ON THE MODELING OF MOTION IN WYNER-ZIV VIDEO CODING

M. Tagliasacchi, S. Tubaro, A. Sarti *

Dipartimento di Elettronica e Informazione
Politecnico di Milano,
Milan - Italy

ABSTRACT

In the past few years, a number of practical video coding schemes following distributed source coding principles have emerged. One of the main goals of distributed video coding (DVC) is to enable a flexible distribution of the computational complexity between the encoder and the decoder, while approaching the coding efficiency of conventional closed-loop motion-compensated predictive codecs. In this paper we perform a rate-distortion analysis of a well-known Wyner-Ziv architecture, while focusing our attention on the impact of the motion modeling that is used for generating the side information at the decoder. Our analysis is structured according to a Kalman filtering problem and it allows us to compare three different scenarios: motion estimation at the encoder; motion interpolation at the decoder; and motion extrapolation at the decoder.

Index Terms— Video coding, motion analysis, Kalman filtering

1. INTRODUCTION

Distributed Video Coding (DVC) is a new video coding paradigm based on the principles of distributed source coding [1, 2]. DVC enables a flexible distribution of the computational complexity between the encoder and the decoder, together with an increased robustness against channel losses. In this paper we consider the Wyner-Ziv codec that was first presented in [2] and further developed in [3], with the goal of evaluating the rate-distortion performance when different motion models are used.

The work presented in this paper is partially inspired by [4]. The main differences lie in the fact that our analysis is based on a Kalman filtering approach. This choice allows us to decouple the noise term related to the natural evolution of scene motion, from the observation noise, which is introduced by motion estimation inaccuracy. In addition, we explicitly model the case that uses motion interpolation, and this is a widely adopted choice in the literature.

2. PDWZ VIDEO CODEC ARCHITECTURE

The pixel domain Wyner-Ziv (PDWZ) video codec we refer to in this paper is based on the work in [2]. This coding architecture offers a pixel domain intra-frame encoder and inter-frame decoder with very low computational encoder complexity. The proposed encoding scheme is by far (several orders of magnitude) less complex then traditional video coding that performs motion estimation at the encoder. Figure 1 illustrates the global architecture of the PDWZ codec. Previously reconstructed frames are used at the decoder to generate the side information. In the literature we distinguish two cases: motion interpolation, when the previous ($\hat{X}_{t-1}$) and the next ($\hat{X}_{t+1}$) frame are used to synthesize the side information for X_t; motion extrapolation, when the the previous two frames ($\hat{X}_{t-1}$, $\hat{X}_{t-2}$) are used instead. More complex algorithms can be used, which combine both motion interpolation and extrapolation, although our analysis will focus on these two cases only. Each pixel in the Wyner-Ziv frame is uniformly quantized. Bitplane extraction is performed from the entire image and then each bitplane is fed to a turbo encoder to generate a sequence of parity bits. At the decoder, the generated side information will be used by the turbo decoder and reconstruction modules. The decoder operates in a bitplane-by-bitplane basis and begins by decoding the most significant bitplane and it only proceeds to the next bitplane after each bitplane is successfully turbo-decoded (i.e. when most of the errors are corrected).

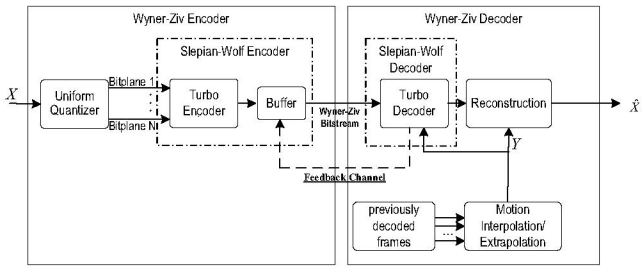

Fig. 1. Block diagram of the pixel domain Wyner-Ziv codec.

3. PROBLEM STATEMENT

Today's video coding architectures conforming to the Wyner-Ziv paradigm are unable to achieve the coding efficiency of conventional motion-compensated predictive codecs. The existing gap can be attributed to different reasons:

1. *Lack of side information at the encoder*: Let X and Y be two correlated random sequences. 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 $R^{WZ}_{X|Y}(D)$ the case when only the decoder has access to Y. The Wyner-Ziv theorem states that, in general, $R^{WZ}_{X|Y}(D) \geq R_{X|Y}(D)$ but $R^{WZ}_{X|Y}(D) = R_{X|Y}(D)$ for Gaussian memoryless sources and MSE as distortion measure. Therefore, a coding efficiency loss $\Delta R_1 = R^{WZ}_{X|Y}(D) - R_{X|Y}(D)$ is observed in

*Distributed Source Coding Project, funded by the Italian Ministry of Education, University and Research

applications when the hypothesis of the Wyner-Ziv theorem are not satisfied.

2. *Entropy coding losses*: DVC coding schemes use channel coding tools to perform source coding. Since the practical channel codes used in the context of DVC (Turbo, LDPC, etc.) only approach the Shannon's bound, a coding efficiency loss ΔR_2 stems from this fact.

3. *Motion model inaccuracy*: While motion-compensated predictive codecs do their best in order to accurately model the motion at the encoder side, Wyner-Ziv video coders perform the same operation at the decoder. Therefore, the side information Y available at the encoder side, i.e. the best motion compensated prediction of the current frame, is not available at the decoder, where a worse version of Y, say $\tilde{Y}$, can be generated by motion interpolation and/or extrapolation. This results in a coding efficiency loss ΔR_3. This paper will focus on the estimation of this term.

4. ANALYSIS OF RATE-DISTORTION PERFORMANCE

Following the same steps as in [4], let $X(t)$ denote the current frame and $\hat{X}(k), k \in D$ the previously-decoded frames in the frame buffer D. Let $Y_i(t)$ be the side information generated by the side information generator g_i

$$Y_i(t) = g_i(\hat{X}(k), k \in D), \quad i = 1, 2\,. \tag{1}$$

The residual frame is

$$e_i(t) = X(t) - Y_i(t), \quad i = 1, 2\,. \tag{2}$$

The power spectrum of the residual frame can be expressed as

$$\Phi_{ee}(\omega) = \Phi_{ss}(\omega) - 2Re\{\Phi_{ce}(\omega)\} + \Phi_{cc}(\omega) \tag{3}$$

$$\Phi_{cs}(\omega) = \Phi_{ss}(\omega) E[e^{-j\omega^T \Delta}] = \Phi_{ss}(\omega) e^{\frac{1}{2}\omega^T \omega \sigma_\Delta^2} \tag{4}$$

$$\Phi_{cc}(\omega) = \Phi_{ss}(\omega) \tag{5}$$

where $\Delta = (\Delta_x, \Delta_y)$ is the motion vector error, i.e. the difference between the motion vector used and the true motion vector.

$$\Phi_{ee}(\omega) = 2\Phi_{ss}(\omega) - 2\Phi_{ss}(\omega) e^{\frac{1}{2}\omega^T \omega \sigma_\Delta^2} \tag{6}$$

$$\frac{\Phi_{ee}(\omega)}{\Phi_{ss}(\omega)} = 2 - 2e^{\frac{1}{2}\omega^T \omega \sigma_\Delta^2} \tag{7}$$

The rate saving over INTRA-frame coding is [5]:

$$\Delta R = \frac{1}{8\pi^2} \int_{-\pi}^{+\pi} \int_{-\pi}^{+\pi} \log_2 \frac{\Phi_{ee}(\omega)}{\Phi_{ss}(\omega)} d\omega\,. \tag{8}$$

Hence, the difference between the two systems using two motion vectors MV_1 and MV_2 is

$$\begin{aligned} \Delta R_{1,2} &= \Delta R_1 - \Delta R_2 \\ &= \frac{1}{8\pi^2} \int_{-\pi}^{+\pi} \int_{-\pi}^{+\pi} \log_2 \frac{\Phi_{ee,1}(\omega)}{\Phi_{ee,2}(\omega)} d\omega \\ &= \frac{1}{8\pi^2} \int_{-\pi}^{+\pi} \int_{-\pi}^{+\pi} \log_2 \frac{1 - e^{\frac{1}{2}\omega^T \omega \sigma_{\Delta_1}^2}}{1 - e^{\frac{1}{2}\omega^T \omega \sigma_{\Delta_2}^2}} d\omega\,. \end{aligned} \tag{9}$$

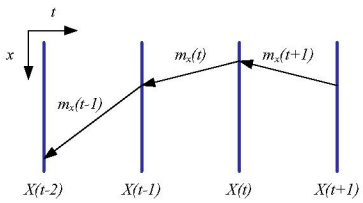

Fig. 2. Motion trajectory model.

5. MOTION MODELING

Wyner-Ziv video codecs generate the side information at the decoder, using previously reconstructed frames. Two main approaches have been explored in the literature:

- Motion interpolation: One every N frames (most often $N = 2$) is labeled as key frame and it is INTRA encoded. The resulting group of picture (GOP) structure is $I - WZ - \ldots - WZ - I$. For each Wyner-Ziv frame, the side information is obtained by motion compensated interpolation of the previous and next key frame.
- Motion extrapolation: Only the first frame is a key frame and is INTRA coded. The resulting GOP structure is $I - WZ - WZ - \ldots$. The side information for Wyner-Ziv frame is estimated through the motion extrapolation of the previously-decoded frames (both Wyner-Ziv and key frames).

In this paper we want to compare these two approaches with the conventional case where the encoder performs motion estimation. In order to be able to model both motion interpolation and motion extrapolation we drift apart from the work in [4], introducing our analysis based on Kalman filtering.

We denote by $\mathbf{m}(t) = (m_x(t), m_y(t))$ the true motion that a pixel/block is subjected to. Due to its nature, the motion is time-varying. We assume here a simple auto-regressive model

$$\mathbf{m}(t) = \rho \mathbf{m}(t-1) + \mathbf{z}(t)\,. \tag{10}$$

Unlike [4], where $\mathbf{m}(t-1)$ is the motion associated to the same spatial location at time $t-1$, here the autoregressive model is assumed to be valid throughout the motion trajectories. This means that $\mathbf{m}(t-1)$ is the motion vector corresponding to the spatial location pointed by $\mathbf{m}(t)$. Figure 2 illustrates this fact by means of an example for the case where only one of the two motion vector components are considered.

We assume that the two components of the noise term $\mathbf{z}$ are statistically independent, and each of them can be modeled as a zero mean white random process with the same variance $\sigma_z^2 = \sigma_{z_x}^2 = \sigma_{z_y}^2$.

From equation (10), we can immediately derive that

$$\sigma_m^2 = \sigma_{m_x}^2 = \sigma_{m_y}^2 = \frac{\sigma_z^2}{1 - \rho^2}\,. \tag{11}$$

Intuitively, σ_m^2 is an indication of motion complexity, i.e. a high value of σ_m^2 suggests that large displacements are expected. On the other hand, ρ measures the temporal coherence of the motion model. A value of ρ close to one indicates that motion has approximately uniform velocity (no acceleration/deceleration), covered/uncovered areas and scene changes are negligible. In spite of ρ, we use an

Table 1. σ_m^2 and $cSNR$ (Frame size QCIF, Block size 8×8

Sequence	Foreman	Coast.	Carphone	Stefan	Table	Silent
σ_m^2	9.46	0.76	7.23	15.40	2.09	1.93
$cSNR$(dB)	1.54	5.69	0.44	1.58	1.05	0.86

equivalent representation introduced in [4], the motion temporal correlation signal-to-noise ratio

$$cSNR = 10 \log_{10} \frac{\sigma_m^2}{\sigma_z^2} = 10 \log_{10} \frac{1}{1-\rho^2} . \quad (12)$$

Table 1 gives an indication of the parameters for a set of real sequences [4].

In the following, we distinguish three cases:

- *Motion estimation (ME) at the encoder*: The motion estimation algorithm has access to $X(t)$ and $X(t-1)$. The observed motion is
$$\mathbf{n}(t) = \mathbf{m}(t) + \mathbf{w}(t) . \quad (13)$$
The observed motion differs from the true motion $\mathbf{m}(t)$ only because of motion vector accuracy (we neglect errors due to quantization, reflections, illumination changes). If $1/M$ pixel accuracy is used, the noise term $w_{x,y}(t)$ can be assumed to be uniformly distributed between $-1/2M$ and $+1/2M$. The resulting variance is $\sigma_w^2 = \sigma_{w_x}^2 = \sigma_{w_y}^2 = 1/12M^2$
- *Motion interpolation (MI) at the decoder*: We assume here a $I - WZ - I$ GOP structure. The motion interpolation algorithm has access to $X(t-1)$ and $X(t+1)$. The estimated motion is therefore
$$\mathbf{n}(t) = \mathbf{m}(t) + \mathbf{m}(t+1) + \mathbf{w}(t) , \quad (14)$$
where $\mathbf{w}(t)$ is defined as above.
- *Motion extrapolation (MX) at the decoder*: The motion extrapolation algorithm has access to $X(t-1)$ and $X(t-2)$. The estimated motion is therefore
$$\mathbf{n}(t) = \mathbf{m}(t-1) + \mathbf{w}(t) , \quad (15)$$
where $\mathbf{w}(t)$ is defined as above.

It is possible to combine equation (10) with equations (13), (14) and (15), respectively, to write the problem in the canonical form prescribed by Kalman filtering

$$\mathbf{m}(t) = \mathbf{F}\mathbf{m}(t-1) + \mathbf{v_1}(t) \quad (16)$$

$$\mathbf{n}(t) = \mathbf{H}\mathbf{m}(t) + \mathbf{v_2}(t) . \quad (17)$$

Table 5 shows how the three aforementioned problems are mapped onto a set of state-observation equations together with the other quantities needed to solve the Kalman filtering problem.

Going back to our original problem, we want to obtain an estimate $\hat{\mathbf{m}}(t) = E[\mathbf{m}(t)|\mathbf{n}(t), \mathbf{n}(t-1), \ldots, \mathbf{n}(0)] = \hat{\mathbf{m}}(t|t)$ of $\mathbf{m}(t)$ given a noisy observation $\mathbf{n}(t)$. Kalman filter theory states that it is possible to relate the variance of the error on the state of the Kalman predictor ($\nu(t) = \mathbf{m}(t) - \hat{\mathbf{m}}(t|t-1)$) at time $t+1$ with that at time t via the RDE (Riccati Differential Equation)

$$P(t+1) = FP(t)F^T + V_1 - K(t)(HP(t)H^T + V_2)K^T , \quad (18)$$

where $P(t) = E[\nu(t)\nu(t)^T]$ and the Kalman gain $K(t)$ is defined as $K(t) = (FP(t)H^T + V_{12})(HP(t)H^T + V_2)^{-1}$. The variance of the error on the state of the Kalman filter is related to the one of the Kalman predictor by

$$E[(\mathbf{m}(t) - \hat{\mathbf{m}}(t|t))^2] = \quad (19)$$
$$= P(t) - P(t)H^T[HP(t)H^T + V_2]^{-1}HP(t) .$$

Upon convergence, $P(t) = P(t-1) = P$. Using this equation into (18) we obtain the ARE (Algebraic Riccati Equation) and we can solve for P. We set $P^{filt} = P - PH^T[HPH^T + V_2]^{-1}HP$.

$$P^{filt} = \begin{bmatrix} \sigma_\Delta^2 & 0 \\ 0 & \sigma_\Delta^2 \end{bmatrix} , \quad (20)$$

where σ_Δ^2 is used in equation (9) in order to evaluate the rate-distortion gain of a given motion modeling scheme.

6. SIMULATION RESULTS

In this section, we compare the rate-distortion performance of the three algorithms used to generate the side information: motion estimation (ME), motion interpolation (MI) and motion extrapolation (MX). As a benchmark, we consider a system that encodes the difference between X_t and X_{t-1}, i.e. assuming zero motion. We dub this system FD, for frame difference. In this case, the expected variance is $\sigma_\Delta^2 = \sigma_m^2$.

For each of the three cases, we numerically evaluate σ_Δ^2 for various values of ρ and σ_m^2 and we use equation (9) to assess the rate rebate with respect to the FD system. Figure 3 plots ΔR as a function of $cSNR$ for different values of σ_m^2 (from the rate-distortion theory we know that $\Delta PSNR \simeq 6\Delta R$).

We notice the same behavior in all cases. When we keep the variance of the motion vectors σ_m^2 fixed, increasing temporal coherence of motion ($cSNR$), we expect larger gains with respect to FD, that does not exploit motion at all. We can gain a further insight comparing the plots in Figure 3 for the same value of σ_m^2. In fact, we notice that the maximum rate rebate is obtained for ME, followed by MI and finally by MX. For example, setting $\sigma_m^2 = 1$, $\Delta R_{ME/MI}$ is as large as -0.8 bpp, while $\Delta R_{MI/MX}$ is as large as -3.5 bpp. We notice that the gap between MI and ME gets narrower increasing $cSNR$ and decreasing σ_m^2. This is reasonable, since motion interpolation works well when the temporal coherence is high and displacements tend to be small.

We have to point out that these figures refer to the encoding of a single Wyner-Ziv frame. In a complete video coding architecture, the total rate, including key frames, should be considered. While for ME and MX all frames (but the first one) take advantage of inter frame dependencies, for MI one every N frames (N = 2 in this case), is intra coded. Therefore, these results tend to overestimate the performance of motion interpolation. A fairer analysis should take into account the rate needed to encode key frames. The latter depends on the spatial power spectral density of the sequence ($\Phi_{ss}(\omega)$), whereas our simplified analysis depends only on considerations based on motion.

Table 3 compares the performance of practical algorithms used to generate the side information [4]. The behavior suggested by our analysis is confirmed by these figures. In fact, ME outperforms the other approaches for all of the tested sequences, followed by MI and MX in this order. The gain of ME over MI ranges between +0.44dB for *Coastguard* up to +2.08dB for *Carphone*. We notice that these two sequences are characterized by the highest and lowest temporal correlation ($cSNR$) respectively, validating the conclusions of our analysis.

ME	MI	MX
$m_{x,y}(t+1) = \rho m_{x,y}(t) + z_{x,y}(t)$ $n_{x,y}(t) = m_{x,y}(t) + w_{x,y}(t)$	$m_{x,y}(t+1) = \rho m_{x,y}(t) + z_{x,y}(t)$ $n_{x,y}(t) = (1+\rho) m_{x,y}(t) + z_{x,y}(t) + w_{x,y}(t)$	$m_{x,y}(t+1) = \rho m_{x,y}(t) + z_{x,y}(t)$ $n_{x,y}(t) = \frac{1}{\rho} m_{x,y}(t) - \frac{1}{\rho} z_{x,y}(t-1) + w_{x,y}(t)$
$F = diag[\rho]$ $H = diag[1]$ $V_1 = diag[\sigma_z^2]$ $V_2 = diag[\sigma_w^2]$	$F = diag[\rho]$ $H = diag[1+\rho]$ $V_1 = diag[\sigma_z^2]$ $V_2 = diag[\sigma_z^2 + \sigma_w^2], V_{12} = diag[\sigma_z^2]$	$F = diag[\rho]$ $H = diag[1/\rho]$ $V_1 = diag[\sigma_z^2]$ $V_2 = diag[(1/\rho^2)\sigma_z^2 + \sigma_w^2]$

Table 2. Systems of equations corresponding to the three motion models: ME - Motion Estimation, MX - Motion Extrapolation, MI - Motion Interpolation.

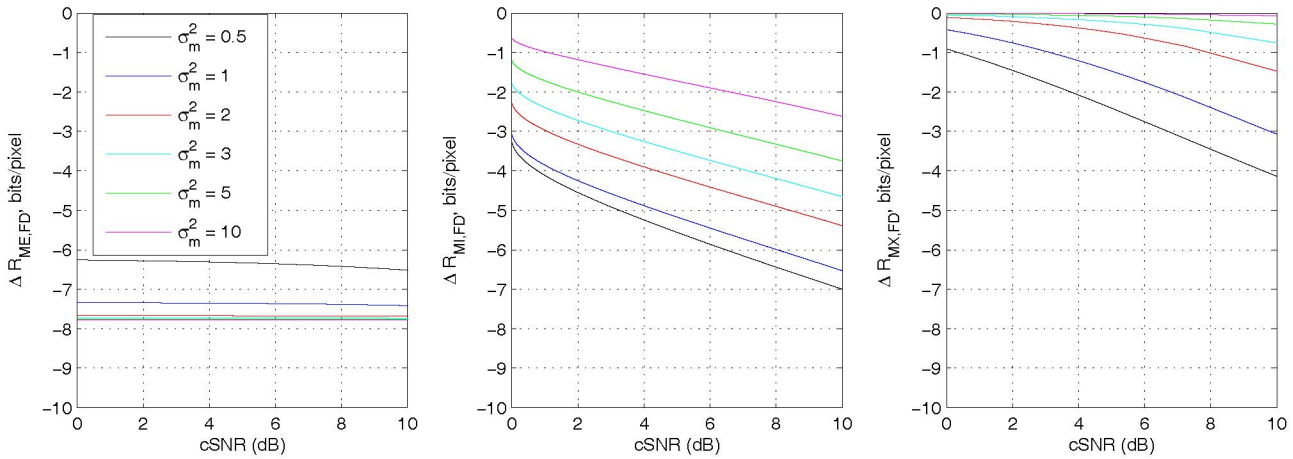

Fig. 3. Comparison of the rate-distortion performance between motion estimation (ME), motion interpolation (MI) and motion extrapolation (MX). The rate rebate with respect to simple frame differencing (FD) is shown

Table 3. Comparison of different algorithms used to generate the side information (in dB)

Sequence	Foreman	Coast.	Carphone	Stefan	Table	Silent
FD	28.17	27.32	29.77	19.48	27.10	32.43
MX	30.66	30.82	29.03	22.88	30.48	33.62
MI	32.56	32.17	31.72	23.54	32.20	36.09
ME	33.21	32.61	33.80	24.84	33.86	38.11
$ME - FD$	+5.04	+4.29	+4.03	+5.36	+6.76	+5.68
$ME - MX$	+2.55	+1.79	+4.77	+1.96	+3.38	+4.49
$ME - MI$	+0.65	+0.44	+2.08	+1.30	+1.66	+2.02

7. CONCLUSIONS

In this paper we analyze the coding efficiency of a Wyner-Ziv coding architecture, with respect to conventional schemes that perform motion estimation at the encoder. Unlike similar approaches recently appeared in the literature, our analysis builds upon Kalman filtering in order to explicitly model the case that performs motion compensated interpolation at the decoder. The results of the analysis are validated by experimental results on real test sequences. Our current activities focus on studying the rate-distortion performance of other motion modeling schemes adopted in the literature such as mixed motion extrapolation/interpolation [6], motion interpolation with longer GOP size and motion extrapolation using hash functions [7]. We need to emphasize that the results presented in this paper are valid for the coding architecture summarized in Section 2. In fact, they do not apply to other distributed source coding based coding schemes such as PRISM [1], where the decoder is able to build a motion model comparable to the one that can be obtained by performing motion estimation at the encoder.

8. REFERENCES

[1] Rohit Puri and Kannan 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.

[2] Anne Aaron, Rui Zhang, and Bernd Girod, "Wyner-Ziv coding of motion video," in *Proceedings of the 36th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2002, vol. 1, pp. 240–244.

[3] Bernd Girod, Anne Aaron, Shantanu Rane, and David Rebollo Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, January 2005.

[4] Li Zhen and E. J. Delp, "Wyner-Ziv video side estimator: Conventional motion search methods revisited," in *Proceedings of the International Conference on Image Processing*, Genova, Italy, September 2005, pp. 825 – 828.

[5] T. Berger, *Rate Distortion Theory*, Prentice Hall, 1971.

[6] Luís Natário, Catarina Brites, João Ascenso, and Fernando Pereira, "Extrapolating side-information for low-delay pixel-domain distributed video coding," in *Internationa Workshop on Very Low Bitrate Video Coding*, Costa del Rei, Sardinia, Italy, September 2005.

[7] Anne Aaron and Bernd Girod, "Wyner-ziv video coding with low-encoder complexity," in *Picture Coding Symposium*, San Francisco, CA, December 2004.