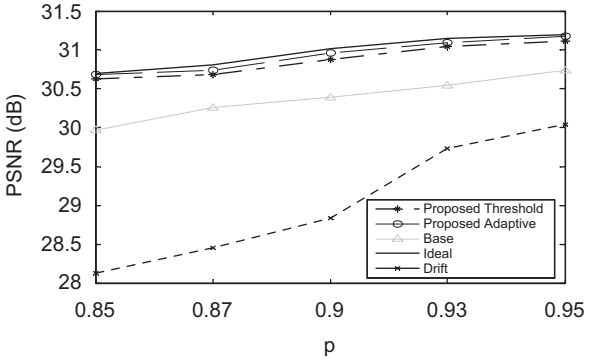

Fig. 8. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Foreman" sequence.

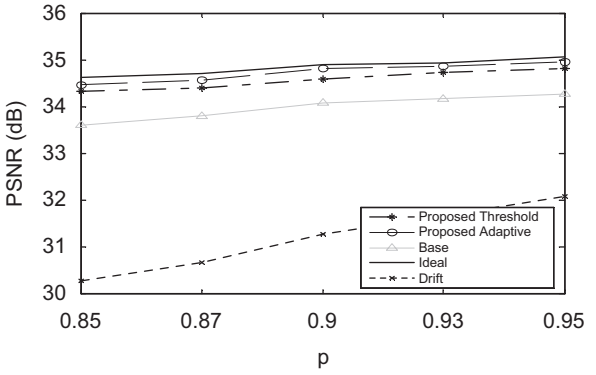

Fig. 9. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Susie" sequence.

traditional coder approach. With a liberal approach, i.e., threshold sets TS_1 and TS_2, switching occurs less often and takes a longer time to offset the decrease in quality introduced by drift.

5.1.4. Results

The average PSNR of the received frames for the "Foreman", "Susie", "Carphone", "Akiyo", "Container" and "Salesman" sequences are plotted in Figs. 8–13, respectively, for different values of p at 64 kbps. For the "News" sequence in Fig. 14, we used 128 kbps as the channel rate. The threshold set is selected as (1.2, 1.5) dB and base rate corresponding to $Pr_{base} = 0.999$ is used for the proposed coder. From the plots we can infer that the adaptive selection of the buffer case always performs slightly better than the threshold case for all sequences. When compared to the "Ideal" case, different sequences show different performance. For the "Akiyo" and "News" sequences, both the adaptive and threshold cases lose in compression efficiency when compared to the "Ideal case". Traditional "Base case" does not suffer from drift but the PSNR is less for the considered cases. The results also show that the proposed coder is very close to the "Ideal case" (unrealizable) and outperforms the "Base case" by 0.4–0.5 dB. From the plots, we can infer that the drift has been regulated without compromising on coding efficiency and quality.

The base layer rate and the thresholds that were used with the "Foreman" and "Susie" sequences were also tested on the "Carphone" sequence. The figures reiterate that the proposed coder performs better than the traditional "Base case" and manages drift efficiently. This also confirms that our selection of the thresholds and the base rates suits all the other sequences considered, which have different motion content.

5.2. Heterogeneous network

The user rate is set to be in the range of 32 to 320 kbps. The base rate is fixed at 32 kbps for both the "Base case" and the "Proposed case". The encoder generates a bitstream at 320 kbps and the encoded bitstream is decoded at {32, 64, 96, 128, 192, 224, 256, 294, 320} kbps. The following cases are considered:

- "Base case": The prediction is always based on base layer only at 32 kbps and the enhancement layers are added to improve the video quality for the rate of interest.
- "Drift case": The video is encoded at 320 kbps and all the bits are used for prediction.

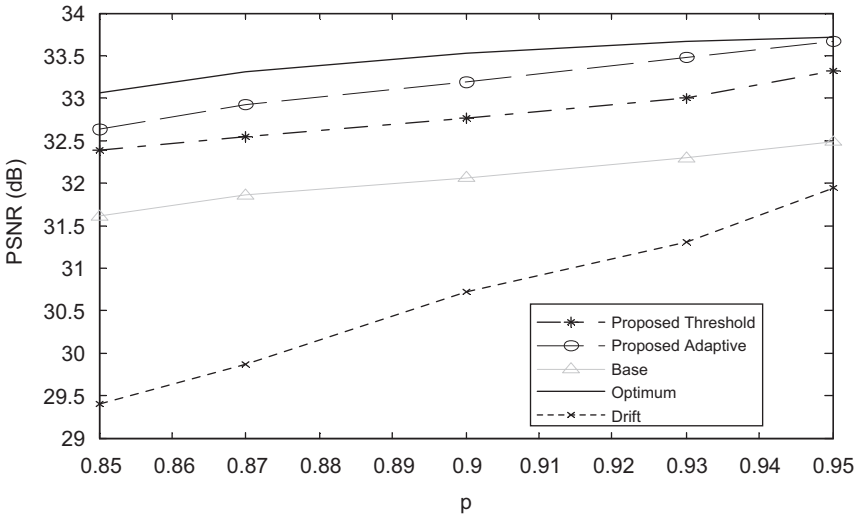

Fig. 10. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Carphone" sequence.

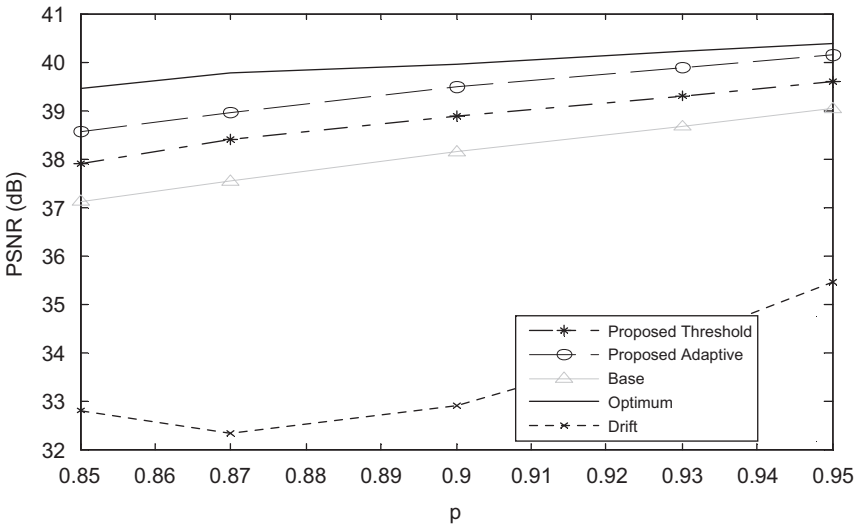

Fig. 11. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Akiyo" sequence.

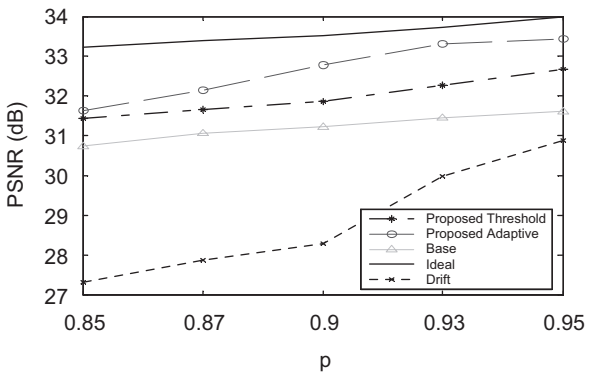

Fig. 12. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Container" sequence.

- "Proposed case": The adaptive method described in Section 4.2 is used to generate a bitstream at 320 kbps.
- "Ideal case": The encoder generates separate bitstreams for the decoder rates considered.

Figs. 15 and 16 show the performance of the proposed coder operating at 32 to 320 kbps for the "Foreman" and "Susie" sequences, respectively. The "Drift case" loses almost 5 dB compared to the "Proposed case" for both sequences. The proposed scheme outperforms the base only prediction case for bitrates greater than 64 kbps and performs very close to the "Ideal case" for rates near the average rate. At bitrates less than 64 kbps, the proposed

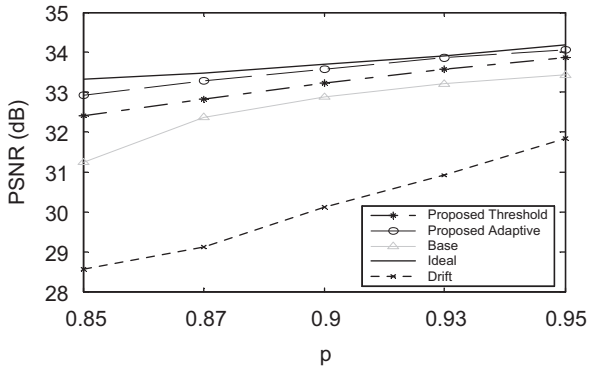

Fig. 13. Packet loss network at 64 kbps: comparison of the proposed techniques for different p values for the "Salesman" sequence.

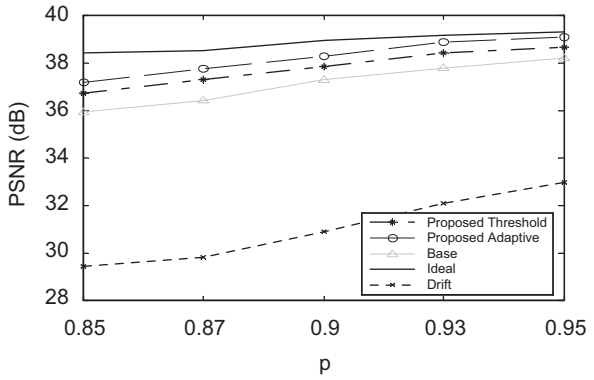

Fig. 14. Packet loss network at 128 kbps: comparison of the proposed techniques for different p values for the "News" sequence.

coder does not perform as well as the "Base case". This is expected since all the bits that were used for prediction for the "Proposed case" will not be available at the lower channel rates. It is to be noted that the "Ideal case" matches the "Base case" at 32 kbps. At higher bitrates, the proposed method outperforms the "Base case" by almost 1.5 db for the "Susie" sequence and 2.5 dB for the "Foreman" sequence.

6. Conclusion

The drift problem in traditional motion compensated predictive coders can be completely eliminated by using the base layer prediction only as in the MPEG4-FGS case. Also, a periodic introduction of intra frames will erase drift. But in both cases, we need more bits to eliminate drift. In wavelet based video coders using 3D subband coding methods, drift is eliminated and high compression efficiency is also achieved. But, the 3D scheme has to process a group of frames to take wavelet transforms and it introduces high coding delays in transmission. We proposed a novel scheme that gives a performance better than the traditional ME/MC coders and without any delays in transmission. Our proposed coder controls drift without significant loss in compression efficiency. Two different channel models were considered and we showed that the proposed encoder and decoder perform efficiently for the considered channel models. Hence, for a given transport channel mechanism, the proposed

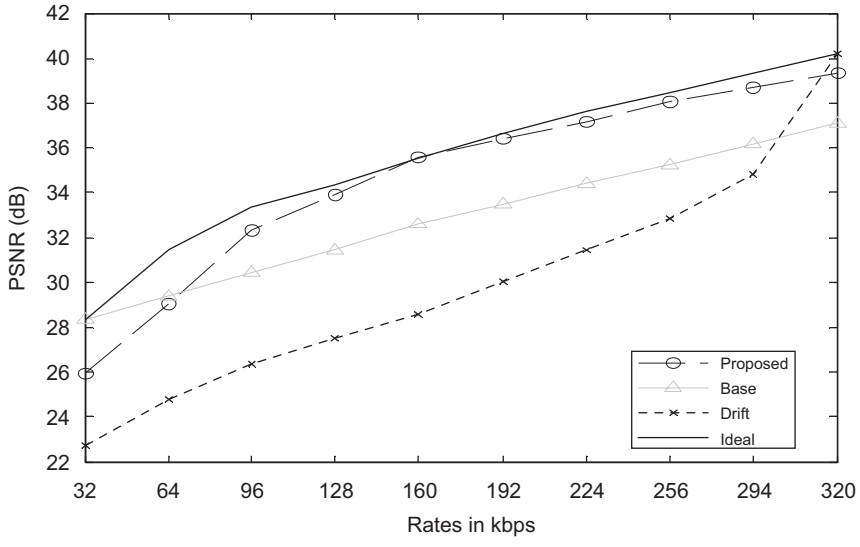

Fig. 15. Heterogeneous network: PSNR vs channel rates for the "Foreman" sequence.

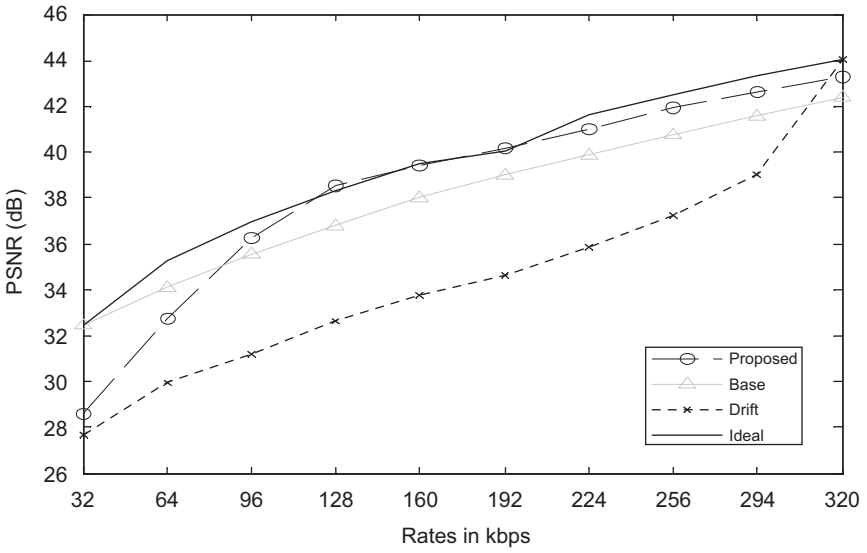

Fig. 16. Heterogeneous network: PSNR vs channel rates for the "Susie" sequence.

scheme offers a drift-free output with high compression efficiency.

References

[1] Y. Andreopoulos, A. Munteanu, G. VanderAuwera, P. Schelkens, J. Cornelis, Wavelet-based fully scalable video coding with in-band prediction, in: Benelux Signal Processing Symposium, Leuven, BE, 2002.

[2] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, Complete-to-overcomplete discrete wavelet transforms for fully scalable coding with MCTF, in: Proceedings of the SPIE Conference on Visual Communications and Image Processing, vol. 5150, 2003, pp. 719–731.

[3] R. Aravind, M.R. Civanlar, A.R. Reibman, Packet loss resilience of MPEG-2 scalable video coding algorithms, IEEE Trans. Circuits Syst. Video Technol. 6 (5) (1996) 426–435.

[4] J.F. Arnold, et al., Efficient drift-free signal-to-noise ratio scalability, IEEE Trans. Circuits Syst. Video Technol. 1 (2000) 70–82.

[5] S. Choi, J. Woods, Motion compensated 3-D subband coding of video, IEEE Trans. Image Process. 8 (1999) 155–167.

[6] S. Cui, Y. Wang, J.E. Fowler, Multihypothesis motion compensation in redundant wavelet domain, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, Barcelona, Spain, 2003, pp. 53–56.

[7] M. Flierl, B. Girod, Investigation of motion compensated lifted wavelet transforms, in: Proceedings of PCS, 2003.

[8] C.-Y. Hsu, A. Ortega, A.R. Reibman, Joint selection of source and channel rate for VBR video transmission under ATM policing constraints, IEEE J. Sel. Areas Commun. 15 (6) (1997) 1016–1028.

[9] H. Jenkac, T. Stockhammer, G. Kuhn, On video streaming over variable bit rate and wireless channels, in: Packet Video, France, 2003.

[10] B.J. Kim, Z. Xiong, W.A. Pearlman, Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Technol. 10 (2000) 1374–1387.

[11] L.P. Kondi, F. Ishtiaq, A.K. Katsaggelos, Joint source-channel coding for motion-compensated DCT-based SNR scalable video wireless systems, IEEE Trans. Image Process. 11 (9) (2002) 1043–1052.

[12] L.P. Kondi, D. Srinivasan, D.A. Pados, S.N. Batalama, Layered video transmission over multi-rate DS-CDMA wireless systems, in: Proceedings of SPIE Conference on Image and Video Communications and Processing, San Jose, CA, 2003, pp. 289–300.

[13] X. Li, L. Kerofski, S. Lei, All-phase motion compensated prediction in the wavelet domain for high performance video coding, in: Proceedings of the IEEE International Conference on Image Processing, vol. 3, Thessaloniki, GR, 2001, pp. 538–541.

[14] J.-R. Ohm, Three dimensional subband coding with motion compensation, IEEE Trans. Image Process. 3 (5) (1994) 559–571.

[15] H.W. Park, H.S. Kim, Motion estimation using low-band-shift method for wavelet-based moving-picture coding, IEEE Trans. Image Process. 9 (2000) 577–587.

[16] B. Pesquet-Popescu, V. Bottreau, Three-dimensional lifting schemes for motion compensated video compression, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2001, pp. 1793–1796.

[17] A.R. Reibman, L. Bottou, Managing drift in a DCT-based scalable video encoder, in: Proceedings of the IEEE Data Compression Conference, 2001, pp. 351–360.

[18] A.R. Reibman, L. Bottou, A. Basso, DCT-based scalable video coding with drift, in: Proceedings of the IEEE International Conference on Image Processing, 2001, pp. 989–992.

[19] A.R. Reibman, L. Bottou, A. Basso, Scalable video coding with managed drift, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 131–140.

[20] A.R. Reibman, H.G. Haskell, Constraints on variable bit-rate video for ATM networks, IEEE Trans, Circuits Syst. Video Technol. 2 (4) (1992) 361–372.

[21] J. Ribas-Corbera, P. Chou, S. Regunathan, A flexible decoder buffer model for JVT coding, in: Proceedings of the IEEE International Conference on Image Processing, vol. 3, Rochester, NY, 2002, pp. 538–541.

[22] A. Said, W. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Technol. 6 (1996) 243–250.

[23] M. van der Schaar, D. Turaga, Unconstrained motion compensated temporal filtering (UMCTF) framework for wavelet video coding, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2003.

[24] A. Secker, D. Taubman, Lifting based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression, IEEE Trans. Image Process. 12 (2003) 1530–1542.

[25] V. Seran, L.P. Kondi, Drift-controlled scalable video coding in the over-complete wavelet domain, in: Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, Niagara Falls, Canada, 2004.

[26] V. Seran, L.P. Kondi, Drift control in variable bitrate wireless channels for scalable wavelet based video coding in the overcomplete discrete wavelet transform domain, in: Proceedings of the IEEE International Conference on Image Processing vol. 3, September 2005, pp. 237–240.

[27] J. Shapiro, Embedded image coding using zerotrees of wavelets coefficients, IEEE Trans. Signal Process. 41 (2003) 3445–3462.

[28] T. Stockhammer, H. Jenkac, G. Kuhn, Streaming video over variable bit-rate wireless channels, IEEE Trans. Multimedia 6 (2) (2004) 268–277.

[29] C. Tillier, B. Pesquet-Popescu, M. van der Schaar, Highly scalable video coding by bidirectional predict-update 3-band schemes, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Montreal, Canada, 2004.

[30] J. Xu, S. Li, Y.Q. Zhang, Z. Xiong, A wavelet codec using 3-D ESCOT, Proceedings of IEEE-PCM2000, December 2000.

[31] R. Zhang, et al., Video coding with optimal inter/intra-mode switching for packet loss resilience, IEEE J. Sel. Areas Commun. 18 (2000) 966–976.