HIDDEN MARKOV MODEL FOR DISTRIBUTED VIDEO CODING

V. Toto-Zarasoa, A. Roumy and C. Guillemot

IRISA/University of Rennes, Campus Universitaire de Beaulieu, 35042 Rennes-Cedex, France.

ABSTRACT

This paper addresses the problem of asymmetric distributed coding of correlated binary Hidden Markov Sources, modeled as a Gilbert-Elliott process. The model parameters are estimated with an estimation-decoding Expectation-Maximization algorithm. The rate gain obtained by accounting for the memory of the sources is first assessed theoretically. The method is then shown to improve the PSNR versus rate performance of a Distributed Video Coding system, based on Low-Density Parity-Check codes.

Index Terms— Source coding, Channel coding, Parameter estimation, Linear codes.

1. INTRODUCTION

Distributed Source Coding (DSC) refers to the problem of separate encoding and joint decoding of correlated sources. Slepian and Wolf (SW) have established in 1973 [1] that, for two dependent binary sources X and Y, the lossless compression rate bound $H(X,Y)$ can be achieved when encoding the two sources separately, as long as each single rate is greater than the conditional entropy, and joint decoding is performed. The lossy equivalent of the Slepian-Wolf theorem for two correlated continuous-valued sources has been formulated by Wyner and Ziv (WZ) [2]. The correlation between the two sources can be modeled as a virtual Binary Symmetric Channel (BSC). A way to achieve this compression rate is to use channel codes, this solution is shown to be optimal in [3] in the sense that a capacity achieving channel code can be turned into an optimal DSC code. The first practical DSC solutions used syndrome-based channel codes [4], as Low-Density Parity-Check (LDPC) codes [5].

In this paper, we investigate non-uniform memory sources where the memory is spread over the entire sampled data. More precisely, the sequence of symbols is generated by a Hidden Markov Model (HMM): the two-state Gilbert-Elliott (GE) process [6]. The probability of a given symbol is dependent only on the current state. The GE channel was investigated by Garcia-Frias [7] as a model for the correlation between X and Y, we use it as a model for the source. The estimation of the GE channel parameters was carried out by [8]. We propose the joint estimation of the model parameters and the decoding of X using an *Expectation Maximization* (EM) algorithm [9].

Distributed video coding (DVC) considers the successive images of some video sequence as correlated distributed sources. The main idea in DVC is to migrate the system complexity at the decoder side; meaning that the encoder only carries out the most basic operations, and the decoder is assumed to have enough power to carry out the joint decoding. The *WZ video coding system* DISCOVER [10], is one of the most effective and developed distributed video codec of the moment. We show that the source model we propose fit to the generated video bit planes, then we place our estimation-decoding module in the DISCOVER codec, and we show that a considerable gain results by exploiting the memory.

This paper is structured as follows. Section 2 presents the GE source model, and defines its parameters. Section 3 describes the EM algorithm used for the joint estimation-decoding. The performance of the EM algorithm is then assessed with synthetic sources in section 4. Finally, section 5 presents the integration of the EM algorithm to the DISCOVER codec, and shows the rate gain.

2. THE HMM MODEL FOR DISTRIBUTED SOURCE CODING

In order to efficiently model the video data flow, we consider a two-state HMM: the GE process. The advantage of this model is that one can model a source with infinite memory with only few parameters. Let us first review the GE process. The source X is dependent on an underlying and persistent Markov state process. Let Σ be a finite Markov process having two realizations (called *states*) $\mathbf{0}$ and $\mathbf{1}$. Σ is Markovian with memory one in the sense that its realization at position n only depends on its realization at position $(n-1)$. In each state, the source X is drawn according to a Bernoulli law of parameter $p_{\mathbf{0}}$ and $p_{\mathbf{1}}$ respectively (Fig 1). We define the transition probabilities $t_{\mathbf{00}}$, $t_{\mathbf{10}}$, $t_{\mathbf{01}}$ and $t_{\mathbf{11}}$ between the states. Since $t_{\mathbf{00}} = 1 - t_{\mathbf{01}}$ and $t_{\mathbf{11}} = 1 - t_{\mathbf{10}}$, the set of parameters of the model is $\theta = (p_{\mathbf{0}}, p_{\mathbf{1}}, t_{\mathbf{10}}, t_{\mathbf{01}})$. They are defined by:

$$\begin{array}{ll} p_{\mathbf{0}} = \mathbb{P}_\theta(X_n = 1 | \Sigma_n = \mathbf{0}), & p_{\mathbf{1}} = \mathbb{P}_\theta(X_n = 1 | \Sigma_n = \mathbf{1}) \\ t_{\mathbf{10}} = \mathbb{P}_\theta(\Sigma_n = \mathbf{0} | \Sigma_{n-1} = \mathbf{1}), & t_{\mathbf{01}} = \mathbb{P}_\theta(\Sigma_n = \mathbf{1} | \Sigma_{n-1} = \mathbf{0}) \end{array} \quad (1)$$

where $\mathbf{\Sigma} = \Sigma_1^N$ is an N-long state sequence, and $\sigma = \sigma_1^N = \{\mathbf{0}, \mathbf{1}\}^N$ its realization, and where $\mathbf{X} = X_1^N$ is an N-long source sequence with realization $\mathbf{x} = x_1^N = \{0,1\}^N$.

This research was partially supported by the French European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract n.216715).

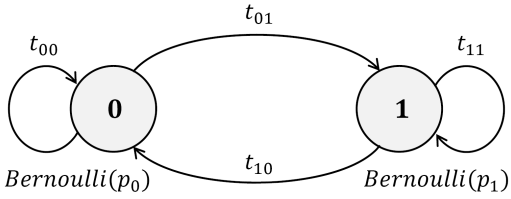

Fig. 1. Diagram for the state generation.

In DSC, a second source Y with realization $\mathbf{y}$ is correlated to X, where the correlation is modeled as a virtual BSC with parameter p. More precisely, let Z represent the difference between X and Y, we assume that $Y = X \oplus Z$ with $\mathbb{P}(Y \neq X) = \mathbb{P}(Z = 1) = p$. In asymmetric DSC, the source Y is available at the decoder and X has to be recovered. As the sources have memory, X can be compressed at the conditional entropy-rate $H(\mathcal{X}|\mathcal{Y}) = \lim_{N\to\infty} \frac{1}{N} H(\mathbf{X}|\mathbf{Y})$. For our GE model, this rate can be computed as in [11]. This entropy-rate is lower than that of a uniform source. As an illustration, for a GE source having the parameters ($p_{\mathbf{0}} = 0.07, p_{\mathbf{1}} = 0.7, t_{\mathbf{10}} = 0.03, t_{\mathbf{01}} = 0.01$), $H(\mathcal{X}|\mathcal{Y}) = 0.5$ occurs for a BSC of parameter $p = 0.299$, instead of $p = 0.11$ if the source is uniform. $H(\mathcal{X}|\mathcal{Y})$ is significantly decreased.

Motivated by the higher compression rate achievable when taking into account the memory of the sources, we propose an *estimation-decoding* algorithm that decodes the source, and estimates its model parameters.

3. ESTIMATION-DECODING EM ALGORITHM

The compression of the source X is done with a channel code. Let $\mathbf{H}$ be the parity-check matrix of a linear (N, K) code. In the DSC setup, $\mathbf{x}$ of length N is mapped to its syndrome $\mathbf{s_x} = \mathbf{Hx}$ of length $(N - K)$, and $\mathbf{y}$ is transmitted to the decoder. The decoder must estimate $\hat{\mathbf{x}}$ from $\mathbf{s_x}$, $\mathbf{y}$, and the correlation factor p: $\hat{\mathbf{x}}$ is the closest sequence to $\mathbf{y}$ with the syndrome $\mathbf{s_x}$. When the source has no memory and when the channel code is an LDPC code, this search can be efficiently performed with a modified Message-Passing (MP) [5].

Our aim is to jointly estimate the memory source and its parameter θ. This can be performed by an iterative algorithm called the EM algorithm [9]. More precisely, the EM is an iterative optimization procedure that learns new parameters of a stochastic model based on the improvement of a likelihood computed from a set of observables. As side products we obtain estimates of the hidden variables $\mathbf{x}$ and σ. Therefore the EM can be seen as an *estimation-decoding* algorithm. Let l be the label of the current iteration, and $\{\mathbf{x}^l, \theta^l, \sigma^l\}$ the current estimates. Then, the next value of θ is computed so as to maximize the mean log-likelihood function

$$\theta^{(l+1)} = \arg\max_{\theta} \mathbb{E}_{\mathbf{X},\mathbf{\Sigma}|\mathbf{Y},\mathbf{S_X},\theta^l} \log\left(\mathbb{P}_\theta(\mathbf{y}, \mathbf{x}, \sigma, \mathbf{s_x})\right)$$

where $\mathbb{P}_\theta(\mathbf{y}, \mathbf{x}, \sigma, \mathbf{s_x})$ stands for the likelihood function. Fig. 2 presents the graph that helps us describe the *message-passing* (MP) [12] that is performed for the joint estimation-decoding of θ, σ and $\mathbf{x}$. In the following, we explicit the expectation and the maximization steps.

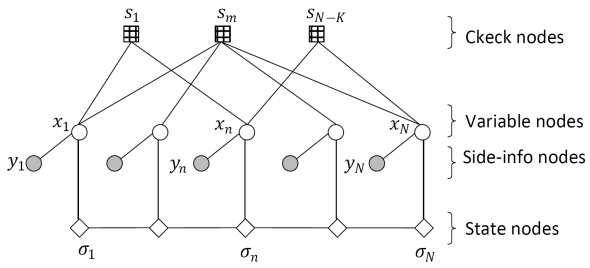

Fig. 2. Graph describing the *joint estimation-decoding* that is performed.

3.1. Expectation step

Using the Bayes rule:$\mathbb{P}_\theta(\mathbf{y}, \mathbf{x}, \sigma, \mathbf{s_x}) = \mathbb{P}_\theta(\mathbf{y}, \mathbf{s_x}|\mathbf{x}, \sigma)\mathbb{P}_\theta(\mathbf{x}, \sigma)$ where $\mathbb{P}_\theta(\mathbf{y}, \mathbf{s_x}|\mathbf{x}, \sigma)$ is independent of θ. Therefore, we only need to compute $\mathbb{E}_{\mathbf{X},\mathbf{\Sigma}|\mathbf{Y},\mathbf{S_X},\theta^l}\left(\log\left(\mathbb{P}_\theta(\mathbf{x}, \sigma)\right)\right)$, where

$$\log\left(\mathbb{P}_\theta(\mathbf{x}, \sigma)\right) = \log\left(\mathbb{P}_\theta(\sigma_1)\right) + \sum_{n=2}^{N} \sum_{i=\mathbf{0}}^{\mathbf{1}} \sum_{j=\mathbf{0}}^{\mathbf{1}} \delta_{\sigma_{n-1}=i,\sigma_n=j} \log(t_{ij})$$

$$+ \sum_{n=1}^{N} \sum_{i=\mathbf{0}}^{\mathbf{1}} \delta_{\sigma_n=i,x_n=1} \log(p_i) + \delta_{\sigma_n=i,x_n=0} \log(1-p_i) \tag{2}$$

$$\text{where } \delta_{bool} = \begin{cases} 1, \ if \ bool = true \\ 0, \ otherwise \end{cases}$$

3.2. Maximization step

Here, the mean log-likelihood function is maximized with respect to θ under the constraints $p_i \in [0, 1]$,$t_{ij} \in \ ]\,0, 1\,[$, and $\sum_{j\in\{\mathbf{0},\mathbf{1}\}} t_{ij} = 1$. Using Lagrange multipliers, the new parameters are given by

$$p_i^{(l+1)} = \frac{\sum_{n=1}^{N} \mathbb{P}_{\theta^l}(\Sigma_n = i|\mathbf{y}, \mathbf{s_x}) \mathbb{P}_{\theta^l}(X_n = 1|\Sigma_n = i, \mathbf{y}, \mathbf{s_x})}{\sum_{n=1}^{N} \mathbb{P}_{\theta^l}(\Sigma_n = i|\mathbf{y}, \mathbf{s_x})}$$

$$t_{ij}^{(l+1)} = \frac{\sum_{n=2}^{N} \mathbb{P}_{\theta^l}(\Sigma_{n-1} = i, \Sigma_n = j|\mathbf{y}, \mathbf{s_x})}{\sum_{n=2}^{N} \mathbb{P}_{\theta^l}(\Sigma_n = i|\mathbf{y}, \mathbf{s_x})}$$

Since the graph of our source model has cycles (see Fig. 2), the exact *a posteriori* probabilities (APP) are too complex. However, the MP algorithm is known to provide a good approximation for these quantities. In order to simplify the notation, the APP and their approximation through the MP algorithm are denoted the same. $\mathbb{P}_{\theta^l}(\sigma_n|\mathbf{y}, \mathbf{s_x})$ and $\mathbb{P}_{\theta^l}(\sigma_{n-1}, \sigma_n|\mathbf{y}, \mathbf{s_x})$ are computed in section 3.3 using a BCJR-like *forward-backward algorithm*. $\mathbb{P}_{\theta^l}(x_n|\sigma_n, \mathbf{y}, \mathbf{s_x})$ is computed using an LDPC decoding *sum-product algorithm* (section 3.4).

3.3. BCJR-like forward-backward algorithm

The forward-backward algorithm is run on the trellis which states are $\mathbf{0}$ and $\mathbf{1}$ generating the source symbols. We define

$$\begin{aligned}\gamma_{i,j}^{n,(n+1)} &= \mathbb{P}_\theta(y_n|\Sigma_n = i, \mathbf{s_x}) \cdot \mathbb{P}_\theta(\Sigma_{n+1} = j|\Sigma_n = i, \mathbf{s_x}) \\ \alpha_j^n &= \sum_{i\in\{\mathbf{0},\mathbf{1}\}} \alpha_i^{(n-1)} \cdot \gamma_{i,j}^{(n-1),n} \\ \beta_i^n &= \sum_{i\in\{\mathbf{0},\mathbf{1}\}} \gamma_{i,j}^{n,(n+1)} \cdot \beta_j^{(n+1)}\end{aligned} \tag{3}$$

The equations in (3) define the recursions, where $\gamma_{i,j}^{n,(n+1)}$ is the transition probability between the states i at position n and j at position $(n+1)$, α_i^n is the forward probability for the source being in state i at position n, β_i^n is the backward probabilities for the source being in state i at position n.

Now we define the source states APP:

$$\begin{aligned}&\mathbb{P}_\theta(\Sigma_n = i, \mathbf{y}|\mathbf{s_x}) = \alpha_i^n \cdot \beta_i^n \\ &\mathbb{P}_\theta(\Sigma_{n-1} = i, \Sigma_n = j, \mathbf{y}|\mathbf{s_x}) = \alpha_i^{(n-1)} \cdot \gamma_{i,j}^{(n-1),n} \cdot \beta_j^n\end{aligned} \tag{4}$$

Renormalizing $\mathbb{P}_\theta(\sigma_n, \mathbf{y}|\mathbf{s_x})$ and $\mathbb{P}_\theta(\sigma_{n-1}, \sigma_n, \mathbf{y}|\mathbf{s_x})$, we get $\mathbb{P}_\theta(\sigma_n|\mathbf{y}, \mathbf{s_x})$ and $\mathbb{P}_\theta(\sigma_{n-1}, \Sigma_n = j|\mathbf{y}, \mathbf{s_x})$.

3.4. LDPC decoding sum-product algorithm

The messages are passed on the bipartite graph composed of the variable nodes and the check nodes, knowing the soft estimates of σ^l and θ^l. Let d_{xn} be the degree of x_n, and d_{sm} be the degree of s_m (see Fig. 2).

• Messages from y_n to x_n: Computation of the intrinsic: $I_n = (1-2y_n)\log\left(\frac{1-p}{p}\right) + \log\left(\frac{1-\hat{p}_X}{\hat{p}_X}\right)$, with $\hat{p}_X = p_{\mathbf{0}}^l \cdot \mathbb{P}_\theta(\Sigma_n = \mathbf{0}|\mathbf{y}, \mathbf{s_x}) + p_{\mathbf{1}}^l \cdot \mathbb{P}_\theta(\Sigma_n = \mathbf{1}|\mathbf{y}, \mathbf{s_x})$
each I_n is mapped to the corresponding $E_{n,k}^{(in)}$.

• Variable to check messages , $\forall e \in [1, d_{xn}]$:

$$E_{n,e}^{(out)} = I_n + \sum_{k=1,k\neq e}^{d_{xn}} E_{n,k}^{(in)}$$

each $E_{n,e}^{(out)}$ is mapped to the corresponding $Q_{m,e}^{(in)}$.

• Check to variable messages, $\forall e \in [1, d_{sm}]$:

$$Q_{m,e}^{(out)} = 2\tanh^{-1}\left[(1-2s_n)\prod_{k=1,k\neq e}^{d_{sm}} \tanh\frac{Q_{m,e}^{(in)}}{2}\right]$$

each $Q_{m,e}^{(out)}$ is mapped to the corresponding $E_{n,k}^{(in)}$.

• Messages to the state nodes. We note $P_n = I_n + \sum_{k=1}^{d_{xn}} E_{n,k}^{(in)}$:

$$\begin{aligned}\mathbb{P}_\theta(X_n = 0|\sigma_n, \mathbf{y}, \mathbf{s_x}) &= \frac{e^{P_n}}{1+e^{P_n}} \\ \mathbb{P}_\theta(X_n = 1|\sigma_n, \mathbf{y}, \mathbf{s_x}) &= 1 - \mathbb{P}_\theta(X_n = 0|\sigma_n, \mathbf{y}, \mathbf{s_x})\end{aligned} \tag{5}$$

In this LDPC decoding, we have decided to propagate LLRs, which implies their conversion to probabilities in (5), for use at the maximization (3.2) and with the BCJR (3.3).

4. PERFORMANCE OF THE ESTIMATION-DECODING ALGORITHM

We consider a GE source of length $N = 1584$ (same length as the video bit planes) with parameters $\theta = (p_{\mathbf{0}} = 0.07, p_{\mathbf{1}} = 0.7, t_{\mathbf{10}} = 0.03, t_{\mathbf{01}} = 0.01)$. The LDPC code we use is of rate $\frac{1}{2}$, created using the Progressive Edge Growth principle. The syndrome, as well as the side-information, are transmitted to *two* different decoders: (1) the standard decoder that views X as a uniform source, (2) the proposed decoder knowing that X has memory. For the decoder exploiting the memory, the EM algorithm is initialized with $(p_{\mathbf{0}} = 0.49, p_{\mathbf{1}} = 0.51, t_{\mathbf{10}} = 0.1, t_{\mathbf{01}} = 0.1)$. The maximum iteration number is set to 100, which represents a good compromise between complexity and efficiency. The BERs of X and the estimated parameters are shown in Fig 3.

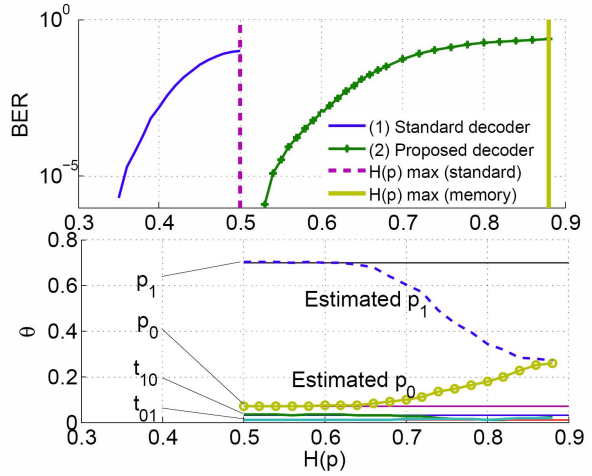

Fig. 3. Performances of the two decoders. The source has parameters $\theta = (p_{\mathbf{0}} = 0.07, p_{\mathbf{1}} = 0.7, t_{\mathbf{10}} = 0.03, t_{\mathbf{01}} = 0.01)$. When exploiting the memory, $H(\mathcal{X}|\mathcal{Y}) = 0.5$ occurs for $H(p) = 0.88$ ($p = 0.299$), instead of $H(p) = 0.5$ ($p = 0.11$).

Fig. 3 shows that the estimation-decoding EM algorithm manages to recover X better than the standard decoder, in particular X is recovered error-free for $H(p) = 0.5$ which corresponds to the SW bound for a uniform source. Besides the parameters are well estimated if the correlation between X and Y is high ($H(p) < 0.65$). When $H(p)$ gets closer to 0.88, the decoding of X fails (BER(X) $> 10^{-2}$ for $H(p) > 0.65$) and the parameter estimation fails as well.

5. DISTRIBUTED VIDEO CODING USING HMM

5.1. Review of the DISCOVER codec

The DISCOVER encoder [10] first separates the video frames into two sets: key frames and WZ frames. Key frames are conventionally encoded using an H264/AVC encoder in intra mode. WZ frames are first transformed using Discrete Cosine Transform (DCT), the resulting transform coefficients are quantized and organized into bands, where each band contains the coefficients associated to the same frequency in different blocks. The bits representing these coefficients are ordered bit plane per bit plane, and fed into an LDPC-based SW encoder which computes their syndromes. The syndrome bits are stored in a buffer, and progressively transmitted to the decoder until the bit plane is correctly decoded. That incremental rate-adaptive decoding prevents from re-encoding the data at each request. The bit planes are usually considered as independent uniformly distributed binary sources, but we

show in section 5.2 that they are better approximated by the GE model.

5.2. Accuracy of the GE source modeling

To evaluate the accuracy of the model, we observe the bursts lengths, i.e. the number of consecutive 1's, in the sequence of each bit plane. First, consider a source X without memory, it can be modeled as a Bernoulli process with parameter $p_X = \mathbb{P}(X = 1)$. In this case, the burst length distribution, $\mathcal{P}_k$, is defined as the probability of having a sequence with k consecutive 1's given that the sequence starts with a 0 (i.e. $0\underbrace{1\ldots1}_{k}0$). For the Bernoulli process, $\mathcal{P}_k = (1-p_X)p_X^{k-1}$. Therefore, for non memory binary sequences, the log-scale burst distribution, $\log(\mathcal{P}_k)$, is linear in the burst length.

We consider now the bit plane sequences obtained by the DISCOVER codec. For those sequences, we plot the *empirical burst length distribution* and the synthetic one obtained with the GE parameters. More precisely, given the GE parameters estimated with the EM algorithm described in section 3, the burst length distribution is generated: this is called the *synthetic distribution*. Fig. 4 shows the comparison for two different bit planes: one with memory (1) and the other with almost no memory (3). Interestingly, the empirical and the synthetic distributions match each other well.

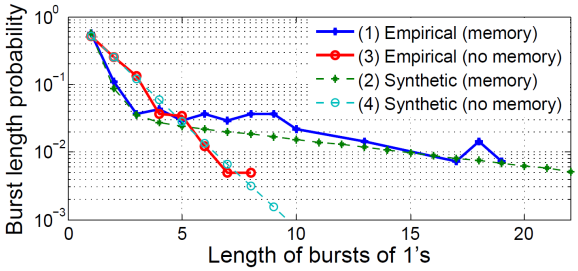

Fig. 4. Distribution of the bursts of 1's in two selected bit planes of *soccer*. The estimated parameters are $(p_0 = 0.112, p_1 = 0.974, t_{10} = 0.0621, t_{01} = 0.0435)$ for the bit plane with memory and $(p_0 = 0.483, p_1 = 0.484, t_{10} = 0.0942, t_{01} = 0.0937)$ for the one without memory.

5.3. Our contribution for DVC and experimental results

We place the estimation-decoding EM module in lieu of the SW decoder of DISCOVER. The results presented in Fig. 5 are obtain for all the frames of the $15Hz$ QCIF video sequences *hall monitor*, *foreman*, and *soccer*.

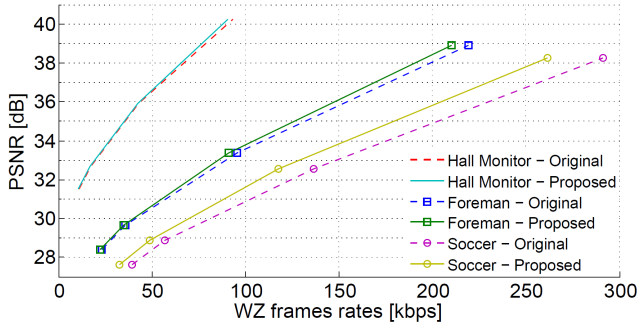

Fig. 5. The decoder that exploits the bit planes memory needs less rate to render the videos at the same PSNR. The GOP size is 2.

As we do not improve the rate of the key frames, only the results for the WZ frames are shown. While our model remains relatively simple (only two states), the gain, in terms of decreasing the rate allocated to the WZ frames, is significant. The decrease of rate for *hall monitor* is $2.54kbps$ (-2.73%) at the highest PSNR, while it is $8.76kbps$ (-4%) for *foreman*, and $29.55kbps$ (-10.14%) for *soccer*. This proves that it is worth taking into account the memory of the source.

6. CONCLUSION

We have proposed an asymmetric SW codec for two correlated binary non-uniform memory sources, based on LDPC codes. A hidden Markov model is taken for the source. We showed that substantial gain comes, which reduces the BER of the DSC system, when the iterative memory estimation is carried out together with the LDPC decoding. The decoder was finally incorporated into the DISCOVER codec, for practical DVC, using the memory on the WZ bit planes, and we presented results that prove the efficiency of that modification: the gain is up to 10.14% for the video sequence *soccer*.

7. REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Th.*, vol. 22, pp. 1–10, January 1976.

[3] A. Wyner, "Recent results in the shannon theory," *IEEE Transactions on Information Theory*, vol. 20, pp. 2–10, 1974.

[4] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE DCC*, pp. 158–167, March 1999.

[5] A. D. Liveris, Z. Xiong, , and C. N. Georghiades, "Compression of binary sources with side information at the decoder using ldpc codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, October 2002.

[6] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell-System Tech J.*, pp. 1977–1997, Sept. 1963.

[7] J. Garcia-Frias, "Decoding of low-density parity-check codes over finite-state binary markov channels," *IEEE Transactions on Communications*, vol. 52, no. 11, pp. 1840–1843, 2004.

[8] M. Mushkin and I. Bar-David, "Capacity and coding for the gilbert-elliot channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277–1290, Nov. 1989.

[9] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, pp. 257–286, 1989.

[10] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The discover codec: Architecture, techniques and evaluation," in *PCS*, Nov 2007.

[11] M. Rezaeian, "Computation of capacity for gilbert-elliott channels, using a statistical method," *CTW*, pp. 56–61, Feb. 2005.

[12] H-A. Loeliger, "An introduction to factor graphs," *IEEE Sig. Proc. Mag.*, vol. 21, no. 1, pp. 28–41, January 2004.