

Robust wireless video multicast based on a distributed source coding approach [☆]

M. Tagliasacchi^{a,*}, A. Majumdar^b, K. Ramchandran^b, S. Tubaro^a

^a*Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci, 32 20133-Milano, Italy*

^b*EECS Department, University of California-Berkeley, Cory Hall, Berkeley, CA 94720, USA*

Received 15 June 2005; received in revised form 1 December 2005; accepted 27 January 2006

Available online 3 May 2006

Abstract

In this paper, we present a scheme for robust scalable video multicast based on distributed source coding principles. Unlike prediction-based coders, like MPEG-x and H.26x, the proposed framework is designed specifically for lossy wireless channels and directly addresses the problem of drift due to packet losses. The proposed solution is based on recently proposed PRISM (power efficient robust syndrome-based multimedia coding) video coding framework [R. Puri, K. Ramchandran, PRISM: a new robust video coding architecture based on distributed compression principles, in: Allerton Conference on Communication, Control and Computing, Urbana-Champaign, IL, October 2002] and addresses SNR, spatial and temporal scalability. Experimental results show that substantial gains are possible for video multicast over lossy channels as compared to standard codecs, without a dramatic increase in encoder design complexity as the number of streams increases.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Video coding; Robust delivery; Scalability; Multicast over wireless networks

1. Introduction

Motivated by emerging multicast and broadcast applications for video-over-wireless, this paper addresses the robust scalable video multicast problem. Examples of such applications include broadcasting TV channels to cellphones, users

sharing video content with others with their PDAs/cellphones, etc. Naturally, in a broadcast setting, each receiving device has its own constraints in terms of display resolution and battery life. Fig. 1 depicts this scenario where each device receives a video stream corresponding to the desired spatial resolution, frame rate and quality. In order to target this class of applications, we need a video coding framework capable of addressing several competing requirements:

- *Robustness to channel losses:* The wireless medium is typically unreliable. For this reason we need to cope with medium to high probabilities of packet/frame losses.

[☆]Parts of this work were presented in [1,2].

*Corresponding author. Tel.: +39 2399 7373;
fax: +39 2399 3413.

E-mail addresses: marco.tagliasacchi@polimi.it (M. Tagliasacchi), abhik@eecs.berkeley.edu (A. Majumdar), kannanr@eecs.berkeley.edu (K. Ramchandran), stefano.tubaro@polimi.it (S. Tubaro).

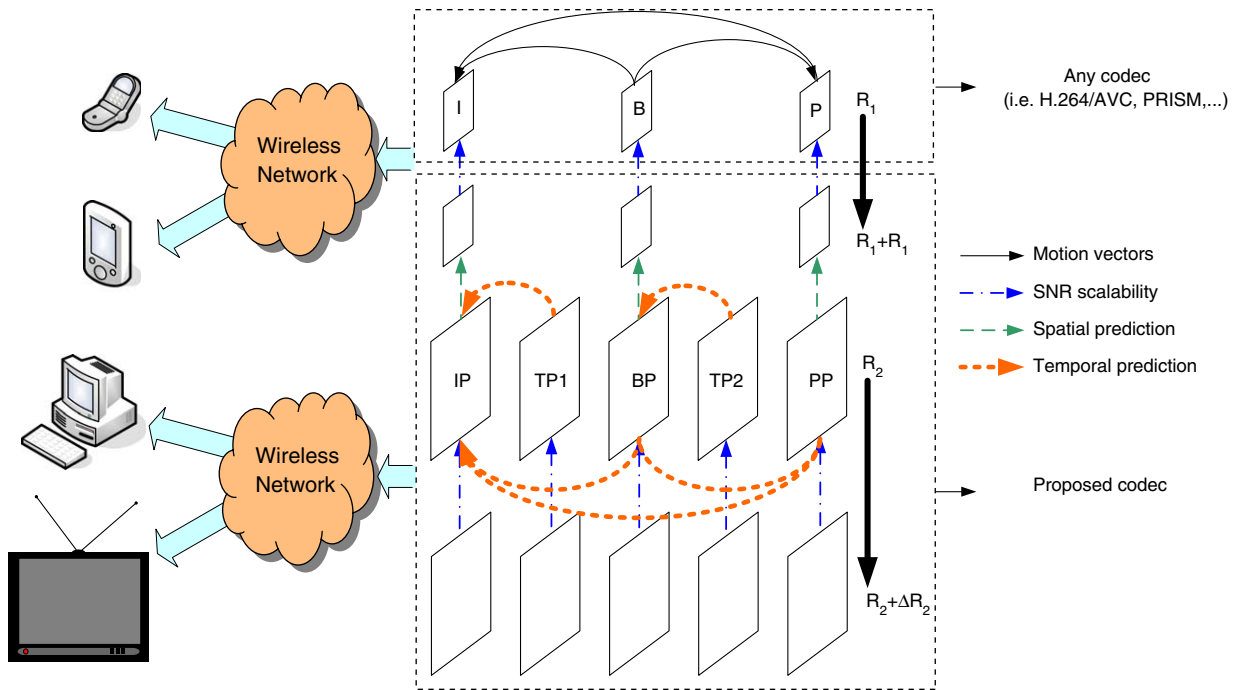


Fig. 1. Each device subscribes to a video stream fitting its characteristics in terms of spatio-temporal resolution and quality. On the right we show the group of picture (GOP) structure adopted in this paper. First, the base layer (I, B and P frames) is encoded. Then, a spatial enhancement layer (IP, BP and PP frames) is built on top of the base layer. Lastly, a temporal enhancement layer is added (TP1 and TP2). Solid arrows represent the motion vectors estimated at the base layer, which are also used as coarse motion information at the enhancement layer. The other arrows point to the frame used as reference to build the side information at the decoder.

- *Scalability in all dimensions*: i.e. spatio-temporal and SNR¹ scalability. In a multicast environment, the receiving devices are heterogeneous, resulting in the need for a flexible bit-stream that can adapt to the characteristics of the receiver. As recommended by the MPEG Ad Hoc group on scalable video coding, at least two levels of spatial and temporal scalability are desirable along with SNR medium granularity scalability (MGS) [3].
- *Lack of “state explosion” at the encoder*: Scalability should not come at too steep a price in encoder complexity. This means that the encoder should not be forced to keep individual state, i.e. keep track of the different reconstructed sequences that can be generated at the several heterogeneous decoders, as is typical in a closed-loop DPCM framework such as MPEG.
- *High coding efficiency*: While achieving the other requirements, any video coding framework should be reasonably competitive with state-of-

the-art non-scalable predictive coders, i.e. H.264/AVC [4].

State-of-the-art closed-loop video coders such as H.264/AVC are able to provide very high coding efficiency by adaptively exploiting a very accurate motion model on a block-by-block basis. Each block is coded with respect to a single deterministic predictor that is obtained by searching over a range of candidates from current and previously encoded frames. Furthermore, to avoid the well-known drift issue the encoder needs to be in sync with the decoder. Although the coding efficiency of this scheme is very good as far as unicast streaming over a noiseless channel is concerned, it fails to meet the aforementioned requirements for video multicast over wireless:

- Being tied to a single predictor, closed-loop coders are inherently fragile in face of channel loss. If the deterministic predictor used at the encoder is not available at the decoder, i.e. because of packet losses, drift occurs as encoder

¹Also referred to as rate or quality scalability.

and decoder work on different data and the errors propagate until the next intra-frame refresh is inserted.

- It is challenging to keep synchronization between encoder and decoder while achieving scalability. Two solutions provided by the standards, i.e. MPEG4-FGS [5] and H.263+ [6] fail to fulfill the requirements stated before: MPEG4-FGS adopts a single loop scheme favoring a simple encoder design at the price of a coding efficiency loss with respect to broadcast. On the other hand, H.263+ uses a multiple loop structure taking into account the presence of different predictors at the heterogenous decoders. Consequently, H.263+ bit-streams suffer less of a hit in terms of loss over the non-scalable case. However, the multiple loop structure leads to added complexity and limits the number of possible rates at which the stream can be decoded

One approach to overcoming these limitations and combating both channel loss and scalability issues at once is to have a more *statistical* rather than a *deterministic* mindset. This motivates the proposed scalable solution based on PRISM [7,8] (Power-efficient, Robust, hIgh compression, Syndrome-based Multimedia coding), a video coding framework built on top of distributed source coding principles. The PRISM codec is inherently robust to losses in the bit-stream and significantly outperforms standard video coders, such as H.263+ for transmission over packet loss channels [9]. Although the theoretical foundations of distributed source coding date back to the theorems of Slepian and Wolf [10] (for lossless compression) and to Wyner and Ziv [10] (for lossy compression) theorems (see Section 2), PRISM represents a concrete instantiation of these concepts to video coding. In a distributed setting, when encoding two correlated variables X and Y , it is possible to perform separate encoding but joint decoding, provided that the encoder has access to their joint statistics. To this regard, the key aspect here is that PRISM does not use the exact realization of the best motion compensated predictor Y while encoding block X , but only the correlation statistics. If the correlation noise between any candidate predictor at the decoder and the current block is within the noise margin estimated at the encoder, the current block can be decoded. Informally speaking, PRISM sends specific bit-planes of the current block X , unlike predictive coders which send information about the

difference between the block and its predictor, i.e. $X - Y$. Consequently, in the PRISM framework, every time a block can be decoded, it has an effect similar to that of intra-refresh (irrespective of any errors that may have occurred in the past). On the other hand, for predictive coders, once an error occurs, the only way to recover is through an intra-refresh. Section 3 briefly reviews the main concepts of the PRISM framework.

Besides PRISM, other video coders based on distributed source coding techniques and exhibiting error resilience properties have also been proposed [11,12]. In [12] the input frames are divided into non-overlapping blocks, DCT transformed and quantized as in intra-frame coding. The Wyner–Ziv encoders sends parity bits of the source. The decoder receives such parity bits and uses them together with the previously decoded frames as side information to decode the current frame. A feedback channel is needed to inform the encoder when no more parity bits are needed. While PRISM performs decoding of each block independently allowing for motion search at the decoder, in [12] the side information is built by pre-warping the reference frame according to a coarse motion information. This motion model is obtained from a lower resolution and heavily quantized representation of the current frame as well as from intra-coded high frequency DCT coefficients.

Scalable video coding has been thoroughly investigated over the last few years. In order to overcome the aforementioned limitations that plague MPEG4-FGS and H.263+, the MPEG Ad Hoc group on scalable video coding has undertaken the study of the most promising technologies capable of addressing the scalability requirements while minimally compromising the coding efficiency vis-a-vis state-of-the-art non-scalable H.264/AVC codecs. The coding architecture that has been chosen to become the new standard is heavily built upon the syntax and tools of H.264/AVC adopts a multi-layered approach [13,14], where each layer improves either the quality or the spatio-temporal resolution of the decoded sequence. The coding scheme we propose in this paper is partially inspired to this architecture as it works in a multilayer fashion.

Recently, scalable video coders based on distributed source coding have been proposed in [15–17]. The algorithm of [15] is similar in philosophy to MPEG4-FGS and the goal is to provide a progressive bit-stream that can be decoded at any rate (within a certain range). In [16] the coding mode is

adaptively switched between FGS and Wyner–Ziv on a block by block basis in order to take full advantage of the temporal correlation existing at the enhancement layer resolution. In [17] a SNR scalable extension of H.264/AVC is proposed where distributed coding is used to prevent the “state explosion” at the encoder. With respect to these coding schemes these proposed solution targets not only SNR but also spatial and temporal scalability. Moreover building on the PRISM framework we provide enhanced robustness.

As mentioned above, the proposed scalable video coding solution is built on the PRISM framework and is designed specifically to provide good performance in the face of channel losses. While the PRISM framework allows for a flexible distribution of complexity between encoder and decoder, in this paper we focus on the case when most of the motion compensation task is performed at the decoder and only part of motion search is done at the encoder. This choice is motivated by the recent results of [18], wherein it was shown (under certain modeling assumptions), that the rate rebate obtained by doing extensive motion search at the encoder decreases as channel noise increases.

It is valid to question the utility of shifting the complexity from the encoder to the decoder (or to share it arbitrarily) when in a codec solution, it is the sum of these complexities that is relevant. To address this, we observe the following network configuration for the PRISM codec (see Fig. 2) introduced in [7]. Here, the uplink direction consists of a transmit station employing the motion-free low-complexity PRISM encoder interfaced to a PRISM decoder in the base station. The base

station has a “trans-coding proxy” that efficiently tailors the decoded PRISM bit-stream for a high-complexity motion-based PRISM encoder which is interfaced to a low-complexity motion-based PRISM decoder on the down-link. Alternatively, it could also convert the decoded bit-stream into a standard bit-stream (e.g. that output by a standard MPEG encoder). The down-link then consists of a receiving station that has the standard low-complexity video decoder. Under this architecture, the entire computational burden has been absorbed into the network device. Both the end devices, which are battery constrained, run power efficient encoding and decoding algorithms.

The paper is organized as follows. We start by summarizing the basic ideas behind Wyner–Ziv coding in Section 2 and the PRISM framework in Section 3. Section 4 thoroughly revises the proposed architecture detailing how spatial, temporal and SNR scalability are achieved. Section 5 contains the results of the simulations carried out with the proposed coding architecture, emphasizing the robustness features.

2. Background on Wyner–Ziv

Consider the problem of communicating two correlated random variables X and Y taking values from a discrete finite alphabet. Separate entropy coding allows the communication of these variables at the rates of $R_X \geq H(X)$ and $R_Y \geq H(Y)$ where $H(X)$ and $H(Y)$ are the entropies of the two sources. It is obviously possible to do better by performing joint encoding, taking advantage of the fact that X and Y are correlated. For this case

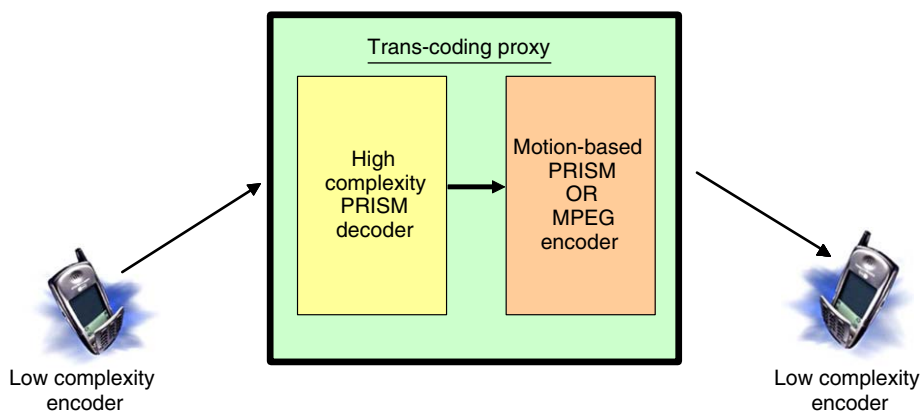


Fig. 2. System level diagram for a network scenario with low complexity encoding and decoding devices.

information theory dictates that the achievable rate region for encoding the sources X and Y is

$$\begin{aligned} R_X + R_Y &\geq H(X, Y), \\ R_X &\geq H(X|Y), \\ R_Y &\geq H(Y|X). \end{aligned} \quad (1)$$

In a distributed source coding setting variables X and Y are separately encoded but jointly decoded. The Slepian–Wolf theorem [19] states that it is possible to attain the same achievable region, with a probability of erroneously decoding X and Y that goes to zero with increasing block length.

These results were extended to the lossy case by Wyner–Ziv [10] a few years later (for the case when Y is known perfectly at the decoder). Again, X and Y are two correlated random variables. The problem here is to decode X to its quantized reconstruction \hat{X} given a constraint on the distortion measure $E[d(X, \hat{X})]$ when the side information Y is available only at the decoder. Let us denote by $R_{X|Y}(D)$ the rate-distortion function for the case when Y is also available at the encoder, and by $R_{X|Y}^{\text{WZ}}(D)$ the case when only the decoder has access to Y . The Wyner–Ziv theorem states that, in general, $R_{X|Y}^{\text{WZ}}(D) \geq R_{X|Y}(D)$ but $R_{X|Y}^{\text{WZ}}(D) = R_{X|Y}(D)$ for Gaussian memoryless sources and MSE as distortion measure. In [20] it was proved that for $X = Y + N$, only the innovations N needs to be Gaussian for this result to hold.

For the problem of source coding with side information, the encoder needs to encode the source within a distortion constraint, while the decoder needs to be able to decode the encoded codeword subject to the correlation noise N (between the source and the side information). While, the results proven by Wyner and Ziv are non-constructive and asymptotic in nature, a number of constructive methods to solve this problem have since been proposed wherein the source codebook is partitioned into cosets of a channel code that is matched to the correlation noise N . The number of partitions or cosets depends on the statistics of N . The encoder communicates the coset index to the decoder. The decoder then decodes to the codeword in the coset that is jointly typical with the side information. Specifically for the problem at hand, we use the concepts detailed in [21] and partition the source codebook into cosets of a multilevel code (as detailed in our earlier work in [9] and briefly summarized in Section 3).

3. Background on PRISM

The PRISM video coder is based on a modified source coding with side information paradigm, where there is inherent uncertainty in the state of nature characterizing the side information (a sort of “universal” Wyner–Ziv framework, see [22] for details). For the PRISM video coder, the video frame to be encoded is first divided into non-overlapping spatial blocks of size 8×8 . The source \mathbf{X} is then the current block to be encoded, while the side information \mathbf{Y} is the best (motion-compensated) predictor for \mathbf{X} in the previous frame(s), where it is assumed that $\mathbf{X} = \mathbf{Y} + \mathbf{N}$. The encoder quantizes \mathbf{X} and then performs syndrome encoding on the resulting quantized codeword; i.e. the encoder finds a channel code that is matched to the noise \mathbf{N} and uses that channel code to partition the source codebook into cosets of the channel code. Intuitively, this means that we need to allocate a number of cosets (therefore, a number of bits) that is proportional to the noise variance. Such noise can be modeled as the sum of three contributions: “correlation noise,” due to the changing state of nature of the video sequence (illumination changes, camera noise, occlusions), quantization noise, since the side information available at the decoder is usually quantized, and channel noise due to packet losses that might corrupt the side information. The encoder transmits the syndrome (indicating the coset for \mathbf{X}) as well as a CRC² calculated on the quantization indices. In contrast to traditional, hybrid video coding, it is the task of the decoder to perform the motion search, as it searches over the space of candidate predictors, one by one, seeking a block from the coset labeled by the syndrome. When the decoded block matches the CRC, decoding is declared to be successful. In essence, the decoder tries successive versions of side information \mathbf{Y} until it finds one that permits successful decoding. Thus, the computational burden of motion estimation is shifted from the encoder to the decoder, so that the encoder is on the same order of complexity as frame-by-frame intra-frame coding.

3.1. Coding strategy

Encoder: The video frame to be encoded is divided into non-overlapping spatial blocks. (We choose blocks of size 8×8 in our implementations.)

²Cyclic Redundancy Checksum.

We now enlist the main aspects of the encoding process.

1. *Classification*: Real video sources exhibit a spatio-temporal correlation noise structure whose statistics are highly spatially varying. Within the same frame, spatial blocks that are a part of the scene background are highly correlated with their temporal predictor blocks (“small” N). On the other hand, blocks that are a part of a scene change or occlusion have little correlation with the previous frame (“large” N). This motivates the modeling of the video source as a composite or a mixture source where the different components of the mixture correspond to sources with different correlation (innovation) noise. In our current implementation, we use 16 classes corresponding to the different degrees of correlation varying from maximum to zero correlation. These classes range from the SKIP mode at one hand where the correlation noise is so small that the block is not encoded at all, to the INTRA mode at the other extreme, corresponding to high correlation noise (poor correlation), so that intra-coding is appropriate. The appropriate class for the block to be encoded is determined by thresholding the scalar mean squared error between the block and the co-located block in the previous frame. The thresholds T_p and T_{p+1} corresponding to the p th class were chosen using offline training. The corresponding block correlation noise N^p vector is considered in the DCT domain where it is modeled as a set of independent Laplacian

random variables $\{N_1^p, N_2^p, N_3^p, \dots\}$. The choice of this model was based on its success as reported previously in literature [23] and by our experiments on statistical modeling of residue coefficients in the transform domain. These classes correspond to different quantization/syndrome channel code choices. The 4-bit classification/mode label for a block, based on the thresholding of its mean squared error with a co-located block in the previous frame, is included as part of the header information for use by the decoder.

2. *Decorrelating transform*: We apply a DCT on the source block. The transformed coefficients X are then arranged in a one-dimensional order by a doing a zig-zag scan on the two-dimensional block.
3. *Quantization*: The scanned transformed coefficients are scalar quantized with the target quantization step size. The step size is chosen based on the desired reconstruction quality.
4. *Syndrome coding*: The quantized codeword sequence is then syndrome encoded.
 - *Multilevel coset codes*: Consider the DCT coefficient X as the source and an m -level partition (see Fig. 3) of a lattice. At each level i , a subcodebook is completely determined by a bit, B_i , for that level and $i - 1$ bits from previous levels, $B_k; 1 \leq k \leq i - 1$. Encoding may then proceed by first quantizing X to the closest point in the lattice at level 0, and then determining the path through the partition tree to the subcodebook at level m , that contains the codepoint representing X . The path will specify the source bits, $B_i; 1 \leq i \leq m$,

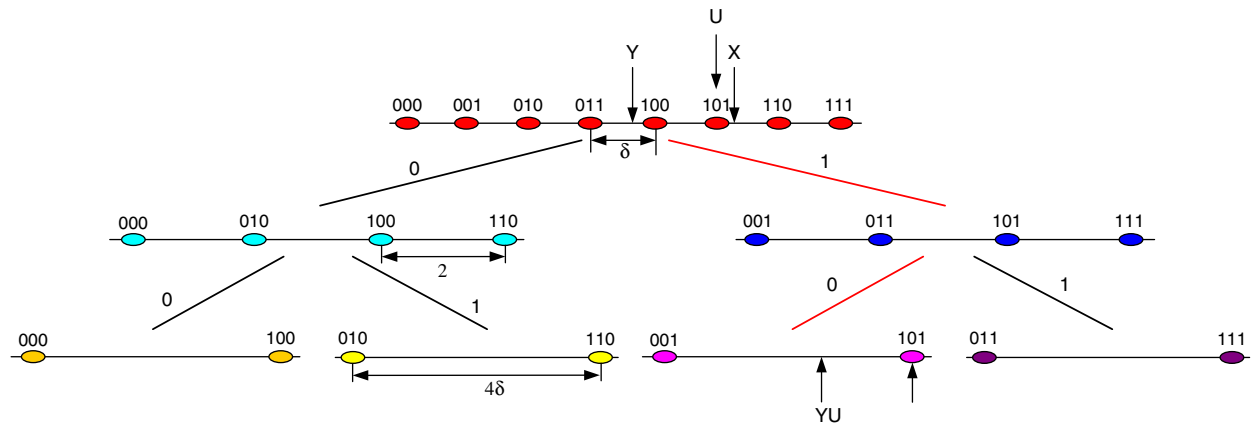


Fig. 3. Multilevel coset coding: partitioning the integer lattice into three levels. X is the source, U is the (quantized) codeword and Y is the side information. The number of levels in the partition tree depends on the effective noise between U and Y given X .

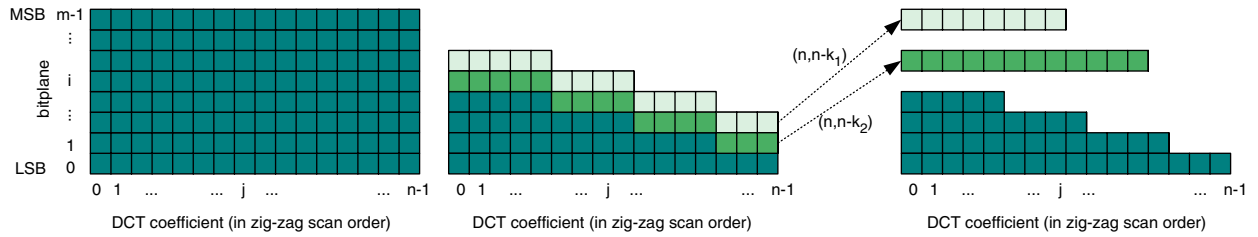


Fig. 4. Syndrome-based encoding of a DCT transformed block. Left: original DCT coefficients. Middle: based on the correlation noise estimate, only the least significant bits of each DCT coefficient are sent. Right: a further rate rebate can be obtained by syndrome encoding the most significant bitplanes.

that are to be transmitted to the decoder. The number of levels in the partition tree can be varied based on the estimated variance of the effective noise between X and Y as shown in Fig. 4, where for each coefficient X_j we assign a different number of levels m_j . The value of m_j also depends on the class the block belongs to, as determined in the classification step.

- *Syndrome generation*: The output of the previous stage can be sent uncoded or can be further processed in order to reduce the rate. The most significant bits of each DCT coefficient can be grouped together to form a binary channel codeword of size n and can then be passed through a parity check matrix of an (n, k) linear error correction code producing in output a syndrome of size $n - k$ bits. The encoding rate will then be $(n - k)/n$. The same procedure can then be applied to lower bitplanes by changing accordingly the rate of the error correction codes. It is clear that low-rate error correction codes, which usually correspond to stronger error correction capabilities, will result in higher encoding rates. Thus, lower levels will require higher encoding rates, because they will have higher uncoded probabilities of error, which comes from the lower correlation with the side information, and therefore demand stronger error correction codes. In practice, the choice of rates (channel codes) for each level should be done jointly to minimize the end-to-end expected distortion. Since, the expected distortion depends on the probability of error, so a set of error correction codes should be chosen to achieve a desired probability of error. This can be done by modeling the test channel to be characterized by the correlation noise \mathbf{N} which was discussed earlier. The probability of error can then be calculated

either analytically or empirically based on the overall noise statistics.

Decoder: For each block the decoder searches candidate blocks taken from the reference frame to be used as side information. Usually, candidate blocks are visited in spiral order starting from the co-located block in the reference frame. For each of them the decoded codeword is obtained by performing multistage decoding that is initiated by decoding the lowest level and then passing the decoded bit to the next level. Each decoded bit is passed to successive levels, until all bits are decoded and an associated codeword is obtained. At each level, a syndrome is received from the encoder. This syndrome can be used to choose a coset of the linear error correction code associated with that level, and then perform soft decision decoding [24,25] on the side information to find the closest codeword within the specified coset. Thus, for each candidate predictor a reconstructed version of the current block is obtained. In order to determine if this reconstruction is correct, a CRC is computed from the reconstructed quantized coefficients and it is compared with the CRC sent by the encoder. If the CRC matches, decoding is declared successful. In our simulations we have never found the CRC to match when the decoded codeword is actually wrong. We need to emphasize that this method grants high robustness in face of channel loss. In fact, when the best motion compensated candidate predictor is not available, decoding might still succeed using other candidate predictors taken from the same reference frame as well as from past frames.

4. Proposed video multicast solution

Building on the PRISM framework, we propose a coding scheme that provides spatial and temporal

scalability based on the principles of distributed video coding. This scalable flavor of PRISM is designed specifically to offer good performance in the face of channel losses.

The proposed architecture is inspired to what has been chosen to become the future scalable video coding standard [14], as an extension of H.264/AVC [4]. First, a multilayer representation of the sequence is built by spatially downsampling the original frames. Fig. 1 gives an example where only two layers are shown. Although the extension to multiple layer is conceptually straightforward, in this paper we refer to a two-layer scheme, where the base layer has half of the resolution of the enhancement layer. First, the base layer is encoded using any coding algorithm. Backward compatibility can be assured at the base layer if a standard codec is used, i.e. H.264/AVC [4]; H.263+ [6] or MPEG2 [26]. In this work we have adopted an IBPBP group of pictures (GOP) structure so that the first temporal scalability layer is supported. For example, if the full spatio-temporal resolution sequence is CIF@30 fps,³ then by decoding the base layer only, we obtain a sequence at QCIF@15 fps or QCIF@7.5 fps (by skipping the B frames). As mentioned in Section 1, in this work we will focus on the case when the encoder does only part of the motion estimation task and most of the motion search is performed at the decoder. In fact the encoder performs motion estimation only at the base layer resolution, at a fraction of the cost of full resolution motion search. This is motivated by the fact that in this paper we are mostly concerned about robustness to channel loss. To this end, it was recently shown that the importance of estimating an accurate motion model at the encoder decreases when the channel noise increases [18].

The base layer quality can be improved from rate R_1 to rate $R_1 + \Delta R_1$ with a SNR enhancement layer encoded as explained in Section 4.2, in such a way that different users can decide to subscribe to the stream they are interested in according to their network bandwidth constraints. Like H.263+ we want to be able to exploit the temporal correlation at the SNR enhancement layer in order to minimize the coding efficiency loss of MPEG4-FGS. At the same time we do not want to keep multiple loops at the encoder tracking different decoder states. Using PRISM, we encode the enhancement layer based on the statistical correlation between the original

sequence and the side information, that can be generated from the SNR enhancement layer of previously decoded frames as well as from the base layer of the current frame.

The spatial enhancement layer is encoded on top of the higher quality base layer with the proposed distributed source coding approach detailed in Section 4.3. The frames labeled IP, BP and PP form the spatial enhancement layer (achieving CIF@15 fps) and these frames can leverage the base layer as a spatial predictor as well as previously decoded frames as temporal predictors.

Subsequently, the temporal enhancement layer is added (frames TP1, TP2) in order to obtain the full resolution version of the sequence, CIF@30 fps. In both cases, the motion information available in the base layer is exploited to build the temporal predictor as will be detailed in Section 4.4. The main issue here is to tune the estimation of the statistical correlation based on the temporal distance between the frame to be encoded and its reference.

A further SNR scalability layer can be added in order to improve the quality at full spatial resolution increasing the target bitrate to $R_2 + \Delta R_2$. Therefore, in our current implementation we are able to decode the sequence at two target bitrates for each spatial and temporal resolution.

The proposed scalable solution inherits the robustness features of PRISM, when video is streamed over a noisy channel. Experimental results (see Section 5) demonstrate that it outperforms state-of-the-art predictive-based video codecs at medium to high packet loss rates even when forward error correcting (FEC) codes are used to prevent errors. Furthermore, the layered organization of the bit-stream makes the proposed solution amenable for unequal error protection (UEP) in order to further improve its robustness.

4.1. Information theoretic setup

With respect to Fig. 5 we explain the encoding/decoding of an enhancement layer on top of a base layer. We consider here an information theoretic perspective, postponing to the next sections the description of the actual coding algorithm.

Decoder 1 has a rate constraint of R , while decoder 2 has a rate constraint of $R + \Delta R$. \mathbf{Y}_b and \mathbf{Y}_g are the predictor blocks (from previously decoded frame(s)) available to decoders 1 and 2, respectively. \mathbf{Y}_b and \mathbf{Y}_g form the side informations

³CIF resolution: 352×288 , QCIF resolution 176×144 .

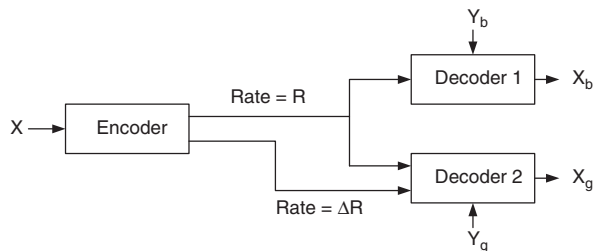


Fig. 5. SNR scalability: decoder 1 subscribes to the base layer stream at rate R while decoder 2 to both streams at rate R and ΔR . Y_b and Y_g are the side informations respectively available at the two decoders. Y_g is a better side information than Y_b .

for the respective decoders. Since decoder 2 receives data at a higher rate, it will have a better predictor (and hence a better side information) than the decoder 1. In the case of SNR scalability, Y_b and is generated from previously decoded frames at rate R , while Y_g from previously decoded frame at rate $R + \Delta R$ as well as from the same frame decoded at rate R . The same scenario holds for spatial scalability, where the rate increment between the base and the enhancement layer ΔR is used to increase the spatial resolution instead of improving the reconstruction quality. X_b and X_g are the reconstructions of the source X by decoders 1 and 2, respectively. X_g is a better reconstruction than X_b .

Heegard and Berger [27] provided the optimal rate-distortion region for this problem for the case when $\Delta R = 0$. Steinberg and Merhav [28] have recently extended the result of [27] to cover the case of non-zero ΔR , where $X - Y_g - Y_b$ forms a Markov chain. The Markov chain implies that the lower rate user's side information is a degraded version of the better user's. The entire optimal rate-distortion region for this problem is provided in [28]. In the interests of simplicity, we will restrict ourselves to one important operating point in this region. This point corresponds to the case where the entire rate R can be utilized by decoder 1.

The solution for this case calls for the generation of two source codebooks C_1 and C_2 . The rate of codebook C_1 is R while that of C_2 is ΔR . The source X is quantized using C_1 and C_2 to generate the codewords U and W , respectively. Conceptually the decoding process is as follows: the codeword U is first decoded by both decoders. X_b is the reconstruction by decoder 1 and let X'_g be the reconstruction by decoder 2. X'_g is a better reconstruction of X than X_b due to greater estimation gains (because of the presence of the better side information at decoder 2). Note that this estimation gain comes from

multiple independent looks at the source data [10]. Now, the codeword W is decoded using X'_g as the side information. Note that it would be sub-optimal here to assume that the reconstruction by decoder 2 is also X_b and we get a rate rebate by using the better reconstruction X'_g .

Multiple users: The extension to more than two users is relatively straightforward. For example, let there be a third client in the system with a rate constraint of $R + \Delta R + \Delta R'$. Then we will encode the R and ΔR bit-streams just like in the two-client case while the new $\Delta R'$ bit-stream will be coded keeping in mind the better reconstruction that the third client has after it has decoded the R and ΔR bit-streams. This allows to target MGS (medium granularity scalability).

Unlike the H.263+ encoder, our encoder needs to maintain a relatively small amount of "state" information relating to the statistical correlation between the current frame and the different predictors at the decoders. While details depend on the exact implementation (e.g. a single scalar quantity representing the estimated correlation noise might suffice), the key difference is that in the predictive coding framework, *deterministic* copies of each predictor frame need to be kept in the encoder state. This allows our algorithm to scale with the number of users.

4.2. SNR scalability

Fig. 1 shows that two SNR scalability layers are made available both at the base layer and at the spatial enhancement layer resolution.

The encoding process of the SNR scalability layer follows the algorithmic steps of the PRISM codec described in Section 3. Each block having size 8×8 is encoded independently with the previously decoded blocks at the decoder serving as the side information.

As in Section 4.1, let us again consider the case when the entire rate R can be utilized by decoder 1 (see Fig. 5). As in the single client case, an estimate of the correlation noise between the block to be encoded and the side information is needed. For this purpose we use the frame-difference-based classification algorithm described in Section 3.1. Since the entire rate R can be utilized by decoder 1, the design of the first codebook C_1 (using the notation of Section 4.1) is identical to the single client setup described in Section 3. The second codebook C_2 essentially consists of extra bit-planes that can

further refine the reconstruction at decoder 2. Since, the side information \mathbf{Y}_g (present at decoder 2) is better than that of \mathbf{Y}_b (present at decoder 1), these bit-planes can be further compressed using channel codes to achieve rate-savings.

At the decoder, the side information can be obtained either from the decoded base layer at rate R and/or from the previously decoded frames at rate $R + \Delta R$. The decoding process for the first codeword (U) is identical to that described for the one-client case in Section 3.1. Each client will independently perform motion search to generate side information that can be used to correctly decode the codeword. Upon decoding U , decoder 1 will reconstruct the source \mathbf{X} to \mathbf{X}_b , and decoder 2 will reconstruct \mathbf{X} to \mathbf{X}'_g . The decoder 2 now needs to decode the second codeword (W). At this step, \mathbf{X}'_g will serve as the side information to the decoder. The decoding process is identical to regular Wyner–Ziv decoding.⁴

4.3. Spatial scalability

In the proposed solution, the spatial enhancement layer is encoded on top of the higher quality base layer. As shown in Fig. 1 when it comes to encode frames IP, PP and BP both the base layer and the previously decoded frames at the enhancement layer can serve as side information. Moreover, since the enhancement layer encoder is not allowed to perform any motion search, the correlation noise between the current block and the unknown best predictor, that will be revealed only during decoding, needs to be computed in a computationally efficient manner. To this end, in the original (non-scalable) version of PRISM [7], each block \mathbf{X} is classified according to the mean square error computed using as a predictor the block in the reference frame at the same spatial location, i.e. using a zero motion temporal predictor, block \mathbf{Y}^{TZM} . An offline training scheme provides an estimate of the correlation noise for each DCT coefficient based on the measured MSE and the best motion compensated predictor that will be available at the decoder, \mathbf{Y}^{TFM} .⁵ Unfortunately, this method is likely to fail when there is significant, yet simple, motion such as camera panning. The proposed solution takes advantage of the existence of the base layer in

two different aspects: as a spatial predictor for those blocks that cannot be properly temporally predicted, e.g. because of occlusions, as well as using the motion vectors available at the coarser resolution to provide a better estimate of the correlation noise. The encoding process for the three types of frames are as follows:

- *Frame IP*: A spatial predictor is computed by interpolating the quantized I frame of the base layer. The prediction residual is quantized and entropy coded as in H.263+.
- *Frame PP*: Spatial, temporal and spatio-temporal predictors are built using only the coarse motion vectors of the base layer (see Fig. 6). Then, the best predictor is chosen according to a MSE metric and the correlation noise is estimated based on the statistics collected offline. The block is then quantized and encoded as described in Section 3, sending only the least significant bits of the DCT coefficients as the most significant ones will be recovered using the side information.

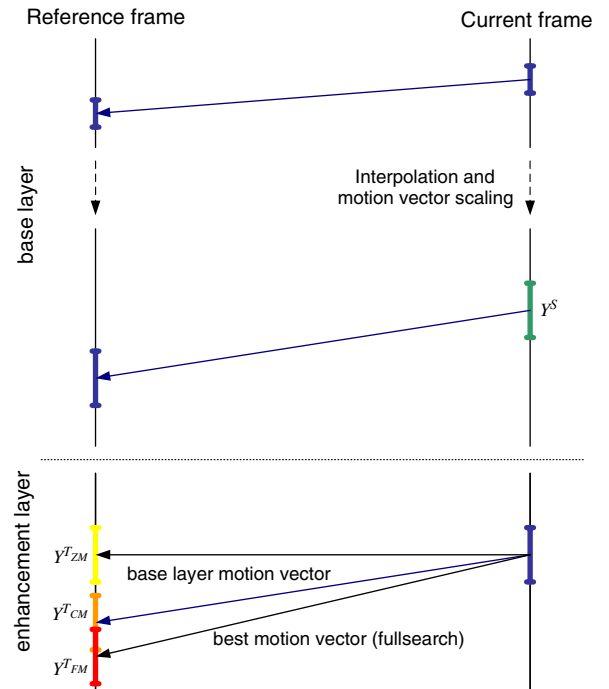


Fig. 6. When encoding block \mathbf{X} the encoder has access to its spatial predictor \mathbf{Y}^S and the coarse temporal predictor \mathbf{Y}^{TCM} obtained by scaling the base layer motion vector. At the decoder also the best motion-compensated predictor \mathbf{Y}^{TFM} is available as side information. This figure does not show the spatio-temporal predictors available at the encoder (\mathbf{Y}^{STCM}) and at the decoder (\mathbf{Y}^{STFM}) computed as a simple average between the spatial and temporal predictors.

⁴Note that since there is no further motion search at this step, no CRCs are required to verify decoding of W .

⁵Subscript FM stands for full motion.

- *Frame BP*: The encoding algorithm is similar to that of the PP frame, except for the fact that the temporal predictor can use the forward and/or backward motion vectors (bi-directional prediction). The prediction mode as well as the motion vectors are the same used in the base layer.

At the decoder, the algorithm tests different predictors until it finds one that is good enough to correctly decode the block. If the CRC of the decoded block matches the one computed at the encoder side, a decoding success is declared. The decoder is allowed to use any predictor—spatial, temporal and spatio-temporal—for the purpose of decoding the encoded block.

As mentioned above, the correlation between the block to be encoded and the (unknown) best predictor (which will be found at the decoder after a full motion search) needs to be estimated. This is the task of the classifier. At the encoder, only the motion information available at the base layer (termed the “coarse” motion vector) is used to provide an estimate of the correlation. Three different prediction types are allowed—spatial \mathbf{Y}^S , temporal $\mathbf{Y}^{T_{CM}}$ and spatio-temporal $\mathbf{Y}^{ST_{CM}} = (\mathbf{Y}^S + \mathbf{Y}^{T_{CM}})/2$ (see Fig. 6)—and the best among these choices is computed based on the “coarse” motion vector. The classifier works on training sequences. Using the data collected offline, the classifier estimates the correlation statistics between the block to be encoded and the best motion compensated predictor available at the decoder (either spatial \mathbf{Y}^S , temporal $\mathbf{Y}^{T_{FM}}$ or spatio-temporal $\mathbf{Y}^{ST_{FM}} = (\mathbf{Y}^S + \mathbf{Y}^{T_{FM}})/2$) for each type of predictor that can be selected at the encoder (\mathbf{Y}^S , $\mathbf{Y}^{T_{CM}}$ and $\mathbf{Y}^{ST_{CM}}$). See Fig. 7 for more details.

Although only two levels are considered in the current implementation, the proposed scheme supports any number of levels of spatial scalability. In fact, the same concepts can be extended to a multilayer architecture, where each layer can use the upper layer as a spatial predictor. Furthermore, the ratio between the resolutions of two succeeding layers is not constrained to be 2:1. All that is needed is an interpolating algorithm that is able to build the spatial predictor of the appropriate size starting from the base layer.

4.4. Temporal scalability

Fig. 1 shows that by encoding frames TP1 and TP2 is possible to get full spatio-temporal resolu-

tion. The encoding of the temporal enhancement layer is more involved since we can rely only partially on the information available at the base layer. Specifically, we have neither a spatial predictor available nor a motion field that covers completely the frame to be encoded. For these reasons we allow only temporal prediction. The motion field is inferred by that available at the base layer. In our current implementation, the estimation of the coarse motion field for TP1 frames proceeds as follows. First, the motion field of frame BP is extracted from the base layer by simply scaling the motion vectors in order to match the spatial resolution of the enhancement layer. Then, the motion field of frame TP1 is estimated by “interpolating” the motion trajectories from BP to IP (or PP). Fig. 8 gives a pictorial representation of the algorithm and it shows the different scenarios that can happen:

- Block *a* is completely covered by the projection of block *A* and no other block overlaps with it. We apply to block *a* the scaled version of \mathbf{MV}_A , i.e. $\mathbf{MV}_a = \mathbf{MV}_A/2$.
- Block *b* is only partially covered by the projection of block *B*. As before, we apply to block *b* the scaled version of \mathbf{MV}_B , i.e. $\mathbf{MV}_b = \mathbf{MV}_B/2$.
- Block *c* is covered by the projections of block *B* and *C*. The motion vector of the block that covers the most is selected; so $\mathbf{MV}_c = \mathbf{MV}_C/2$.
- Block *e* is covered by the projections of blocks *E*, *D* and *F*. As before, the block with the widest coverage, i.e. *E*, is selected and its scaled motion vector is assigned to block *e*.
- Block *g* is not covered by any block. In this case we can either use the zero motion vector or assign a vector that is estimated from its causal neighbors, i.e. blocks *d* and *e* in this case.

Although more sophisticated methods can be used for this operation, the overall coding algorithm is not very sensitive to the accuracy of the coarse motion field. In fact the coarse motion vector is used to determine MSE_c . Based on the value of MSE_c , the block is assigned to one of the classes, therefore, driving the coset bit allocation. Similar values of MSE_c thus lead to the same decision in the classification process.

We have to point out that the backward motion vector from BP to IP (or PP) might not be available in the base layer. This can happen in two circumstances: the block is intra-coded or the block

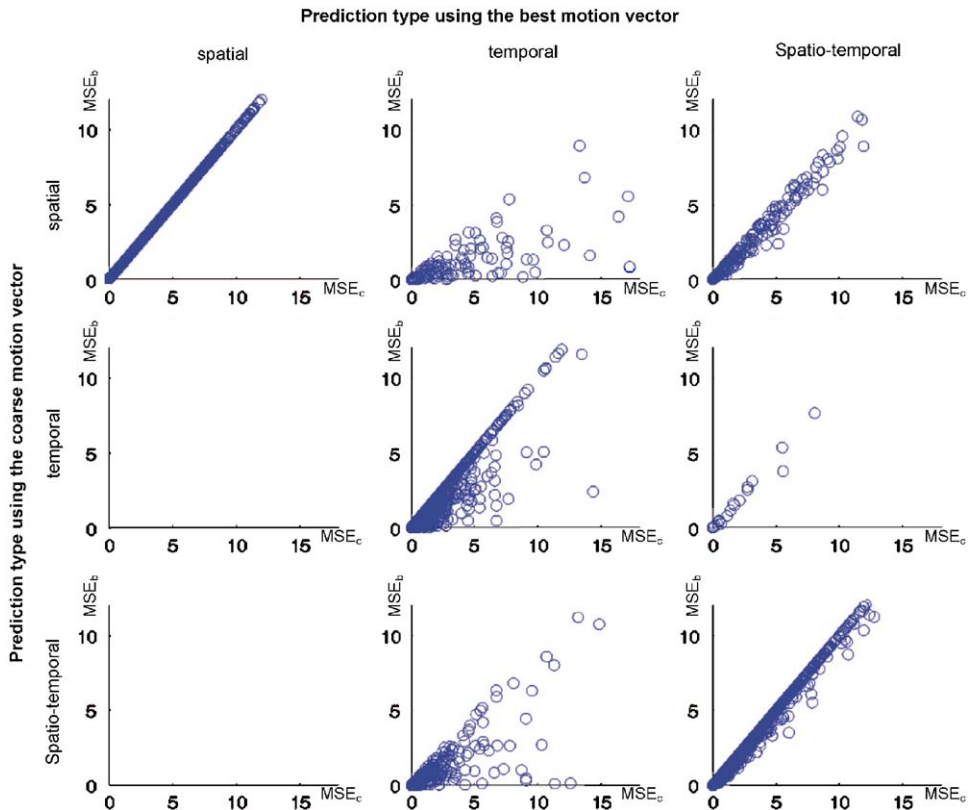


Fig. 7. Working offline, the classifier computes a mapping between the residue computed using the “coarse” predictor and the residue computed using the best predictor obtained from a full motion search. Each block is represented by a circle in one of the nine scatter plots, according to the prediction type computed using the coarse motion vector available at the encoder (determining the row) and the best motion vector (determining the column). In each scatter plot, the x -axis is the MSE computed using the coarse motion vector (MSE_c), while the y -axis the MSE computed with the best motion vector (MSE_b). MSE_b is an aggregate measure of the correlation noise at the block level and it is not directly used in the actual encoding algorithm. In fact MSE_c determines the class a block belongs to (see Section 3). For each class, the MSE of each DCT coefficient is estimated offline and used to drive the coset bit allocation.

is inter-coded but only the forward motion vector is available. In the former case, the same policy adopted for blocks not covered by any projection is applied. In the latter case, the backward motion vector is obtained by simply inverting the forward motion vector.

The estimation of the motion field of frame TP2 follows the same algorithm as that for frame TP1. In this case we can leverage either the backward motion field from PP to IP (or PP) or the forward motion field from BP to PP.

We note that separate statistics of the correlation noise are collected for each type of frame. This is due to the fact that the distance between the current frame and its temporal reference is different for each type of frame. Hence the accuracy of the estimated coarse motion field varies with frame type (typically the motion fields estimated for the frames of type

BP and PP are more precise than for the frames of type TP1 and TP2).

5. Experimental results

In this section we present results to showcase the promise of our approach. In Section 5.1 we present results for SNR scalability, followed by results for spatial and temporal scalability in Section 5.2. In all experiments the GOP size is equal to 32 frames. Therefore, one intra-coded frame is inserted every 16 frames at 15 fps or every 32 frames at 30 fps.

5.1. SNR scalability tests

We present results for the two client/two rate case using the “uplink” PRISM framework (i.e. one in which motion compensation is performed at the

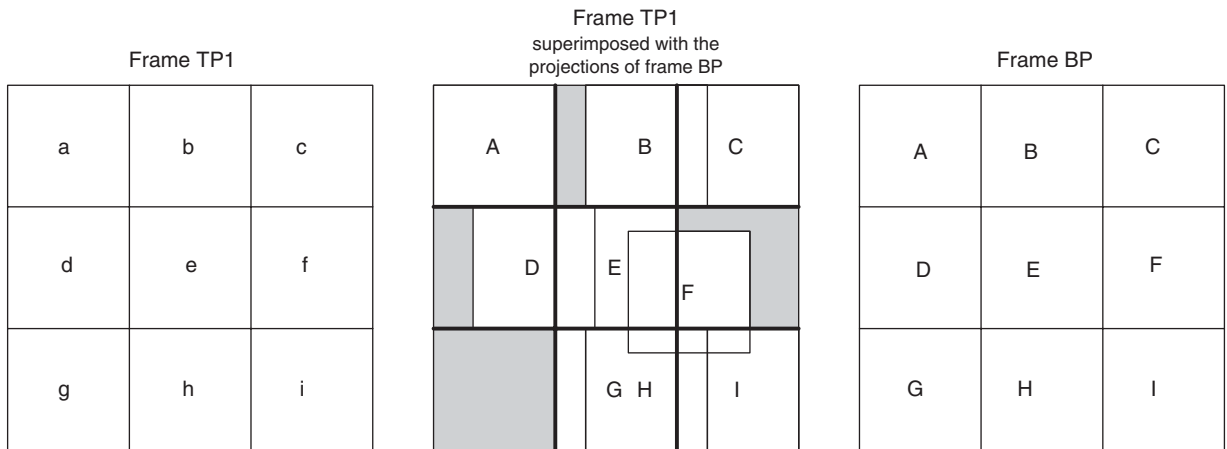


Fig. 8. Estimation of motion field of frame TP1 from motion field of frame BP. Block *e* is covered by the projections of blocks *E*, *D* and *F*. The block with the maximum overlap, i.e. *E*, is selected and so $MV_e = MV_E/2$. Similarly, $MV_a = MV_A/2$, $MV_b = MV_B/2$ and $MV_c = MV_C/2$.

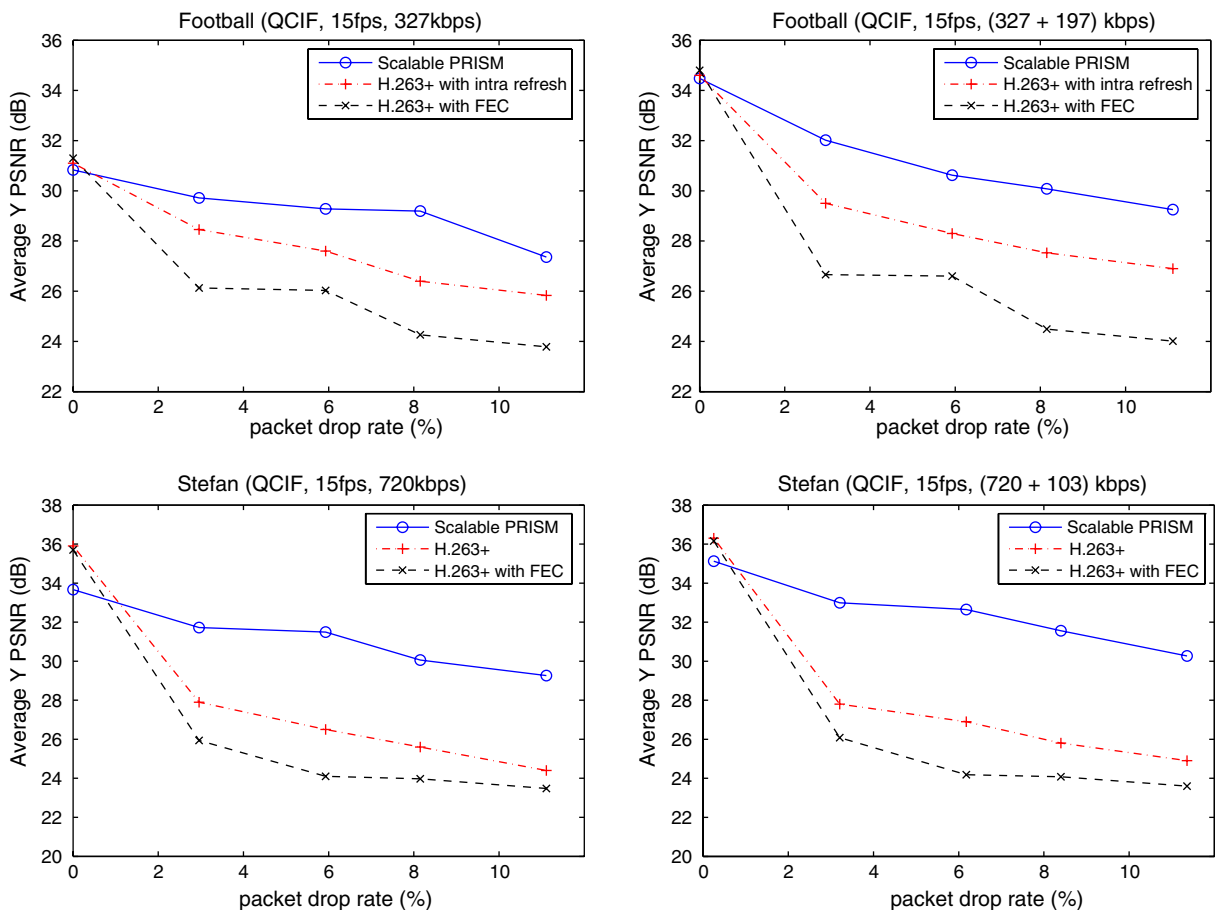


Fig. 9. Performance comparison (for Multicast) of scalable PRISM, H.263+ protected with FECs (Reed–Solomon (RS) codes used, 20% of the total rate used for parity bits) and H.263+ protected with block-based intra-refresh (15% of the blocks are forced to be intra-coded) for the *Football* and *Stefan* sequences. For the FEC case, protection was given only to the base layer.

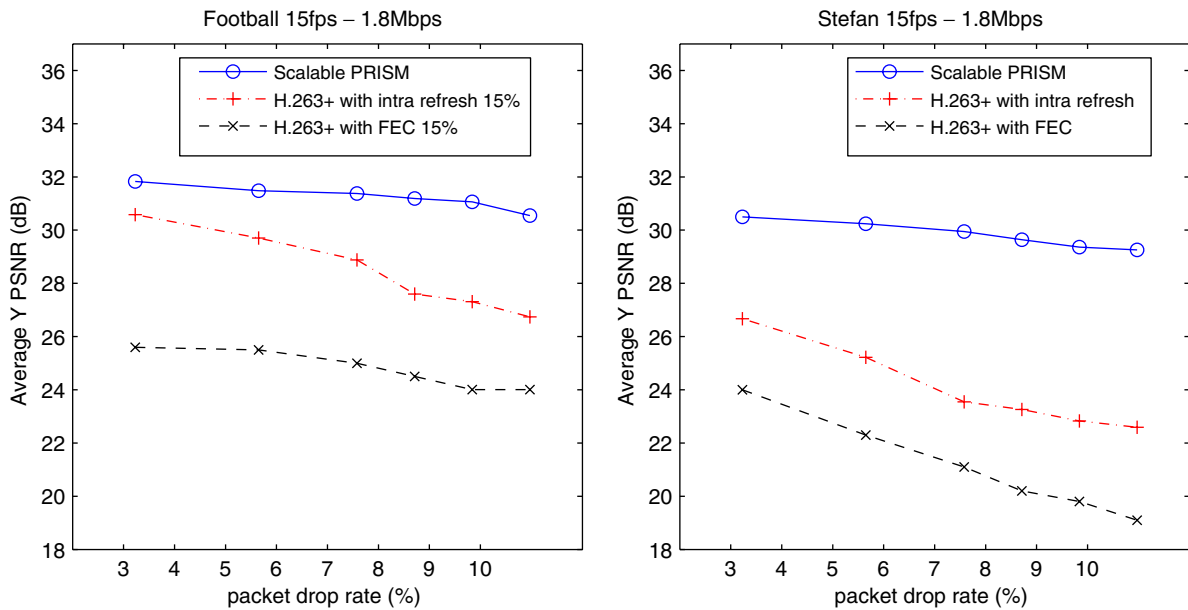


Fig. 10. Performance comparison of proposed scalable solution, H.263+ protected with FECs (Reed–Solomon (RS) codes used, 20% of the total rate used for parity bits) and H.263+ protected with block-based intra-refresh (15% of the blocks are forced to be intra-coded) for the *Football* and *Stefan* sequences (CIF, 15 fps, 1800 kbps).

decoder) and compare it to the SNR scalable version of the H.263+ video coder⁶ protected with FEC and block-based intra-refresh.

For the case of scalable H.263+ protected with FEC, we use Reed–Solomon (RS) codes with 20% of the total rate allocated to parity bits.⁷ No unequal error protection scheme is applied in our simulations, and it is assumed that the same packet loss rate affects both the base layer and the enhancement layer. For the case of block-based intra-refresh, approximately 15% of the blocks are forced to be intra-coded. In this experiment we used H.263+ as a benchmark instead of the state-of-the-art H.264/AVC codec as the former has built in support for SNR scalability. For our tests, we restrict ourselves to the case when the entire rate R can be utilized by the lower rate client (decoder 1 in Fig. 5).

For the case of SNR scalable PRISM, the baseline version of PRISM as described in Section 3 is used at the base layer, whereas the algorithm

described in Section 4.2 is employed at the enhancement layer.

We tested our scheme using a wireless channel simulator.⁸ This simulator adds packet errors to multimedia data streams transmitted over wireless networks conforming to the CDMA2000 1X standard [29].⁹ For each SNR layer, a frame is divided into horizontal slices (four or 16 slices at QCIF/CIF resolution, respectively) and each slice is sent as a packet. We assume here that either a packet is received or it is completely lost. In the latter case we use a simple error concealment technique by pasting the co-located blocks taken from the reference frame.

Fig. 9 shows the performance comparison for the *Football* (QCIF, 15 fps) and *Stefan* (QCIF, 15 fps) sequences. As can be seen from Fig. 9, the scalable PRISM codec is superior to scalable H.263+ as well as scalable H.263+ protected with FECs by a very wide margin (5–8 dB). Although assigning 20% of rate to FECs seems to overprotect the video stream, given the largest packet drop rate being equal to 10%, this is not the case under two important testing conditions: (a) FECs are computed across

⁶Free Version of H.263+ obtained from University of British Columbia.

⁷Evolving standards for video broadcast over cellular networks (such as 3GPP) typically allocate extra rate of about 20% for FECs and/or other error correcting mechanisms.

⁸Courtesy of Qualcomm, Inc.

⁹The packet error rates are determined by computing the carrier to interference ratio of the cellular system.

one frame at the time, in order to avoid delay; (b) packet loss patterns observed in the tested network configuration are not random, as large bursts of errors occur in practice. This explains why the performance of H.263+ protected with FEC drops even at low packet loss rates.

5.2. Spatial and temporal scalability tests

For tests on spatial and temporal scalability, the base layer was coded using PRISM and the spatial and temporal enhancement layers are encoded as described in Sections 4.3 and 4.4, respectively. The proposed system was compared at full spatial resolution against the H.263+ video codec under two testing conditions: (a) protected with FECs with 20% of the total rate used for parity bits (RS codes were used); (b) protected with intra-refresh blocks, with approximately 15% of the blocks being forced to be intra-coded. As in Section 5.1, we tested these schemes using the wireless channel simulator conforming to the CDMA2000 1X standard. We assumed that packet losses hit the base and the enhancement layer with the same probability.

Figs. 10 and 11 show the performance comparison for the *Stefan* sequence at 15 fps and *Football*

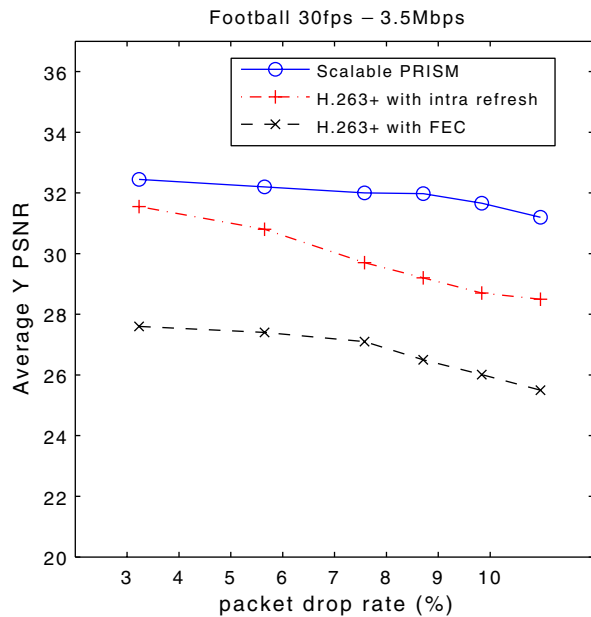


Fig. 11. Performance comparison of proposed scalable solution, H.263+ protected with FECs (Reed–Solomon (RS) codes used, 20% of the total rate used for parity bits) and H.263+ protected with block-based intra-refresh (15% of the blocks are forced to be intra-coded) for the *Football* sequence (CIF, 30 fps, 3500 kbps).

sequence at 15 and 30 fps. The scalable PRISM implementation clearly out-performs H.263+ in both configurations (protected with FECs and intra-refresh) by a wide margin (up to 6 and 4 dB, respectively, at high packet loss rates for *Football*). Fig. 12 shows the reconstruction of a particular frame (the middle frame of the GOP) of the *Stefan* sequence by the proposed scalable PRISM coder and H.263+. As can be seen from Fig. 12 the visual quality provided by the scalable PRISM coder is clearly superior to that provided by H.263+. As can be seen from Figs. 10 and 12, the scalable PRISM coder is able to provide good quality reconstruction even when parts of the base layer is lost. This is in



Fig. 12. Comparison of Frame 8 of the *Stefan* sequence (15 fps, 1800 kbps) reconstructed by the proposed solution and H.263+ at a channel error rate equal to 8%. (a) Proposed codec: base layer only (QCIF). (b) Proposed codec: base layer and enhancement layer (CIF). (c) H.263+ (CIF).

marked contrast to standard (prediction-based) scalable video coders where loss of the base layer often severely affects the video quality.

6. Conclusions

We proposed a fully scalable coding scheme based on distributed source coding targeting wireless video multicast applications. Experimental results showcase the robustness features of the proposed approach, showing significant objective and subjective gains with respect to predictive coders like H.263+. Currently, we are in the process of making the codec to work efficiently at lower encoding rates and running extensive tests over different types of channels to further validate our approach.

References

- [1] A. Majumdar, K. Ramchandran, Video multicast over lossy channels based on distributed source coding, in: Proceedings of the International Conference on Image Processing, Singapore, October 2004.
- [2] M. Tagliasacchi, A. Majumdar, K. Ramchandran, A distributed-source-coding based robust spatio-temporal scalable video codec, in: Picture Coding Symposium, San Francisco, CA, December, 2004.
- [3] Requirements and applications for scalable video coding v.5, ISO/IEC JTC1/SC29/WG11 MPEG Document N6505, July 2004.
- [4] ITU-T, Information Technology Coding of Audio-visual Objects-Part 10: Advanced Video Coding, May 2003, ISO/IEC International Standard 14496-10:2003.
- [5] MPEG-4 video, proposed draft amendment (PDAM), ISO/IEC FGS v. 4.0 14 496-2, March 2000.
- [6] ITU-T, Video coding for low bitrate communication, January 1998, ITU-T Recommendation H.263, Version 2.
- [7] R. Puri, K. Ramchandran, PRISM: a new robust video coding architecture based on distributed compression principles, in: Allerton Conference on Communication, Control and Computing, Urbana-Champaign, IL, October 2002.
- [8] R. Puri, K. Ramchandran, PRISM: A video coding architecture based on distributed compression principles, Technical Report no. UCB/ERL M03/6, ERL, UC Berkeley, March 2003.
- [9] A. Majumdar, J. Chou, K. Ramchandran, Robust distributed video compression based on multilevel coset codes, in: Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, November 2003.
- [10] A.D. Wyner, J. Ziv, The rate distortion function for source coding with side information at the decoder, *IEEE Trans. Inform. Theory* 22 (January 1976) 1–10.
- [11] A. Sehgal, N. Ahuja, Robust predictive coding and the Wyner–Ziv problem, in: Proceedings of the IEEE Data Compression Conference, Snowbird, UT, October 2003.
- [12] B. Girod, A. Aaron, S. Rane, D.R. Monedero, Distributed video coding, *Proc. IEEE* 93 (January 2005) 71–83.
- [13] H. Schwarz, D. Marpe, T. Wiegand, Snr-scalable extension of H.264/AVC, in: Proceedings of the International Conference on Image Processing, Singapore, October 2004.
- [14] Scalable video model version 3.0, ISO/IEC JTC1/WG11 Doc. N6716, November 2004.
- [15] Q. Xu, Z. Xiong, Layered Wyner–Ziv video coding, in: Visual Communications and Image Processing, Proceedings of SPIE, San Jose, CA, January 2004.
- [16] H. Wang, A. Ortega, WZS: Wyner–Ziv scalable predictive video coding, in: Picture Coding Symposium, San Francisco, CA, December 2004.
- [17] A. Sehgal, A. Jagmohan, N. Ahuja, Scalable video coding using Wyner–Ziv codes, in: Picture Coding Symposium, San Francisco, CA, December 2004.
- [18] A. Majumdar, R. Puri, P. Ishwar, K. Ramchandran, Complexity/performance trade-offs for robust distributed video coding, Genova, Italy, 2004.
- [19] J.D. Slepian, J.K. Wolf, Noiseless coding of correlated information sources, *IEEE Trans. Inform. Theory* 19 (July 1973) 471–480.
- [20] S.S. Pradhan, J. Chou, K. Ramchandran, Duality between source coding and channel coding and its extension to the side information case, *IEEE Trans. Inform. Theory* 49 (May 2003).
- [21] S.S. Pradhan, K. Ramchandran, Distributed source coding using syndromes (DISCUS): design and construction, *IEEE Trans. Inform. Theory*, March 2003.
- [22] P. Ishwar, V.M. Prabhakaran, K. Ramchandran, Towards a theory for video coding using distributed compression principles, in: Proceedings of the International Conference on Image Processing, Barcelona, Spain, September 2003.
- [23] A. Aaron, R. Zhang, B. Girod, Wyner–Ziv coding of motion video, in: Proceedings of the 36th Asilomar Conference on Signals, Systems, and Computers, vol. 1, Pacific Grove, CA, October 2002, pp. 240–244.
- [24] J.K. Wolf, Efficient maximum likelihood decoding of linear block codes using a trellis, *IEEE Transactions on Information Theory* 24 (1) (January 1978) 76–80.
- [25] M.P.C. Fossorier, S. Lin, Soft decision decoding of linear block codes based on order statistics, *IEEE Transactions on Information Theory* 41 (5) (September 1995) 1379–1396.
- [26] ISO/IEC 13818-2, Information technology—generic coding of moving pictures and associated audio information: video, 1995, MPEG-2 video coding standard.
- [27] C. Heegard, T. Berger, Rate distortion when side information may be absent, *IEEE Trans. Inform. Theory* 31 (November 1985) 727–734.
- [28] Y. Steinberg, N. Merhav, On successive refinement of the Wyner–Ziv problem, *IEEE Trans. Inform. Theory* 50 (8) (August 2004) 1636–1654.
- [29] TIA/EIA, interim standard for CDMA2000 spread spectrum systems, May 2002.