A MINIMUM MEAN SQUARE ERROR ESTIMATION AND MIXTURE-BASED APPROACH TO PACKET VIDEO ERROR CONCEALMENT

Daniel Persson and Thomas Eriksson

Chalmers University of Technology
Department of Signals and Systems
412 96 Göteborg
Sweden

ABSTRACT

In this paper, a minimum mean square error-optimized mixture-based estimator is used for packet video error concealment. At the same time as on-line computational complexity is reduced, performance in peak signal-to-noise ratio (PSNR) is increased in comparison with a Gaussian mixture-based estimator that obtains its parameters through probability density estimation by means of the expectation maximization algorithm. Moreover, our method increases performance in PSNR compared to several other previous error concealment algorithms.

Index Terms— Packet video, error concealment, estimation.

1. INTRODUCTION

Today's frequently used video compression methods utilize block-based motion-compensated inter-frame prediction, discrete cosine transform-based quantization, and variable length coding, to reduce the bit stream. High compression ratios are achieved in this way, but the video stream sensitivity to communication channel and storage media errors also increases. Among the various error resilience techniques used for combating this problem, methods that work at the decoder side without extra redundancy from the encoder are referred to as error concealment schemes. Error concealment methods are usually categorized into spatial approaches, that use only spatially surrounding pixels for estimation of lost blocks, and temporal approaches, that use motion information and pixels from previous frames [1].

In this paper, we combine spatial and temporal error concealment. The MVs are considered to be available through protection in a high priority layer, or estimated by the median of the MVs of the surrounding blocks [2], and we propose a mean square error (MSE) risk- and mixture-based estimator that classifies the situation of local image correlation in order to choose spatial or temporal error concealment. In the case of lost MVs, the proposed method may be combined with any method for finding the MVs. Performance in peak signal-to-noise ratio (PSNR) is increased, while on-line computational complexity is reduced, compared to our own GMM-based estimator with parameters retrieved by means of the expectation maximization (EM) algorithm that was reported on in [3], [4], and [5]. The new method also increases performance in PSNR compared with other previous methods that combine spatial and temporal error concealment, e.g. Shirani *et al*'s method [6], Zhu *et al*'s method [7], and motion compensated copying [2]. The rest of the paper is organized as follows. In Section 2, the minimum MSE (MMSE) estimation framework is defined, and an argumentation for a mixture-based estimator is given. Section 3 describes the experimental setup, and presents simulation results. The paper is concluded in Section 4.

2. MMSE ESTIMATION WITH MIXTURE-BASED ESTIMATOR

In this section, mixture-based packet video error concealment is formulated as a MMSE-problem. The proposed technique is a further development of the approach reported on in [3], [4], and [5], that has previously shown to increase PSNR compared to the schemes [6], [7], and [2].

2.1. Motivation of the proposed method

We start by summarizing [3], [4], and [5], and further discussing the advantages and the shortcomings of the method. Thereafter, the new approach is explained. In [3], [4], and [5], a group of neighboring pixel values, that are lost at the decoder side, are represented by elements of the stochastic vector variable X. An estimate

$$\hat{X} = g(Y) \tag{1}$$

is formed, where Y is a stochastic vector whose elements represent pixels in a neighborhood to X. Further, the MMSE problem

$$g^*(Y) = \arg\min_{g(Y)} \mathrm{E}[\|X \pm g(Y)\|_2^2] \tag{2}$$

is posed, that has the solution

$$g^*(Y) = \mathrm{E}(X|Y) = \int x f_{X|Y}(x|y)\, dx. \tag{3}$$

The density $f_{X,Y}(x, y)$ is modeled with a GMM

$$f_Z(z) = \sum_{m=1}^{M} \theta^{(m)} f^{(m)}(z) \tag{4}$$

where $Z^{\mathrm{T}} = [X^{\mathrm{T}}, Y^{\mathrm{T}}]$ and $f^{(m)}(z)$ are Gaussian distributions with means and covariances

$$\theta_Z^{(m)} = \begin{bmatrix} \theta_X^{(m)} \\ \theta_Y^{(m)} \end{bmatrix} \qquad C_{ZZ}^{(m)} = \begin{bmatrix} C_{XX}^{(m)} & C_{XY}^{(m)} \\ C_{YX}^{(m)} & C_{YY}^{(m)} \end{bmatrix} \tag{5}$$

and where the a priori weights $\theta^{(m)}$ are all positive and sum to one. The estimator (3) may then be written

$$g^*(Y) = \sum_{m=1}^{M} \theta^{(m)}(Y) \left(C_{XY}^{(m)} (C_{YY}^{(m)})^{*\,1} (Y \pm \theta_Y^{(m)}) + \theta_X^{(m)} \right) \tag{6}$$

where

$$\theta^{(m)}(Y) = \frac{\theta^{(m)} f^{(m)}(Y)}{\sum_{k=1}^{M} \theta^{(k)} f^{(k)}(Y)} \quad (7)$$

are a posteriori probabilities that sum to 1. A qualitative interpretation of the form of the mixture (6) is that the a posteriori weights $\theta^{(m)}(Y)$ are used for classifying the local image correlation, and choosing appropriate linear estimators $C_{XY}^{(m)}(C_{YY}^{(m)})^{*\,1}(Y \pm \theta_Y^{(m)}) + \theta_X^{(m)}$ for the situation at hand. As seen in [3], [4], and [5], the estimator (6) with $M > 1$ increases performance in PSNR compared to the linear estimator obtained when $M = 1$. Though the above scheme has shown to increase PSNR compared to several previous methods [6], [7], [2], a few remarks about its disadvantages can be made.

- $\pm$ The formulation in (1) and (2) leaves us with a density estimation problem in order to achieve $f_Z(z)$. In [3], [4], and [5], the density was achieved by means of the EM algorithm that is a maximum likelihood (ML) method, and therefore does not necessarily minimize the MSE. A consistent treatment, where all parameters of the final estimator are achieved by minimizing the MSE, would be more eligible. However, a MMSE formulation for finding the $\theta^{(m)}$, $\theta_Z^{(m)}$, and $C_{ZZ}^{(m)}$ of the estimator (6) would be an extremely difficult problem.

- $\pm$ The estimator (6) has a high computational complexity on-line, when X is estimated from Y. The quadratic forms in the exponents of $f^{(m)}(Y)$ contribute significantly to the computational complexity.

- $\pm$ In [3], [4], and [5], a vector Z with 64 dimensions was employed. It would be desirable to increase the number of dimensions of Y substantially, but with the estimator (6), this would be expensive due to matrix multiplications, both for estimator optimization, and for on-line computations.

To summarize, the benefit of a simplification of the estimator (6) is threefold: the parameters may be obtained in the MMSE-sense, on-line complexity is reduced, and the estimate may be based on more information. In what follows, we rephrase the problem stated in (1) and (2) with this in mind. We form the new estimate

$$\hat{X} = g(Y, \theta) \quad (8)$$

where $g(y, \theta)$ is the estimator function with parameters θ. Optimal parameters θ^* are found by minimizing the MSE risk

$$\theta^* = \arg\min_{\theta} \mathrm{E}(\|X \pm g(Y, \theta)\|_2^2). \quad (9)$$

In contrast to the parameters of the estimator (6), that depend on ML estimation, the parameters of (8) are achieved in the MMSE sense. As PSNR is a standard measure of video quality [7], [6], [3], [4], [5], and a function of the MSE, the MSE risk was chosen. By minimizing the MSE, the PSNR is maximized. For the proposed estimator, we choose a form that is heavily inspired by the solution in (6). The means $\theta_X^{(m)}$ and $\theta_Y^{(m)}$ are removed, the matrices $C_{XY}^{(m)}(C_{YY}^{(m)})^{*\,1}$ are replaced by matrices $A^{(m)}$, the functions $\theta^{(m)}$ are replaced by simpler functions $\theta^{(m)}$, and two subsets of Y are employed in the estimator (8): Y_C for classification in the a posteriori weights (C stands for classification), and Y_P for prediction (P stands for prediction). By introducing Y_C and Y_P, more information may be used in the new simple a posteriori weights $\theta^{(m)}$, than for the prediction. We now have

$$g(Y, \theta) = \sum_{m=1}^{M} \theta^{(m)}(Y_\mathrm{C})\, A^{(m)} Y_\mathrm{P} \quad (10)$$

$$\theta^{(m)}(Y_\mathrm{C}) = \frac{\theta^{(m)} h^{(m)}(Y_\mathrm{C})}{\sum_{k=1}^{M} \theta^{(k)} h^{(k)}(Y_\mathrm{C})} \quad (11)$$

$$h^{(m)}(Y_\mathrm{C}) = \exp\left(\pm c^{(m)} \frac{\|Y_\mathrm{C}^{(m),1} \pm Y_\mathrm{C}^{(m),2}\|_2^2}{D^{(m)}} \right) \quad (12)$$

where $Y_\mathrm{C}^{(m),1}$ and $Y_\mathrm{C}^{(m),2}$ are vectors containing elements of Y_C, $c^{(m)} > 0$ is a scalar, and $D^{(m)}$ is the dimension of the vectors $Y_\mathrm{C}^{(m),1}$ and $Y_\mathrm{C}^{(m),2}$. In the discussion of the GMM-based estimator, we concluded that different a posteriori weights focus on different situations of video correlation. In accordance with this, the vectors $Y_\mathrm{C}^{(m),1}$ and $Y_\mathrm{C}^{(m),2}$ should be chosen so that a specific situation of video correlation is given priority. For example, in order to generate a mixture component that focuses on spatial correlation, $Y^{(m),1}$ and $Y^{(m),2}$ should be chosen so that the exponent of (12) incorporates the difference of the values of many spatially neighboring pixels. The exponent of (12) may be represented as

$$\pm c^{(m)} \frac{\|Y_\mathrm{C}^{(m),1} \pm Y_\mathrm{C}^{(m),2}\|_2^2}{D^{(m)}} = \pm \frac{c^{(m)}}{D^{(m)}} Y_\mathrm{C}^\mathrm{T} W^{(m)} Y_\mathrm{C} \quad (13)$$

where the matrices $W^{(m)}$ not are positive definite. This means that the new mixture (10) not is GMM-based though it has the essential functionality of the estimator (6). Also, (10) is easily optimized in the MMSE way, it reduces on-line complexity compared to (6), and it may take a Y_C with high number of dimensions without severely increasing the computational complexity. The parameters θ are $\{\theta^{(m)}, c^{(m)}, A^{(m)}\}$. Though the estimation of X from Y is performed on-line, the optimal estimator parameters θ^* are found off-line.

2.2. Algorithm for solving the MMSE estimation problem

Since the solving of the MMSE problem (9), with the proposed estimator (10), does not have a closed form solution, an algorithm for iterative solving of (9) with (10) is now proposed. We solve iteratively for the parameters $A^{(m)}$, $\theta^{(m)}$ and $c^{(m)}$. The algorithm increases PSNR in every iteration.

The parameters $A^{(m)}$**.** The matrices $A^{(m)}$ are first considered. It is easy to show that the problem that consists in finding

$$A^{(r)*} = \arg\min_{A^{(r)}} \mathrm{E}[\|X \pm g(Y, \theta)\|_2^2] \quad (14)$$

is convex. The proof will be given in a longer journal paper. By setting the derivative with respect to $A^{(r)}$ equal to zero, and thereafter solving for $A^{(r)}$, we achieve

$$A^{(r)*} = R_1 (R_2)^{*\,1} \quad (15)$$

$$R_1 = \mathrm{E}\Bigg[\theta^{(r)}(Y_C) X Y_P^\mathrm{T} \pm \sum_{m=1, m\neq r}^{M} \theta(Y_\mathrm{C})^{(r)} \theta(Y_\mathrm{C})^{(m)} A^{(m)} Y_\mathrm{P} Y_\mathrm{P}^\mathrm{T} \Bigg] \quad (16)$$

$$R_2 = \mathrm{E}\Big[(\theta^{(r)}(Y_\mathrm{C}))^2 Y_\mathrm{P} Y_\mathrm{P}^\mathrm{T} \Big]. \quad (17)$$

The parameters $\theta^{(m)}$ ***and*** $c^{(m)}$**.** Since the MSE is not convex in $\theta^{(m)}$, these parameters are updated by searches in the space of possi-

ble vectors $\theta = [\theta^{(1)}, ..., \theta^{(M)}]^{\mathrm{T}}$. In each iteration, the MSE is compared at the point describing the parameter set θ obtained in the previous iteration, and in the two points $\theta \pm \theta\,[\theta^{(1)}a(1), ..., \theta^{(M)}a(M)]^{\mathrm{T}}$, where θ is some scalar, and a is a normalized random vector with elements $a(1)$ to $a(M)$. The point that yields the minimum value of the MSE is chosen as the new θ. This algorithm is faster than gradient descent with backtracking line search, since many evaluations of the MSE are avoided. The parameters $c^{(m)}$ are updated in the same way.

For optimization in practice, the expectations are replaced by arithmetic means.

3. EXPERIMENTS

In this section, the proposed method is evaluated and compared to methods suggested by other authors. Details of the simulations, which are chosen to fit state-of-the-art block-based video coders, are given in Section 3.1. These conditions are impartial to all the compared schemes. Results of the experiments are presented in Section 3.2.

3.1. Simulation prerequisites

Coding and packetization. We focus on predictively coded frames (P-frames) (An application of the proposed method to restoration of intra-coded frames (I-frames) is completely analogous). MVs are calculated for 8 ± 8-blocks with a search range of 8 pixels for each component. The coder works in the limit of perfect quantization. Each row of 16 ± 16-blocks of pixels is divided into 8 ± 8-blocks of pixels, that are interleaved into two packets, so that if a packet is lost, there is a high probability that surrounding pixels are available. Concerning the MVs, two situations are investigated. In the first scenario, the MVs are coded with their respective pixel information in the same packet, and when a packet is lost, lost MVs are estimated by the median of MVs of the neighboring blocks [2]. In the second scenario, MVs are protected in a high priority layer, and considered available. These assumptions are similar to the assumptions in [7].

Errors. The packets are randomly assigned as lost. Simulations are run for loss probabilities ranging from 0.05 to 0.3. In the case of MVs protected in a high priority layer, only packets containing pixel information are lost.

Data. We use the luminance component of 124 MPEG-1 movies from [8]. The movies are divided into two independent sets, one for off-line optimization of the parameters θ, and another for evaluation. In order to show the robustness of our scheme, we use more movies for the evaluation than for the training. The sets used for parameter optimization and evaluation contain 35 and 89 randomly selected movies respectively.

Benchmarking. The proposed estimator is compared to other schemes that mix spatial and temporal information given the MVs: Shirani *et al*'s method [6], Zhu *et al*'s method [7], and GMM-based error concealment [3], [4], [5], as well as to motion compensated copying [2]. All methods use the same MV information.

Proposed estimator. Each lost 8 ± 8-block is repaired by splitting it into four 4 ± 4-blocks whose pixels are represented by X. The surrounding pixels represented by Y_{C} and Y_{P} will vary in the experiments, and will be described for each experiment. We choose to work with few mixture components since we strive for low on-line complexity. A mixture with $M = 2$ components is investigated. Mixture component 1 focuses on spatial correlation by employing $Y_{\mathrm{C}}^{(1),1}$ and $Y_{\mathrm{C}}^{(1),2}$ such that all possible differences between closest spatial neighbors in Y_{C} are included in the exponent. In the same way, mixture component 2 focuses on temporal correlation by employing $Y_{\mathrm{C}}^{(2),1}$ and $Y_{\mathrm{C}}^{(2),2}$ such that all possible differences between closest temporal neighbors in Y_{C} are included in the exponent.

Varying available information. Spatially surrounding pixels may not be available, because the block in question is a border block, or because several consecutive blocks are lost. Different models are obtained and stored for each of these cases. In the case when no spatial surrounding information is available, we reduce our estimator to the special linear MMSE solution to (9) and (10) $A^{(1)} = E(X^{\mathrm{T}}Y_{\mathrm{P}})(E(Y_{\mathrm{P}}^{\mathrm{T}}Y_{\mathrm{P}}))^{*\,1}$ that is obtained when $M = 1$, and now Y_{P} only represents pixels in the previous frame. By assuming mirror invariance of the model, only four model cases need to be stored.

Off-Line parameter estimation. The parameters of the estimator are found off-line. A choice of $M = 2$ mixture components was previously made. The parameters $\theta = [\theta^{(1)}, \theta^{(2)}]^{\mathrm{T}}$ and $c = [c^{(1)}, c^{(2)}]^{\mathrm{T}}$ are initialized by $\frac{1}{2}[1, 1]^{\mathrm{T}}$, and the parameters $A^{(m)}$ are initialized by the linear MMSE solution to (9) and (10) obtained when $M = 1$, i.e. $A^{(m)} = E(X^{\mathrm{T}}Y_{\mathrm{P}})(E(Y_{\mathrm{P}}^{\mathrm{T}}Y_{\mathrm{P}}))^{*\,1}$. For the update of $\theta^{(m)}$ and $c^{(m)}$, we choose $\theta = 0.1$. In each iteration, 1470 000 realizations of $Z^{\mathrm{T}} = [X^{\mathrm{T}}, Y^{\mathrm{T}}]$ are used. In each of the 10 first iterations, two iterations are performed for $\theta^{(m)}$ and $c^{(m)}$ respectively, as well as one iteration for the $A^{(m)}$. In the ten final iterations, only the $A^{(m)}$ are updated.

3.2. Results

Estimator comparison. The purpose of this first experiment is to see that our estimator strategy yields higher performance in PSNR and lower computational complexity than the GMM-based estimator [3], [4], [5], when both methods use the same number of mixture components and the same information. In this experiment, we set $Y_{\mathrm{C}} = Y_{\mathrm{P}}$, as illustrated in Figure 1, and pixels surrounding a lost 8 ± 8-block are guaranteed to be available. The results presented in Table 1 show that the proposed method gives better performance in PSNR, with around a third of the computational complexity.

Error concealment comparison. The proposed estimator (10) is compared to different error concealment schemes. In preliminary simulations, the approach with Y_{C} and Y_{P} chosen as in Figure 2, was compared to the approach with Y_{C} and Y_{P} chosen as in Figure 1. Since Y_{C} and Y_{P} chosen as in Figure 2 yielded better results while maintaining low complexity, this approach was chosen for comparison to other previous error concealment methods. Note that by choosing Y_{C} as in Figure 2, significantly more information is used for the classification, than if Y_{C} would have been chosen as in Figure 1. In the experiments, pixels surrounding a lost 8 ± 8-block were not guaranteed to be available. The errors propagate in a few tens of frames in each movie. Figure 3 presents the results for the case when the MVs are not available, and replaced by the median of the MVs of the surrounding blocks. In Figure 4, we see the results in the case when the MVs are available. The proposed method gives best performance in PSNR. The GMM-based method with $M = 20$ gives a comparable result, but this comes at a cost of around 26 times higher on-line computational complexity than the proposed method. A GMM-based estimator with $M = 2$ has 2.6 times higher computational complexity than the proposed scheme. Examples in a longer journal paper will show that our method also improves subjective visual performance.

4. CONCLUSION

In this paper, MMSE- and mixture-based error concealment, that may be run in real-time, is presented. For a motivation of the new technique, we use as our starting point an estimator methodology based on GMMs and probability density estimation by means of the

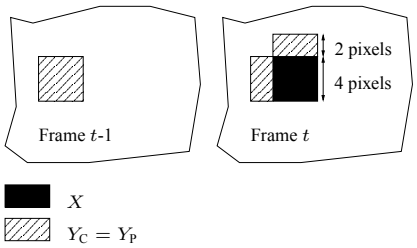

Fig. 1. Illustration of variables to be used with the proposed estimator in the estimator comparison. Blocks of size 8 ± 8 are divided into four 4 ± 4-blocks that are estimated separately. The vector X is lost at the decoder side, and is estimated by using a vector of surrounding pixels $Y_C = Y_P$.

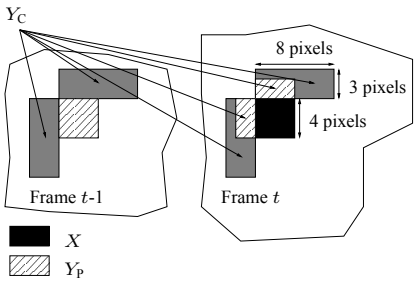

Fig. 2. Illustration of variables to be used with the proposed estimator in the error concealment comparison. Blocks of size 8 ± 8 are divided into four 4 ± 4-blocks that are estimated separately. The vector X is lost at the decoder side, and is estimated by using vectors of surrounding pixels Y_C and Y_P.

MV	Type of test	Proposed method, $M = 2$	GMM, $M = 1$	GMM, $M = 2$
Not available	Closed	32.1	31.6	31.7
Not available	Open	31.4	31.0	31.0
Available	Closed	34.0	33.7	33.7
Available	Open	33.4	33.1	33.1

Table 1. Comparison of results in PSNR for the proposed method, and the GMM-based method. The number M is the number of mixture components.

EM algorithm. All parameters of the new estimator are easily obtained off-line in the MMSE sense. The proposed estimator gives better performance in PSNR than the GMM approach when using the same data and number of mixture components. At the same time, the proposed estimator has lower computational complexity. Moreover, the new technique may, without substantial increase in computational complexity, incorporate much bigger surrounding to the lost block as input to the estimator. We finally show that the proposed method gives an important increase in performance compared to a range of other well-known previous error concealment methods.

5. REFERENCES

[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.

[2] P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by atm cell loss," in *Proc. ICASSP*, Mar. 1992, pp. 545–548.

[3] D. Persson and P. Hedelin, "A statistical approach to packet loss concealment for video," in *Proc. ICASSP*, Mar. 2005, pp. 293 – 296.

[4] D. Persson, T. Eriksson, and P. Hedelin, "Qualitative analysis of video packet loss concealment with gaussian mixtures," in *Proc. ICASSP*, May 2006, pp. II–961 – II–964.

[5] D. Persson, T. Eriksson, and P. Hedelin, "Statistical packet video error concealment," IEEE Trans. Image Processing, in review.

[6] S. Shirani, F. Kossentini, and R. Ward, "A concealment method for video communications in an error-prone environment," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1122–1128, June 2000.

[7] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, June 1993.

[8] "Prelinger archives," http://www.archive.org/details/prelinger, Online resource.

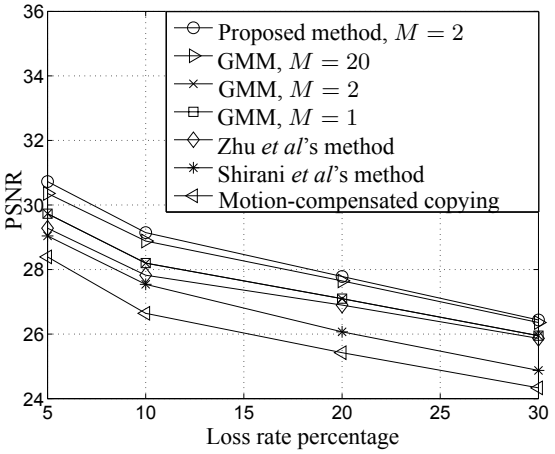

Fig. 3. Comparisons of different error concealment schemes, in the case when the MVs are not available, and replaced by the median of the MVs of the surrounding blocks.

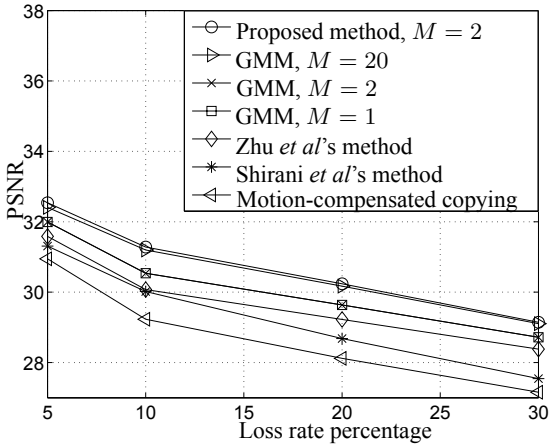

Fig. 4. Comparisons of different error concealment schemes, in the case when the MVs are available.