Estimation-Theoretic Delayed Decoding of Predictively Encoded Video Sequences

Jingning Han, Vinay Melkote, and Kenneth Rose*

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
{jingning,melkote,rose}@ece.ucsb.edu

Abstract

Current video coding schemes employ motion compensation to exploit the fact that the signal forms an auto-regressive process along the motion trajectory, and remove temporal redundancies with prior reconstructed samples via prediction. However, the decoder may, in principle, also exploit correlations with received encoding information of future frames. In contrast to current decoders that reconstruct every block *immediately* as the corresponding quantization indices are available, we propose an estimation-theoretic delayed decoding scheme which leverages quantization and motion information of one or more future frames to refine the reconstruction of the current block. The scheme, implemented in the transform domain, efficiently combines all available (including future) information in an appropriately derived conditional pdf, to obtain the optimal *delayed* reconstruction of each transform coefficient in the frame. Experiments demonstrate substantial gains over the standard H.264 decoder. The scheme learns the autoregressive model from information available to the decoder, and compatibility with the standard syntax and existing encoders is retained.

1 Introduction

Differential pulse code modulation (DPCM) in the form of motion-compensated coding is widely employed in modern video coders [1]. The underlying assumption is that blocks of the video signal along a motion trajectory form an auto-regressive (AR) source. While the emphasis of this paper is on video compression, for now, consider a generic first order AR source $\{x_n\}$, a stationary sequence of zero-mean, real-valued random variables with,

$$x_n = \rho x_{n-1} + z_n \ . \tag{1}$$

*This work was supported in part by the University of California MICRO Program, Applied Signal Technology Inc., Qualcomm Inc., and Sony Ericsson Inc.

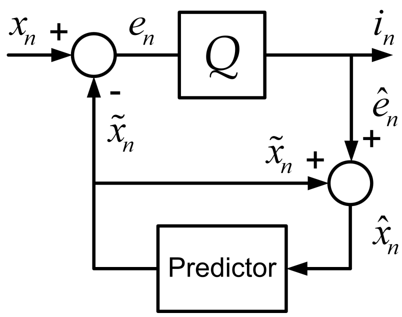

Figure 1: DPCM encoder

The random variables $\{z_n\}$ are independent and identically distributed (i.i.d), with specific probability density function (pdf) $p_Z(z)$, and are the driving innovation process. The correlation coefficient of adjacent samples is ρ. A DPCM encoder (Fig. 1) generates a prediction $\tilde{x}_n$, based on prior reconstructions, and subtracts it from the current sample x_n to generate the prediction error e_n. This is quantized to produce an index i_n which is entropy coded and sent to the decoder. The reconstruction of x_n is $\hat{x}_n = \tilde{x}_n + \hat{e}_n$, where $\hat{e}_n$ is the reconstruction characteristic of the quantizer cell with index i_n. At high rates, $\hat{x}_{n-1} \approx x_{n-1}$, and the optimal predictor is $\tilde{x}_n = \rho\hat{x}_{n-1}$. This form of the predictor is often employed at low bit-rates as well. In the AR source model, x_n is correlated not just with samples from the past, but also with the future, i.e., with $\{x_l\}_{l>n}$. At high rate, the prediction error $e_n \approx z_n \; \forall n$, hence $\{i_n\}$ are approximately i.i.d. In this case, future quantization indices $\{i_l\}_{l>n}$ provide no additional information on the current sample x_n. In practical, limited bit-rate scenarios, however, future indices do contain information about x_n, which could potentially be used to improve its reconstruction at the decoder. Naturally, doing so would entail decoding delay.

Consider now the case of a motion-compensated video compression scheme such as H.264 [1]. It supports variable block-size motion compensation with a minimum luminance partition of 4x4. Thus, inter-mode macroblocks consist of 4x4 blocks, each (potentially) individually motion-compensated from previously decoded frames. 2-D discrete cosine transform (DCT) is applied to the prediction residual and the resulting transform coefficients are quantized, entropy coded and transmitted. The decoder reconstructs these coefficients, applies an inverse DCT to reproduce the prediction error blocks, and adds them to the corresponding motion compensated prediction. Except for the transform, the similarity with DPCM is evident. Note that the decoder as described reconstructs each block immediately as the corresponding quantization indices are available (i.e., with *zero delay*). Whenever decoding delay is admissible, the motion vectors of future frames could potentially be used to continue the motion trajectory of every block in the current frame into subsequent frames, and the information available about these future blocks could then be exploited to improve the reconstruction of the current frame. In other words, similar to the DPCM case, delayed decoding could be employed to improve video reconstruction. This observation is the key premise of the proposed approach.

In the case of DPCM, decoder delay has been previously exploited in [2] and [3], both of which smooth (i.e., filter) the standard DPCM output, $\{\hat{x}_n\}$, with a suitable *non-causal* post-processor. More recently, in [4], an estimation-theoretic (ET), optimal delayed-decoder was proposed, which considerably outperformed the smoothing approaches of [2] and [3], and provided evidence for the substantial gains achievable by delayed-decoding. The ET scheme of [4] motivates the delayed-decoder for video signals proposed herein. We currently limit the decoding delay to one future frame. For each transform coefficient of a block in the current frame, we calculate its pdf conditioned on information on the corresponding subsequent block lying on the same motion trajectory, in addition to the usual available motion compensated information from past frames, and the current quantization index. The reconstruction of the transform coefficient is then obtained as the appropriate conditional expectation. The proposed delayed video decoder is implemented in the H.264/AVC framework, and is compatible with (i.e., requires no modification of) the standard syntax. In other words, the proposed decoder does not necessitate any re-encoding of existing compressed video sequences. The decoder learns the parameters/statistics needed to obtain the conditional pdf, from information available in the H.264 bit-stream. Experiments indicate that the proposed approach can provide PSNR improvements of about 1.7 dB relative to the the standard decoder, at very modest increase in complexity.

We note that, while end-to-end (or playback) latency in the form of bidirectional prediction (i.e., B-frames) [5] is already a feature of H.264, the proposed delayed decoding approach is of particular benefit to P-frames encountered very frequently in the bit-stream. A rate-distortion based scheme that employs encoder-end filtering along the motion trajectory has been described in [6] which, unlike the proposed approach here, incorporates *encoder delay* to exploit correlation with future frames. Other related prior work includes a closed-form quantitative characterization of the trade-off between performance and decoder-latency for generic Gauss-Markov sources [7], where temporal independence and gaussianity of the quantization noise were also assumed. It is noteworthy that the pdf of the innovations in case of motion compensated video is nearly laplacian [8].

This paper is organized as follows. Section 2 reviews the ET delayed decoding algorithm proposed in [4] for regular DPCM. Extension and adaption to delayed video decoding is presented in Section 3. Simulation results are provided in Section 4.

2 Estimation-Theoretic Delayed Decoding

The ET delayed decoding algorithm proposed in [4] is reviewed in this section, in light of the first order AR process (1) and the DPCM scheme described in Sec. 1. The distortion metric used is the mean squared error (MSE). Therefore, the optimal reconstruction of the sample x_n, given all the information (i.e., indices $\{i_l\}_{l\leq n+L}$) available at the decoder for a fixed delay or look-ahead L, is the minimum MSE estimate

$$\hat{x}_n^* = E[x_n|\{i_l\}_{l\leq n+L}] \ , \tag{2}$$

the expectation over the conditional pdf $p(x_n|\{i_l\}_{l\leq n+L})$. Thus, optimal reconstruction can be obtained once this pdf is derived. We use the streamlined notation $p(\cdot)$ to denote any pdf or probabilities, and add a subscript when the interpretation is not obvious from the context. We now note the following:

$$p(x_n|\{i_l\}_{l\leq n+L}) = \frac{p(x_n|\{i_l\}_{l\leq n})p(\{i_l\}_{n<l\leq n+L}|x_n)}{\int p(x_n|\{i_l\}_{l\leq n})p(\{i_l\}_{n<l\leq n+L}|x_n)dx_n} \ . \tag{3}$$

Unless otherwise indicated, integrals are over the real line. The above equation follows from Bayes' rule, and the Markov property of the process (1): given x_n, future events or indices $\{i_l\}_{n<l\leq n+L}$ are independent of any other information preceding x_n (i.e.,

$\{i_l\}_{l\le n})$. Note that $p(x_n|\{i_l\}_{l\le n})$ is the pdf of x_n conditioned on all information up to the current time n. The optimal *zero delay* estimate of x_n is simply the expectation over this pdf. The optimal *delayed* decoder though, weighs it with $p(\{i_l\}_{n<l\le n+L}|x_n)$, representing the probability given x_n of the known future outcomes, to obtain the composite pdf $p(x_n|\{i_l\}_{l\le n+L})$ in (3), that incorporates all known information up to the fixed delay L. The estimate of x_n is then $\hat{x}_n^*$ of (2). Two recursions were employed to obtain the requisite probabilities [4].

One recursion updates the zero-delay pdf $p(x_{n-1}|\{i_l\}_{l\le n-1})$ employed at time $n-1$, to the corresponding pdf $p(x_n|\{i_l\}_{l\le n})$ at time n. Specifically, the pdf of x_n conditioned on all *past* information, $\{i_l\}_{l\le n-1}$, is obtained by,

$$p(x_n|\{i_l\}_{l\le n-1}) = \int p(x_{n-1}|\{i_l\}_{l\le n-1})p_Z(x_n - \rho x_{n-1})dx_{n-1}$$

where the Markov property of the (1) is exploited. Note that the current index i_n (along with $\tilde{x}_n$) now provides the additional information that x_n lies in a particular interval $\mathcal{I}_n$, i.e., the effective quantization interval. Thus, the above pdf is restricted to $\mathcal{I}_n$ (and correspondingly re-normalized) to incorporate the information in i_n and obtain:

$$p(x_n|\{i_l\}_{l\le n}) = \begin{cases} \frac{p(x_n|\{i_l\}_{l\le n-1})}{\int_{\mathcal{I}_n} p(x_n|\{i_l\}_{l\le n-1})dx_n} & x_n \in \mathcal{I}_n \\ 0 & else \end{cases}.$$

A second recursion obtains the the probability $p(\{i_l\}_{n<l\le n+L}|x_n)$ of the L future events, as a function of x_n. Suppose we know the probability, $p(\{i_l\}_{n+m<l\le n+L}|x_{n+m})$, of future events $\{i_l\}_{n+m<l\le n+L}$ given x_{n+m}, where $m < L$. Then the Markov property of (1) can be used to show that,

$$p(\{i_l\}_{n+m-1<l\le n+L}|x_{n+m-1}) = \int_{\mathcal{I}_{n+m}} p(\{i_l\}_{n+m<l\le n+L}|x_{n+m})p_Z(x_{n+m}-\rho x_{n+m-1})dx_{n+m}.$$

Initializing $p(i_{n+L}|x_{n+L-1}) = \int_{I_{n+L}} p_Z(x_{n+L} - \rho x_{n+L-1})dx_{n+L}$, the above recursive equation can be applied $L-1$ times to obtain the requisite probability $p(\{i_l\}_{n<l\le n+L}|x_n)$ of the known future outcomes as a function of x_n. These recursions together with (3) and (2), provide $\hat{x}_n^*$. We note that in combining time correlations with quantization interval information, this approach was inspired by the method in [9] for optimal prediction in scalable coding.

3 Design of the Delayed Video Decoder

The above estimation-theoretic delayed decoding algorithm is next extended and applied, with suitable modifications, to delayed decoding of video sequences encoded by motion compensation. This section discusses the problems encountered in the video coding scenario and their solutions.

3.1 Motion Trajectory Construction

Delayed decoding of a block in the current frame requires the location, in neighboring reconstructed frames, of samples that lie on the same motion trajectory. The statistics

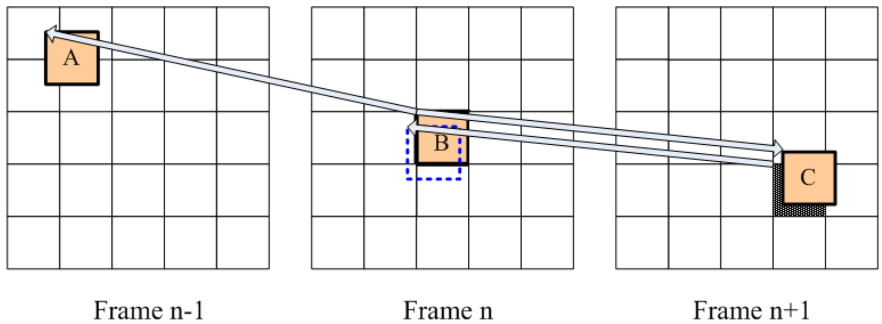

Figure 2: The blocks A, B and C form a sequence in the underlying AR process. Motion vectors available in the H.264 bit-stream are exploited to identify such sequences.

of the underlying AR process can then be exploited using the ET approach in [4]. In Fig. 2, block B is the block of interest in the current frame (indexed n). The available motion vector (in the H.264 bit-stream) indicates the corresponding block A, in the previous frame (indexed $n-1$), which is the predecessor of B in the AR process. We also require the location of block C in the future, i.e., frame $n+1$, which is the succeeding block of the process. Since delayed decoding is employed, the motion vectors of frame $n+1$ are available and can, in principle, be reversed to obtain the location of future blocks relevant to a current block of interest. However, there are complications. Note that motion vectors are available in the bit-stream for blocks that are seated on the grid dividing the frame into 4x4 sections. Thus, the motion vectors of frame $n+1$ map its *on-grid* blocks to corresponding (potentially *off-grid*) blocks of frame n. This is illustrated in Fig. 2. The dark-shaded block on the grid of frame $n+1$ is predicted by motion compensation from the block bounded by the broken lines in the current frame. But the proposed algorithm requires a mapping from on-grid blocks of frame n (for example, block B in Fig. 2), to corresponding (possibly off-grid) blocks of frame $n+1$ (i.e., similar to block C). To this end, we employ an approximation. We seek the on-grid block in the reconstructed frame $n+1$, whose motion compensation maximally overlaps the block of interest, B, in the current frame. The corresponding motion vector is reversed and, with reference to the position of block B, provides the location of the required block C in frame $n+1$. This process of reversing already available motion vectors of the next frame, and referencing them to on-grid blocks of the current frame, is a fast, low complexity alternative to a complete motion search to find a block in the reconstruction of the next frame that most resembles block B in the current frame. Some other methods to construct motion trajectory were also discussed in [6, 10].

3.2 Transform Domain vs Pixel Domain

The ET approach in [4] was developed for a 1-D source. But the blocks A, B, and C are 4x4 blocks of pixels. This issue is circumvented by implementing the ET decoder in the transform domain, i.e., a sequence of corresponding DCT coefficients in blocks along the motion trajectory is assumed to form a scalar AR process in time. Since the function of the transform is to exploit spatial correlation, the transform coefficients within the block are approximately uncorrelated, and estimation can thus be carried out separately for each (spatial) frequency. This effectively reduces the blocks to a set of 16 1-D AR sources. A second, more significant advantage is that the quantization information, i.e., indices or intervals, exploited by the ET decoder (see Sec. 2) is readily available in the transform domain. Such a transform domain approach has also been adopted in [9] for scalable video coding.

3.3 Statistical Model

It is assumed that the evolution of DCT coefficients along the motion trajectory follows the process (1), with x_n denoting the DCT coefficient of a specific "frequency" in a block of the current frame, and x_{n-1} denoting the corresponding DCT coefficient in the previous frame. The ET decoder requires the density $p_Z(z)$ of the innovations z_n to calculate the various conditional pdfs involved in the estimation algorithm. Since the motion-compensation is simply subtracted (without scaling) in standard video coders, we assume for simplicity that the correlation coefficient $\rho \approx 1$. With such a model it has been shown in prior work [8, 11] that the innovation density is well approximated by the zero-mean laplacian distribution, i.e.,

$$p_Z(z) = \frac{\lambda}{2} e^{-\lambda|z|} \ , \tag{4}$$

where λ is the laplacian parameter.

3.4 Required Modifications for Delayed Video Decoding

We are now ready to apply the ET delayed decoding algorithm in the H.264 decoder framework. We restrict our implementation to one frame latency case (i.e., $L = 1$ in Sec. 2). In the following, we describe the application of this algorithm to the AR process $\{x_n\}$ of DCT coefficients of one particular frequency.

It is evident from (3) that the optimal delayed decoder involves the density $p(x_n|\{i_l\}_{l\leq n})$ of the DCT coefficient in the current block, which is conditioned not only on quantization index i_n of the coefficient of interest, but also the indices $\{i_l\}_{l<n}$ corresponding to the same spatial frequency in all preceding blocks along the motion trajectory. Since the encoder transforms the prediction residuals of blocks on the grid, and quantizes and encodes these transform coefficients, the current index i_n is readily available from the bit-stream. But this is generally not the case for the history $\{i_l\}_{l<n}$. Let us say the x_n of interest belongs to the block B in frame n of Fig. 2. The preceding block A is not seated on the grid of frame $n-1$, and hence is not exactly the block that is transformed and encoded as part of the bit-stream for the frame. Therefore the corresponding index i_{n-1} is *not* available. This is in general the case with other past indices too. Therefore an approximation is needed. Specifically,

$$p(x_n|\{i_l\}_{l\leq n}) \approx p(x_n|\tilde{x}_n, i_n) \approx \begin{cases} \frac{p_Z(x_n-\tilde{x}_n)}{\int_{\mathcal{I}_n} p_Z(x_n-\tilde{x}_n)dx_n} & x_n \in \mathcal{I}_n \\ 0 & else \end{cases} \tag{5}$$

In the above, the prediction $\tilde{x}_n = \hat{x}_{n-1}$, the corresponding coefficient obtained by transforming the standard motion compensation for the block, is employed in lieu of the optimal combination (in $p(x_n|\{i_l\}_{l\leq n})$) of all past information. The fact that $\rho \approx 1$ is implicit. The interval $\mathcal{I}_n$ in which x_n lies is determined by $\tilde{x}_n$ and i_n.

The second requirement in (3) is the probability $p(i_{n+1}|x_n)$, i.e., the probability that the transform coefficient in the next frame (see block C of Fig. 2) is associated with the index i_{n+1}. Again, this block in the future frame may not be seated on the grid, and thus the index i_{n+1} need not be available in the bit-stream. This necessitates a second approximation. Note that the location of the block is already determined by the motion trajectory construction of Sec. 3.1, and the standard decoder provides a coarse estimate of the pixels in this block. 2-D DCT is now applied to this coarse reconstruction. Denote $\hat{x}_{n+1}$ as the DCT coefficient in this block, at the same frequency as x_n. Hypothetically, if the interval $\mathcal{I}_{n+1}$ in which the transform coefficient x_{n+1} lay was known, then $p(i_{n+1}|x_n) = \int_{\mathcal{I}_{n+1}} p_Z(x_{n+1} - \rho x_n) dx_n$. Since the index i_{n+1}, and hence this interval, are unknown, we approximate:

$$p(i_{n+1}|x_n) = p_Z(\hat{x}_{n+1} - x_n)\Delta \ . \tag{6}$$

This is interpreted as follows: the true value of x_{n+1} lies within an interval of length Δ around the coarse estimate $\hat{x}_{n+1}$, with its pdf, conditioned on x_n, nearly uniform on the interval. Note that this is indeed the case at high bit-rates. But as observed in the simulation results this approximation leads to good performance even at low and medium bit-rates.

We now apply (5) and (6) to (3), and subsequently obtain the delayed reconstruction $\hat{x}_n^*$ through (2). This procedure is applied to all 16 transform coefficients in the current block, and the inverse DCT is employed to obtain the delayed, pixel domain reconstruction.

3.5 Estimation of the Process Distribution

Each of the 16 transform coefficients (and hence the 16 1-D AR processes) of a 4x4 block is assumed to be characterized by a different value of the laplacian parameter λ, i.e., the laplacian innovation density has a frequency specific variance. The maximum-likelihood estimate of λ, given outcomes $z_0, \cdots, z_{N-1}$ of N independent draws of the random variable Z, is

$$\lambda_{ML} = \frac{N}{\sum_{i=0}^{N-1} |z_i|} \ . \tag{7}$$

Ideally, one would need to obtain the innovations at each frequency from the *original* video signal, and substitute in (7) to estimate the corresponding laplacian parameter. Such estimation would necessitate sending these parameter values as side-information in the bit-stream. This is avoided by instead estimating λ from information already available in the H.264 bit-stream. Specifically, the encoded bit stream contains the information needed to determine the reconstructed prediction error in the transform domain. At high bit-rates these reconstructed errors closely approximate the innovations of the transform domain AR processes. They are thus substituted for z_i in (7) to obtain the estimate of the laplacian parameters for each frequency, at the decoder itself.

In summary, the reconstruction of frame n by the proposed approach involves the following steps:

1. Decode as usual (e.g., by H.264) up to frame $n+1$: the standard reconstructions $\hat{x}_{n-1}$ and $\hat{x}_{n+1}$ are used in (5) and (6).
2. Construct the motion trajectory as described in Sec. 3.1 by employing motion vectors of frames n and $n+1$.
3. With the quantization information of frame n estimate the laplacian parameters, and obtain the interval $\mathcal{I}_n$ used in (5).
4. Apply the ET delayed decoder given by (5), (6), (3), and (2), to each transform coefficient of every block of frame n, and then inverse transform to obtain the spatial domain reconstruction.
5. Deblock the refined reconstruction of frame n.

It should be noted that laplacian density model for the innovation pdf (4) coupled with the approximations (5) and (6) enable closed form expressions (formulae) for the delayed reconstruction, i.e., the expectation (2). This considerably reduces the complexity in obtaining this estimate, which would otherwise require numerical computation of integrals.

4 Simulation Results

The proposed ET delayed decoder was implemented within the framework of H.264 Reference JM 16.0 [12] with frame-rate of 15 fps. The standard motion search, rate-control, and residual compression methods for inter mode prediction and coding were retained. Sub-pixel motion search was deactivated at this point to simplify the workings of the 'inverse' motion mapping described in Sec. 3.1. The standard deblocking function is employed.

The performance (in terms of PSNR) of the delayed decoder is compared with that of standard H.264 in Fig. 3. These results were obtained by decoding the first 30 frames of the sequence *coastguard_qcif.yuv* when encoded in IPPP format. Note that H.264 uses a uniform quantizer whose reconstruction, in every quantizer cell, is the midpoint of the interval. Since the innovations (and the prediction error) are nearly laplacian, even the optimal zero-delay reconstruction is in fact shifted from the midpoint towards the origin. Thus, even with no decoding delay the ET approach offers benefits: the previously described estimation of laplacian parameters (λs) by itself yields an improved reconstruction for each quantizer cell. To illustrate this fact, we include the performance of this ET *zero-delay* decoder in Fig. 3. This causal decoder already provides a gain of about 0.2-0.7 dB in PSNR. The delayed decoder further improves the performance by almost 1 dB at most bit-rates. At very high bit-rates, as explained in Sec. 1 for the regular DPCM case, the gains due to delayed decoding are expected to be minimal. This is also observed in Fig. 3, where all three decoders provide similar performance at high rates. Similar results with the *Foreman* and *Container* sequences are provided in Table. 1. Then Table. 2 illustrates the gains obtained via the ET delayed decoder (which is a temporal processing operation) when the deblocking function (spatial smoothing) is retained or switched off. It is noteworthy that the gains due to the two complementary operations are indeed largely additive.

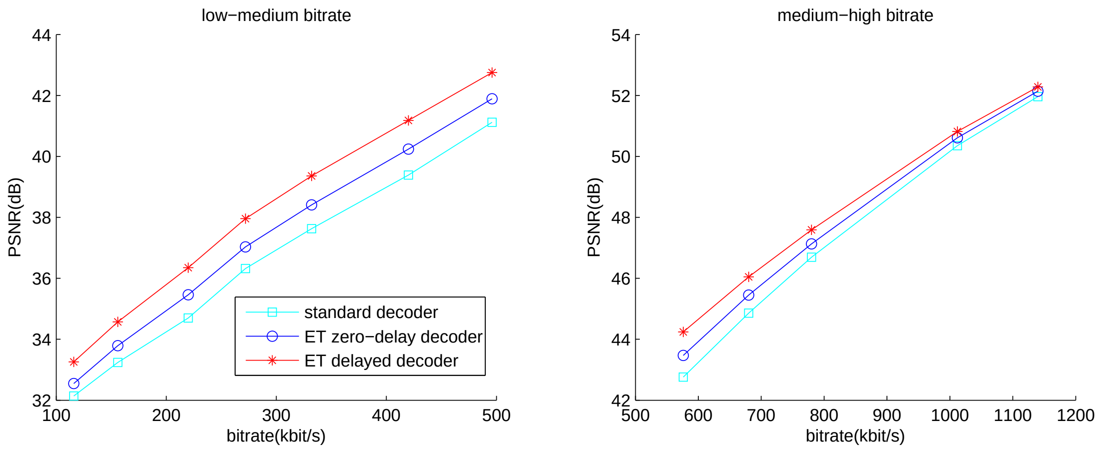

Figure 3: Comparison of the performance of standard H.264, and ET zero-delay and one-frame delayed decoders, on the *Coastguard* sequence

Rate(kbps)	Standard	ET Delayed Decoder
100	32.37	34.41
160	36.38	37.33
200	37.90	38.77
320	40.56	41.63
400	42.00	43.06
580	45.28	46.34
900	50.48	51.13

(a)

Rate(kbps)	Standard	ET Delayed Decoder
56	35.25	35.89
72	36.87	37.64
108	39.69	40.54
150	41.16	42.29
200	42.46	43.74
250	43.71	45.23
400	47.11	48.80

(b)

Table 1: PSNR values for (a) *Foreman* and (b) *Container*; at QCIF resolution

5 Conclusions

An ET delayed decoder for video sequences encoded using motion compensated prediction is proposed in this paper. The approach combines the motion compensation information from the past, the quantization index of transform coefficients in the current frame, and motion vectors and reconstructions of one future frame, in an ET framework, in order to produce the optimal estimate of the current frame. It achieves significant performance gains compared to the standard H.264 reconstruction. The decoder is applicable to all predictive coding systems, and can be specifically applied to any bit-stream that complies with the H.264 standard. It learns the underlying AR process from information available in the standard bit-stream and requires no side information. The additional complexity and memory requirements of the proposed approach are modest.

	Deblocking on		Deblocking off	
Rate(kb/s)	standard	ET delayed	standard	ET delayed
116	32.14	33.26	30.75	31.65
160	33.24	34.57	31.59	32.60
220	34.70	36.35	33.16	34.35
280	36.32	37.96	34.64	35.82
330	37.63	39.36	36.07	37.33
420	39.39	41.18	38.25	39.66
500	41.12	42.75	40.26	41.62
680	44.86	46.05	44.69	45.83

Table 2: PSNR comparison for *Coastguard* at QCIF resolution with deblocking filter active or switched off

References

[1] T. Wiegand et al, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 13, pp. 560–576, July 2003.

[2] M. L. Sethia and J. B. Anderson, "Interpolative DPCM," *IEEE Trans. Comm*, vol. 32, pp. 729–736, June 1984.

[3] W-W. Chang and J. D. Gibson, "Smoothed DPCM codes," *IEEE Trans. Comm.*, vol. 39, pp. 1351–1359, Sep 1991.

[4] V. Melkote and K. Rose, "Optimal delayed decoding of predictively encoded sources," *submitted to IEEE ICASSP*, 2010.

[5] M. Flierl and B. Girod, "Generalized B-pictures and the draft H.264/AVC video-compression standard," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 13, pp. 587–597, July 2003.

[6] O. G. Guleryuz and M. T. Orchard, "Rate-distortion based temporal filtering for video compression," *Proc. IEEE DCC*, pp. 122–131, Jan 1996.

[7] P. Ishwar and K. Ramchandran, "On decoder-latency versus performance trade-offs in differential predictive coding," *Proc. IEEE ICIP*, vol. 2, pp. 1097–1100, Oct 2004.

[8] F. Muller, "On the motion compensated prediction error using true motion fields," *Proc. IEEE ICIP*, vol. 3, pp. 781–785, Nov 1994.

[9] K. Rose and S. L. Regunathan, "Toward optimality in scalable predictive coding," *IEEE Trans. Image Proc.*, vol. 10, pp. 965–976, July 2001.

[10] M. Barkowsky et al., "Temporal trajectory aware video quality measure," *IEEE J. Sel. Topics Signal Process*, vol. 3, pp. 266–279, 2009.

[11] F. Bellifemine et al., "Statistical analysis of the 2D-DCT coefficients of the differential signal for images," *Sig. Proc.: Img. Comm.*, pp. 477–488, May 1992.

[12] JVT, "H.264/AVC reference software," July 2009.