

# COMBINING MCTF WITH DISTRIBUTED SOURCE CODING

*Marco Tagliasacchi, Stefano Tubaro, Augusto Sarti*

Politecnico di Milano  
Dipartimento di Elettronica e Informazione

## ABSTRACT

Motion Compensated Temporal Filtering (MCTF) has proved to be an efficient coding tool in the design of open-loop scalable video codecs. In this paper we propose a MCTF video coding scheme based on lifting where the prediction step is implemented using PRISM (Power efficient, Robust, hIgh compression Syndrome-based Multimedia coding), a video coding framework built on distributed source coding principles. We study the effect of integrating the update step at the encoder or at the decoder side. We show that the latter approach allows to improve the quality of the side information exploited during decoding. We present the analytical results obtained by modeling the video signal along the motion trajectories as an AR(1) process showing that the update step at the decoder allows to half the contribution of the quantization noise. We also include experimental results with real video data that demonstrate the potential of this approach when the video sequences are coded at low bitrates.

## 1. INTRODUCTION

Robust scalable video coding has become an important problem in light of recent proliferation of multimedia applications over wireless and heterogeneous networks. In fact the wireless medium requires robustness to channel losses. At the same time scalability is important in many applications like multicast, surveillance and browsing. Providing a scalable video stream together with superior resilience to packet losses can be useful for example in broadcasting a set of heterogeneous mobile receivers having varying computational and display capabilities.

Motion Compensated Temporal Filtering (MCTF) [2] has been widely used in the design of scalable video codecs. Both DCT [10] and wavelet based [11] video coding architectures recently considered by the MPEG AdHoc group on Scalable Video Coding adopt MCTF in order to reduce temporal redundancy. In this paper we show how we can combine MCTF with PRISM [3, 4], a video coding framework that builds on distributed video coding principles. The PRISM codec is inherently robust to losses in the bit-stream and significantly outperforms standard predictive video coders for transmission over channels characterized by a significant packet loss probability. While the PRISM framework allows for a flexible distribution of the motion search task between the encoder and the decoder, in this work we will focus on the case when most of the motion estimation task is performed at the decoder. This is of particular relevance to the emerging "uplink" multimedia applications (such as users streaming from their cellphones). In our earlier work [14] we considered robust spatial and temporal scalability

based on the PRISM framework whereas [9] discusses SNR scalability. Recently, video codecs based on distributed source coding with scalability properties have been proposed in [8, 12, 13]. However these codecs target SNR scalability. On the other hand this paper addresses one specific aspect related to the robust delivery of scalable video content since we focus on the integration of PRISM into a MCTF scheme. More specifically we study how to take advantage of the update step in a distributed source coding based framework showing the benefits of moving this task at the decoder side.

## 2. BACKGROUND ON PRISM

The PRISM video coder is based on a modification to the 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 [6] for details). The Wyner-Ziv Theorem [1] deals with the problem of source coding with side-information. The encoder needs to compress a source  $X$  when the decoder has access to a source  $Y$ .  $X$  and  $Y$  are correlated sources and  $Y$  is available only at the decoder. From information theory we know that for the MSE distortion measure and  $X = Y + N$  where  $N$  has a Gaussian distribution, the rate - distortion performance for coding  $X$  is the same whether or not the encoder has access to  $Y$  [1]. 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 noise between the source and the side-information. For the PRISM video coder [3, 4], the video frame to be encoded is first divided into non-overlapping spatial blocks of size  $8 \times 8$ . The source  $\mathbf{X}$  is the current block to be encoded. The side-information  $\mathbf{Y}$  is the best (motion-compensated) predictor for  $\mathbf{X}$  in the previous frame and let  $\mathbf{X} = \mathbf{Y} + \mathbf{N}$ . We first encode  $\mathbf{X}$  in the intra-coding mode to come up with the quantized codeword for  $\mathbf{X}$ . Now, we do the syndrome encoding, i.e., we find a channel code that is matched to the noise  $\mathbf{N}$ , and use that to partition the source codebook into cosets of that 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  $\mathbf{X}$ ) and a CRC check of the quantized block. In contrast to MPEG, H.26x, etc., it is the decoder's task to do motion search, as it searches over the space of candidate predictors one-by-one to

---

The authors wish to acknowledge the support provided by the European Network of Excellence VISNET (<http://www.visnetnoe.org>)

decode a block from the set labeled by the syndrome. When the decoded block matches the CRC check, decoding is declared to be successful. For further details please refer to [4].

**Robustness Characteristics of PRISM** : The key aspect here is that PRISM does not use the exact realization of frame  $(N - 1)$  while encoding blocks in frame  $N$ , but only the correlation statistics. Note that if the decoder does not have frame  $(N - 1)$  (or a part of it) due to channel loss, it might still have blocks from frame  $(N - 2)$ . If the correlation noise between any of these blocks and the current block is within the noise margin for which the syndrome code was designed the current block can be decoded. Informally speaking, PRISM sends specific bit-planes of the current block, unlike predictive coders which send information about the difference between the block and its predictor. Consequently, in the PRISM framework, every time a block can be decoded, it has the same effect as an “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.

### 3. BACKGROUND ON MCTF

Motion Compensated Temporal Filtering (MCTF) iteratively decomposes the input sequence into a set of temporal subbands. If there is temporal correlation most of the energy will be concentrated into the low-pass temporal subbands. MCTF is usually performed taking advantage of the lifting scheme. This technique enables to split direct wavelet temporal filtering into a sequence of prediction and update steps in such a way that the process is both perfectly invertible and computationally efficient. In the simple example of Haar filters, the input frames are recursively processed two-by-two, according to the following formulas:

$$h_t = \frac{1}{\sqrt{2}}[x_t - W_{x_{t-1} \rightarrow x_t}(x_{t-1})] \quad (1)$$

$$l_t = \sqrt{2}x_{t-1} + W_{x_t \rightarrow x_{t-1}}(h_t) \quad (2)$$

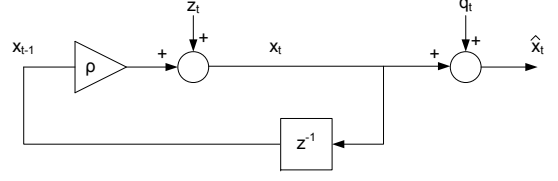
Where  $W_{x_{t-1} \rightarrow x_t}(\cdot)$  is a motion warping operators that warps frame  $x_{t-1}$  into the coordinate system of frame  $x_t$ . These two lifting steps are then iterated on the low-pass subbands  $l_t$  such that for each GOP we end up with only one low-pass subband. The prediction step is the counterpart of motion compensated prediction in conventional closed loop schemes. The energy of frame  $h_t$  is lower than the original frame, thus achieving compression. On the other hand the update step can be thought as a motion compensated averaging along the motion trajectories: the updated frames are free from temporal aliasing artifacts and at the same time  $l_t$  requires fewer bits for the same quality than frame  $x_{t-1}$  because of the motion compensated denoising performed by the update step.

### 4. VIDEO SIGNAL MODEL

In the rest of the paper we model the video source  $x_t(\underline{n})$  as a autoregressive process of order 1 along the motion trajectories. For the sake of simplicity we will consider a 1D signal as the evolution of the sequence along time:

$$x_t = \rho x_{t-1} + z_t, \quad E[x_i z_j] = 0 \quad \forall i < j \quad (3)$$

We recall that this process is completely identified by the correlation coefficient  $\rho$  and the source power  $\sigma_x^2$  since  $\sigma_z^2 = (1 - \rho^2) \sigma_x^2$ .



**Fig. 1.** The input sequence is modelled as an AR(1) process along the motion trajectories.  $\rho$  and  $z_t$  account for the ‘correlation noise’ whereas the quantization noise  $q_t$  is added out of the loop

With respect to what stated in Section 2,  $\sigma_z^2$  represents the “correlation noise” term. PRISM encoding can be thought as being performed in intra mode but at a rate that approaches inter mode. For this reason the quantization noise can be modelled as added out of the loop:

$$\hat{x}_t = x_t + q_t = \rho x_{t-1} + z_t + q_t \quad (4)$$

Figure 1 gives a pictorial representation of the signal model we are assuming in this paper.

### 5. MCTF BASED ON PRISM

The key idea is to combine PRISM encoding with MCTF. The first step consists of replacing the lifting prediction step with PRISM encoding. In other words, instead of computing and encoding frame  $h_t$  we encode frame  $x_t$  directly using PRISM. By doing this we are performing intra coding of  $x_t$  at a rate that approaches the bits that would be spent for  $h_t$  (i.e. inter mode encoding). In the following we analyze three coding schemes that can be adopted, according to whether and where we perform the update step:

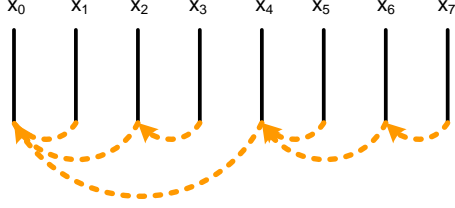
- update step skipped
- update step at the encoder side
- update step at the decoder side

#### 5.1. Update step skipped

If the update step is skipped, the integration of PRISM into the MCTF scheme is rather trivial, since we are simply revisiting the encoding order. As shown in Figure 2 for a GOP size equal to 8, we start encoding frame  $x_0$  in intra mode. Then we encode frame  $x_4$  using  $x_0$  as side information, frames  $x_2$  and  $x_6$  using frame  $x_0$  and  $x_4$  respectively as side information, etc. As PRISM needs to estimate at the encoder side the correlation noise between the encoded block and the best predictor that will be observed at the decoder, we collect off-line different statistics based on the temporal distance of the frame to be encoded and its side information.

#### 5.2. Update step at the encoder side

The PRISM framework allocates bits at the encoder based on the estimate of the noise that the decoder will observe between the block to be decoded and its side information, i.e. the best motion compensated predictor. We might use the update step as a way of reducing such a noise, therefore reducing the allocated rate. A straightforward implementation consists of performing the complete MCTF decomposition iterating prediction and update then



**Fig. 2.** Prediction step is replaced by PRISM encoding. The arrows point from the frame to be encoded to the frame used as side information

encoding  $x_t$  using  $l_t$  instead of  $x_{t-1}$  as side information. The reason for doing this is that  $x_t$  is obviously more correlated to  $l_t$  than to  $x_{t-1}$ , as the former can be rewritten as a temporal average between  $x_{t-1}$  and  $x_t$ . If we restrict our analysis at the first temporal decomposition level, we need to send both  $x_t$  (encoded with PRISM) and  $l_t$  (intra coded). Although this approach seems to be promising at first, there are two main drawbacks that make it unpractical:

- the update step make sense only when we are able to compute a good motion model. It can be shown that when the motion model fails the update step introduces ghosting artifacts in the low-pass frames and reduces the coding efficiency. As we are assuming low-complexity encoding, we have access to a very coarse motion model at the encoder, whereas the motion search task is carried out at the decoder side
- the temporal transform we are implicitly computing is not orthogonal. For this reason the quantization noise is spread unevenly in the reconstructed frames, causing large PSNR and subjective quality fluctuations. Let us assume that the decoder reconstructs correctly  $\hat{l}_t$  and  $\hat{x}_t$ , the quantized copies of  $l_t$  and  $x_t$  respectively. The update step has to be inverted in order to reconstruct  $\hat{x}_{t-1}$ :

$$\hat{x}_{t-1} = \sqrt{2}\hat{l}_t - \hat{x}_t \quad (5)$$

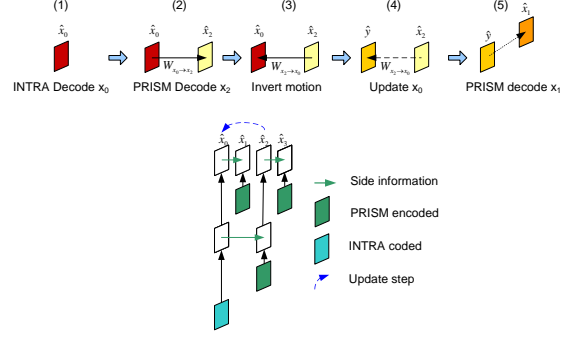
If we assume that the quantization noise in  $\hat{l}_t$  and  $\hat{x}_t$  are uncorrelated, the quantization noise in  $\hat{x}_{t-1}$  turns out to be:

$$\sigma_q^2(\hat{x}_{t-1}) = 2\sigma_q^2(\hat{l}_t) + \sigma_q^2(\hat{x}_t) > \sigma_q^2(\hat{x}_t) \quad (6)$$

From the previous formula we can conclude that the decoded sequence exhibits large quality fluctuations. In order to reduce this annoying effect we should reduce  $\sigma_q^2(l_t)$  but this can be done only by increasing the bit budget allocated to  $l_t$ . Furthermore these fluctuations tend to be amplified as we iterate MCTF on the low-pass frame output of the first temporal decomposition. We can conclude that the effect of performing the update step in this scenario has the same consequences as skipping the update step in a conventional lifting scheme, since in this case also we are forcing the temporal transform to being not orthogonal.

### 5.3. Update step at the decoder side

In order to overcome the limitations addressed in the previous section, we moved the update step at the decoder side. The idea here is two-fold:



**Fig. 3.** Decoder side update. Frame  $x_0$  is updated along the motion trajectories with frame  $x_2$  before being used as side information for decoding  $x_1$

- in a low-complexity encoder scenario, it makes sense to perform the update step at the decoder, since we have access to a better motion model.
- in a distributed source coding based scheme we are encouraged to do whatever we can in order to improve the quality side information or, in other words, to increase the correlation between the block to be decoded and the side information that is available at the decoder

In light of these statements, the decoding process proceeds as follows (see Figure 3):

- intra decode frame  $\hat{x}_0$
- PRISM decode frame  $\hat{x}_2$  using  $\hat{x}_0$  as side information. During decoding the motion model  $W_{x_0 \rightarrow x_2}(\cdot)$  is obtained
- Invert and refine the motion model computing  $W_{x_2 \rightarrow x_0}(\cdot)$
- create the new side information by updating frame  $x_0$ :

$$\hat{y}_0 = \frac{\hat{x}_0 + W_{x_2 \rightarrow x_0}(\hat{x}_2)}{2} \quad (7)$$

The update step carried out as the last step in the previous process allows to denoise the side information before being used. Note that we are encouraged to use non-linear operators in order to reduce ghosting artifacts introduced by the update step when the motion model is not reliable as suggested in [7]. We can prove using the video signal model introduced in Section 4 that the update step at the decoder reduces by half the contribution of the quantization noise. In fact:

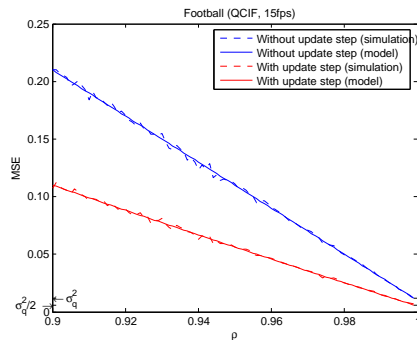
$$E[(x_t - \hat{x}_{t-1})^2] = 2(1 - \rho)\sigma_x^2 + \sigma_q^2 \quad (8)$$

$$E[(x_t - \hat{y}_{t-1})^2] = \frac{1}{2}(1 - \rho)(3 - \rho)\sigma_x^2 + \frac{\sigma_q^2}{2} \quad (9)$$

where:

$$\hat{y}_{t-1} = \frac{\hat{x}_{t-1} + W_{x_{t+1} \rightarrow x_{t-1}}(\hat{x}_{t+1})}{2} \quad (10)$$

In order to derive equations (8) and (9) in close form we neglected the contribution of some cross terms between the quantization and the correlation noise that are not strictly zero. Nevertheless we carried out some simulations in order to validate this simplification. We concluded that for signal-to-noise ratios greater than 15dB the the accuracy of the model is not affected. Figure 4 shows both the analytical expressions and the empirical values obtained in our



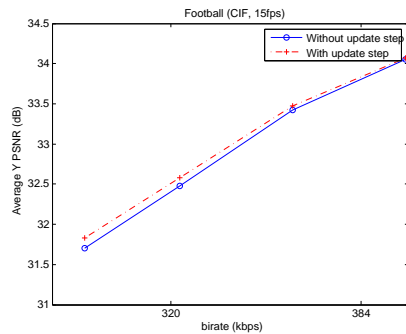
**Fig. 4.** The solid lines represent the right hand side of equations (8) and (9), computed in closed form neglecting some cross terms involving  $z_t$  and  $q_t$ . Dashed lines are the simulations obtained computing directly the left hand side of (8) and (9) on sample data

simulations of (8) and (9) for different values of  $\rho$  and SNR equal to +20dB. We can observe that as  $\rho$  tends to one, the first term in equations (8) and (9) goes to zero and the quantization noise tend to dominate. By performing the update step at the decoder the quantization noise term is halved. The encoder can take advantage of this observation when allocating the bit budget, reducing the rate while achieving the same distortion and keeping the probability of decoding error unchanged.

## 6. EXPERIMENTAL RESULTS

Based on the analytical results obtained in the previous section, we carried out some simulations on real video sequences. We implemented a MCTF temporal pyramid where the prediction step is replaced by PRISM encoding. We set the GOP size equal to 16. In the current implementation we adopt a closed GOP structure for the update step. For this reason the temporal subbands at the GOP boundaries cannot take advantage of the update step (i.e. frame  $x_3$  in Figure 3). Only 12 out of 15 frames have access to an improved side information. Moreover, only a subset of the blocks in each frame is encoded in PRISM mode. In addition to this, the update step is adaptively disabled on a block-by-block basis when the encoder estimates a weak correlation with the side information available at the decoder. In fact when the motion model fails the update step is disabled in order to avoid ghosting artifacts. Figure 5 shows the PSNR as a function of the bitrate for the *Football* sequence. It can be noticed a slight PSNR improvement when the update step is used at the decoder. Note that the gain is higher at lower bitrates, because the effect of quantization noise tends to be more relevant, as suggested by our analytical model.

The proposed codec inherits the robustness features of PRISM. Although not reported in this paper for lack of space, experiments performed simulating a noisy channel with packet losses demonstrate that a gain of up to +6dB can be achieved with respect to conventional predictive codecs such as H.264/AVC or H.263+, even when forward error correcting codes (FEC) are used. The reader is referred to [14] for further results on this matter.



**Fig. 5.** Football - QCIF@15fps

## 7. CONCLUSIONS

We presented a MCTF based coding scheme based on PRISM, a video coding framework built on distributed source coding principles. We focused on the integration of the update step showing that benefits can be obtained when this task is moved at the decoder side. We are currently working on extending MCTF to longer filters than Haar (i.e. 5/3 filters) using our distributed source coding scheme.

## 8. REFERENCES

- [1] A. D. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, pp. 1–10, Jan 1976.
- [2] J.-R. Ohm "3-D Subband Coding with Motion Compensation," *IEEE Trans. Image Process.*, vol. 3, pp. 559-571, Sept. 1994
- [3] R. Puri and K. Ramchandran "PRISM: A New Robust Video Coding Architecture based on Distributed Compression Principles," *Allerton Conference on Communication, Control and Computing*, 2002.
- [4] R. Puri and K. Ramchandran "A video coding architecture based on distributed compression principles," in *ERL Technical Report, Memorandum No. UCB/ERL M03/6*, Mar 2003.
- [5] A. Majumdar, J. Chou, K. Ramchandran, "Robust Distributed Compression Based on Multilevel Coset Codes," *Asilomar 2003*
- [6] P. Ishwar, V. M. Prabhakaran, and K. Ramchandran, "Towards a Theory for Video Coding Using Distributed Compression Principles," *ICIP 2003*, Barcelona, Spain
- [7] N. Mehrseresht, D. Taubman, "Adaptively weighted update steps in motion compensated lifting based on scalable video compression," *ICIP 2003*, Barcelona, Spain
- [8] Q. Xu, Z. Xiong "Layered Wyner-Ziv video Coding," *VCIP 2004*, San Jose, USA
- [9] A. Majumdar, K. Ramchandran, "Video Multicast over Lossy Channels based on Distributed Source Coding," *ICIP 2004*, Singapore
- [10] H. Schwarz, D. Marpe, T. Wiegand "SNR-Scalable Extension of H.264/AVC," *ICIP 2004*, Singapore
- [11] N. Mehrseresht, D. Taubman "An Efficient Content-Adaptive MC 2D-DWT with Enhanced Spatial and Temporal Scalability," *ICIP 2004*, Singapore
- [12] H. Wang and A. Ortega "WZS: Wyner-Ziv Scalable Predictive Video Coding," *PCS 2004*, San Francisco, USA
- [13] A. Sehgal, A. Jagmohan and N. Ahuja "Scalable Video Coding Using Wyner-Ziv Codes," *PCS 2004*, San Francisco, USA
- [14] M. Tagliasacchi, A. Majumdar, K. Ramchandran "A Distributed-Source-Coding Based Robust Spatio-Temporal Scalable Video Codec," *PCS 2004*, San Francisco, USA