# Mixture Model- and Least Squares-Based Packet Video Error Concealment


Daniel Persson and Thomas Eriksson





Daniel Persson and Thomas Eriksson, Mixture Model- and Least Squares-Based Packet Video Error Concealment, 2009, IEEE Transactions on Image Processing, (19), 5, 1048-1054.
http://dx.doi.org/10.1109/TIP.2009.2014261

# Mixture Model- and Least Squares-Based Packet Video Error Concealment

Daniel Persson and Thomas Eriksson

***Abstract*—A Gaussian mixture model (GMM)-based spatio-temporal error concealment approach has recently been proposed for packet video. The method improves peak signal-to-noise ratio (PSNR) compared to several famous error concealment methods, and it is asymptotically optimal when the number of mixture components goes to infinity. There are also drawbacks, however. The estimator has high online computational complexity, which implies that fewer surrounding pixels to the lost area than desired are used for error concealment. Moreover, GMM parameters are estimated without considering maximization of the error concealment PSNR. In this paper, we propose a mixture-based estimator and a least squares approach for solving the spatio-temporal error concealment problem. Compared to the GMM scheme, the new method may base error concealment on more surrounding pixels to the loss, while maintaining low computational complexity, and model parameters are found by an algorithm that increases PSNR in each iteration. The proposed method outperforms the GMM-based scheme in terms of computation-performance tradeoff.**

***Index Terms*—Block-based packet video, spatio-temporal error concealment, least squares (LS) estimation.**

## I. Introduction

VIDEO communication over the Internet and via mobile phones is growing ever more popular. The block-based state-of-the-art video-coding scheme H.264/MPEG-4 part 10 achieves high compression efficiency, but the resulting bit stream is vulnerable to communication channel impairments. Packet errors occur because of various transmission channel problems, and may be characterized by a simultaneous loss of bigger amounts of data locally in the video stream. Many error resilience techniques for combating this problem exist [1]. Methods that work at the decoder side without extra redundancy from the encoder are referred to as error concealment schemes. In order to show how our contribution fits into the history of the error concealment problem, we provide a short revision of previous techniques.

### A. Previous Efforts

Error concealment methods are usually categorized into spatial approaches, that only use spatially surrounding pixels for estimation of lost blocks, and temporal approaches, that estimate the motion field and use pixels from previous frames for replacement of the lost blocks.

Spatial methods may yield better performance than temporal methods in scenes with high motion, or after a scene change. A large variety of spatial strategies have been suggested, see, e.g., [2]–[7]. The method in [8] that replaces lost pixels with weighted averages of boundary pixels forms part of the error concealment approach implemented in the H.264 JM decoder [9].

Details inside lost blocks can not be recreated by spatial schemes. In this case, information from the past frame may improve the result. Rather than using the block in the previous frame at the same spatial position as the lost block for temporal EC, the motion-compensated block should be used [1]. If the motion vector is available at the decoder side, it can be utilized. When the motion vector is also lost, it has to be estimated, which is the challenge in temporal error concealment. Motion estimation is often performed by using the median of the motion vectors of the surrounding blocks, or the motion vector of the corresponding block in the previous frame [10]. The motion vector that yields the minimum variation between a replacement block and its spatial surrounding is chosen as an estimate in [11]. This algorithm has been adopted for temporal error concealment in the H.264 JM decoder [9]. Other more advanced temporal error concealment methods are, e.g., [12]–[16].

From an information-theoretic view, spatio-temporal approaches should improve the pure spatial and temporal efforts. The method in [2] was extended to yield a replacement from received transform coefficients, pixels on the border of the lost block, and pixels from a previous frame in [17]. In [18], motion vectors are estimated, the prediction error is modeled as a Gaussian Markov random field, and a maximum *a posteriori* estimate of the prediction error for the lost block is formed. Some recent contributions are, e.g., mode selection in [19], combined and iterative spatio-temporal error concealment in [20], and error concealment by spatio-temporal boundary matching and a partial differential equation-based algorithm in [21].

A mixture of principal components model for error concealment of tracked objects is proposed in [22]. We have recently proposed a spatio-temporal Gaussian mixture model (GMM)-based error concealment method [23] for block-based packet video that has showed good results in peak signal-to-noise ratio (PSNR) compared to other error concealment schemes [10], [17], [18]. Gaussian mixture models have been employed successively for several other tasks in image processing, for example object detection in images in [24] and

Manuscript received October 25, 2006; revised November 19, 2008. Current version published April 10, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sohail Dianat.

The authors are with the Department of Signals and Systems, Chalmers University of Technology, S-412 96 Göteborg, Sweden (e-mail: f97danp@chalmers.se; thomase@chalmers.se).

Digital Object Identifier 10.1109/TIP.2009.2014261

noise reduction, image compression, and texture classification in [25]. For error concealment of speech, Gaussian mixture models have been used in [26] and [27]. Moreover, a GMM has the ability to approximate probability densities arbitrarily closely asymptotically [25]. Our GMM-based method in [23] divides the error concealment problem into an offline GMM parameter estimation problem, and an online minimum mean-square error (MMSE)-based estimation of the lost blocks using surrounding pixels as well as the estimated GMM parameters.

The GMM-based estimator for lost blocks is, however, computationally complex online, even when fewer pixel values are regarded in the modeling. An often prevailing technique for GMM parameter estimation is the expectation maximization algorithm [29], that yields a nondecreasing likelihood of the realizations in the training set in each iteration. The PSNR of the estimate, which would be a more appropriate performance measure for our application, is overlooked since estimating the GMM parameters by maximizing the estimator PSNR is not mathematically tractable.

### B. Our Contribution

In this paper, we suggest a new mixture-based estimator for spatio-temporal EC. Estimator parameters are obtained in the least squares (LS) sense by an iterative algorithm offline, and in such a way that error concealment performance in PSNR is increased in every iteration. The estimator is thereafter employed on-line for error concealment, using the previously obtained parameters and pixels surrounding the lost area.

Compared to the method in [23], fewer online floating point operations are needed, and more pixels surrounding the lost block may be taken into account in the estimator without a significant increase in computational complexity. Our formulation of the problem has also, compared to the formulation in [23], the advantage that we do not need to estimate probability densities explicitly, which means that many parameters that are insignificant for obtaining our final estimator are avoided. Our formulation can in this way be seen as a means for combating what is known as "the curse of dimensionality," or "the empty space phenomenon," that is the exponential growth of the needed amount of training data with dimension of the estimated probability density [25]. Preliminary investigations for this paper were presented in [28].

The remainder of this paper is organized as follows. Our proposed method is presented in Section II. In Section III, the proposed scheme is compared to previous efforts by means of simulations. The paper is concluded in Section IV.

## II. Mixture Model-Based EC

Since the proposed technique is a further development of the approach reported on in [23], we start by summarizing [23] and discussing the advantages and the shortcomings of the method. Thereafter the new technique is introduced.

### A. Analysis of GMM-Based Error Concealment

In [23], a group of pixel values that are lost at the decoder side are represented by elements of the stochastic vector variable $x$, and are replaced by employing neighboring pixel values represented by elements of the stochastic vector variable $y$. A GMM with $M$ component densities describes the relationship between the vectors $x$ and $y$

$$f(z) = \sum_{m=1}^{M} \rho^{(m)} f^{(m)}(z) \tag{1}$$

where $z^{\mathrm{T}} = [x^{\mathrm{T}}, y^{\mathrm{T}}]$, and the *a priori* weights $\rho^{(m)}$ are all positive and sum to one. The functions $f^{(m)}(z)$ are Gaussian distributions with means and covariances

$$\mu_z^{(m)} = \begin{bmatrix} \mu_x^{(m)} \\ \mu_y^{(m)} \end{bmatrix}, \quad C_{zz}^{(m)} = \begin{bmatrix} C_{xx}^{(m)} & C_{xy}^{(m)} \\ C_{yx}^{(m)} & C_{yy}^{(m)} \end{bmatrix}. \tag{2}$$

An estimator $\hat{x}$ for the lost pixels $x$ is found by considering an MMSE problem

$$\hat{x}(y) = \arg\min_{\hat{x}'(y)} \mathrm{E}_{X,Y}\left[\|x - \hat{x}'(y)\|_2^2\right] \tag{3}$$

which has the solution

$$\hat{x}(y) = \mathrm{E}_X(x|y) = \int x f(x|y)\, dx \tag{4}$$

where $f(x|y)$ is a conditional model pdf that may be derived from $f(x,y)$. The solution in (4) leaves us with a new problem, namely the estimation the parameters of (1). This problem is solved by the expectation maximization algorithm [29]. Using (1) and (2), the estimator (4) can be written [23]

$$\hat{x}(y) = \sum_{m=1}^{M} \phi^{(m)}(y) \times \left(C_{xy}^{(m)} \left(C_{yy}^{(m)}\right)^{-1} \left(y - \mu_y^{(m)}\right) + \mu_x^{(m)}\right) \tag{5}$$

where

$$\phi^{(m)}(y) = \frac{\rho^{(m)} f^{(m)}(y)}{\sum_{k=1}^{M} \rho^{(k)} f^{(k)}(y)}. \tag{6}$$

The functions (6), which sum to 1, are referred to as *posteriori* weights. Error concealment is thus conducted by

1) offline GMM parameter estimation by means of the expectation maximization algorithm using a video database;
2) online error concealment using (5) and the pixels $y$ surrounding the lost pixels $x$.

As seen in [23], the estimator (5) with $M > 1$ increases performance in PSNR compared to the linear estimator obtained when $M = 1$. A qualitative interpretation for the increased performance for $M > 1$ is that the *posteriori* weights (6) classify the situation of local image correlation in a soft manner, and combine together several linear estimators $C_{xy}^{(m)}(C_{yy}^{(m)})^{-1}(y - \mu_y^{(m)}) + \mu_x^{(m)}$ that are specialized for different situations. We will adapt this view of (5) in order to propose the new mixture-based estimator.

Though the above scheme has shown to increase PSNR compared to the previous methods [18], [17], [10], a few remarks about its disadvantages can be made.

- GMM parameter estimation with the expectation maximization algorithm yields a nondecreasing log-likelihood of the realizations in the training set in each iteration [30].

An increase of log-likelihood of the vectors in the training set is, however, not necessarily linked to an increase in PSNR of the estimate (5). A treatment where all GMM parameters are achieved by maximizing the PSNR of (5) would be more eligible. This consistent approach would, however, be an extremely difficult problem.

- The estimator (5) has to be run online, but it has a high computational complexity, which to a large extent depends on the quadratic forms in the exponents of the Gaussian distributions $f^{(m)}(y)$ in (6).
- A vector $z$ with 64 dimensions was employed in [23]. It would be desirable to increase the number of dimensions of $y$ substantially, but this would be computationally demanding because of the matrix operations, both for GMM parameter estimation, and for online pixel estimation by means of (5).

To summarize, the benefit of a simplification of the estimator (5) is threefold: estimator parameters may be obtained by PSNR maximization, online complexity is reduced, and error concealment can be based on more surrounding pixels. In what follows, we will conceive another strategy for mixture-based spatio-temporal error concealment with this in mind.

### B. Mixture-Model- and LS-Based Scheme

We introduce an estimator that is heavily inspired by the solution in (5). The means $\mu_x^{(m)}$ and $\mu_y^{(m)}$, $m = 1, \ldots, M$, are removed, the matrices $C_{xy}^{(m)}(C_{yy}^{(m)})^{-1}$ are replaced by matrices $A^{(m)}$, and the *posteriori* weights $\phi^{(m)}$ are replaced by simpler functions $\pi^{(m)}$. Two subsets of the elements of $y$ are organized into vectors and employed in the estimator: the vector $y_{\mathrm{C}}$ is used for classification in the *posteriori* weights (C stands for classification), and the vector $y_{\mathrm{P}}$ is used for prediction (P stands for prediction). By introducing $y_{\mathrm{C}}$ and $y_{\mathrm{P}}$, more pixels may be used in the new simple *posteriori* weights $\pi^{(m)}(y_{\mathrm{C}})$ than with the prediction. The proposed estimator is thus parametrized by

$$\hat{x}(y, \theta) = \sum_{m=1}^{M} \pi^{(m)}(y_{\mathrm{C}})\, A^{(m)} y_{\mathrm{P}} \quad (7)$$

$$\pi^{(m)}(y_{\mathrm{C}}) = \frac{\rho^{(m)} h^{(m)}(y_{\mathrm{C}})}{\sum_{k=1}^{M} \rho^{(k)} h^{(k)}(y_{\mathrm{C}})} \quad (8)$$

$$h^{(m)}(y_{\mathrm{C}}) = \exp\left(-c^{(m)} \frac{\left\| y_{\mathrm{C}}^{(m),1} - y_{\mathrm{C}}^{(m),2} \right\|_2^2}{D^{(m)}}\right) \quad (9)$$

where $y_{\mathrm{C}}^{(m),1}$ and $y_{\mathrm{C}}^{(m),2}$ are vectors containing elements of $y_{\mathrm{C}}$, $c^{(m)} > 0$ is scalar, $D^{(m)}$ is the dimension of the vectors $y_{\mathrm{C}}^{(m),1}$ and $y_{\mathrm{C}}^{(m),2}$, and

$$\theta = \left\{\rho^{(1)}, \ldots, \rho^{(M)}, c^{(1)}, \ldots, c^{(M)}, A^{(1)}, \ldots, A^{(M)}\right\}. \quad (10)$$

In the discussion of the GMM-based estimator, we concluded that different *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 (9) incorporates the difference of the values of many spatially neighboring pixels. In the same way, in order to generate a mixture component that focuses on temporal correlation, $y^{(m),1}$ and $y^{(m),2}$ should be chosen so that the exponent of (9) incorporates the difference of the values of many temporally neighboring pixels. A more detailed description of different choices of the vectors $y_{\mathrm{C}}^{(m),1}$ and $y_{\mathrm{C}}^{(m),2}$ will be given in the experiment section. Consequently, the new function (7) combines linear estimators by means of *posteriori* weights like the GMM-based estimator (5). Parameter estimates are provided by the LS approach

$$\hat{\theta} = \arg\min_{\theta} \mathrm{E}_{X,Y}\left(\|x - \hat{x}(y, \theta)\|_2^2\right). \quad (11)$$

In contrast to the parameters of the estimator (5), the parameters $\hat{\theta}$ are obtained so that the PSNR

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255^2}{\mathrm{E}_{\mathrm{X,Y}}\left(\|\mathrm{x} - \hat{\mathrm{x}}(\mathrm{y}, \theta)\|_2^2\right)}\right) \quad (12)$$

that is a standard measure of video quality [17], [18], [23] is maximized. The details of the parameter estimation is the topic of Section II.C. For summarizing, error concealment is conducted by

1. offline procurement of $\hat{\theta}$ using (11) and a video database;
2. online employment of the estimator $\hat{x}(y, \hat{\theta})$ for error concealment of $x$.

The proposed mixture-based approach obtains estimator parameters by PSNR maximization, reduces online complexity compared to (5), and may use a vector $y_{\mathrm{C}}$ with high number of dimensions while maintaining low computational complexity.

### C. Algorithm for Solving the LS Estimation Problem

Since the solving of the LS problem (11) with the proposed estimator (7) does not have a closed form solution, an iterative algorithm is now proposed.

*1) The Matrices $A^{(m)}$:* We minimize the mean squared error (MSE) as a function of one matrix $A^{(r)}$ at a time, while keeping all other parameters, i.e., $A^{(m)}$, $m = 1, \ldots M, m \neq r$, as well as $\rho^{(m)}$ and $c^{(m)}$, $m = 1, \ldots M$, constant

$$A^{(r)*} = \arg\min_{A^{(r)}} \mathrm{E}\left[\|x - s(y, \theta)\|_2^2\right] \quad (13)$$

$$= \arg\min_{A^{(r)}} \mathrm{E}\left[\left\| A^{(r)} \pi^{(r)}(y_{\mathrm{C}}) y_{\mathrm{P}} - \left(x - \sum_{\mathrm{m=1, m \neq r}}^{\mathrm{M}} \pi^{(m)}(y_{\mathrm{C}}) A^{(m)} y_{\mathrm{P}}\right)\right\|_2^2\right]. \quad (14)$$

This is the standard linear MMSE problem, whose solution is [31]

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

$$R_1 = \mathrm{E}\left[\pi^{(r)}(y_C)xy_P^{\mathrm{T}}\right] - \sum_{m=1,m\neq r}^{M} A^{(m)}\mathrm{E}\left[\pi^{(r)(y_{\mathrm{C}})}\pi^{(m)(y_{\mathrm{C}})}y_{\mathrm{P}}y_{\mathrm{P}}^{\mathrm{T}}\right] \quad (16)$$

$$R_2 = \mathrm{E}\left[\left(\pi^{(r)}(y_{\mathrm{C}})\right)^2 y_{\mathrm{P}}y_{\mathrm{P}}^{\mathrm{T}}\right]. \quad (17)$$

For optimization in practice, the expectations are replaced by arithmetic means of realizations. The expectation evaluations may hence be computationally expensive if many realizations are used. Successive updates of the different matrices $r = 1,\ldots,M$ may, however, be calculated without evaluating the computationally expensive expectation operations in (16) and (17) more than once.

*2) The Parameters $\rho^{(m)}$ and $c^{(m)}$:* We first minimize the MSE as a function of the parameters $\rho^{(m)}$, $m = 1,\ldots,M$, while keeping all other parameters, i.e., $A^{(m)}$ and $c^{(m)}$, $m = 1,\ldots M$, constant. For each update, the MSE is compared for the parameter vector $\rho = [\rho^{(1)},\ldots,\rho^{(M)}]^{\mathrm{T}}$ obtained in the previous update, and for the two vectors $\rho \pm \alpha[\rho^{(1)}b(1),\ldots,\rho^{(M)}b(M)]^{\mathrm{T}}$, where $\alpha$ is some scalar, and $b(1)$ to $b(M)$ are the elements of a normalized random vector. The vector that yields the minimum value of the MSE is kept. This strategy is also employed for updating the parameters $c^{(m)}$ $m = 1,\ldots,M$.

To summarize, the updates of the parameters $A^{(m)}$, $\rho^{(m)}$ and $c^{(m)}$, all yield nonincreasing MSE. As the MSE decreases, the PSNR defined in (12) increases, and the PSNR thus increases in every iteration. The algorithm is executed off-line, and does thus not affect the on-line computational complexity.

## III. Experiments

In this section, the proposed method is simulated and compared to methods suggested by other authors. Details of the simulations are given in Section III-A, and results of the experiments are presented in Section III-B.

### A. Simulation Prerequisites

The listed conditions are chosen to fit state-of-the-art block-based video coders, and are impartial to all the compared schemes:

*Coder:* We focus on the predictively coded frames (an application of the proposed method to restoration of intra-coded frames is completely analogous). Motion vectors are calculated for $8 \times 8$-blocks. A search for a motion vector is performed by checking every integer displacement vector $(\Delta u, \Delta v)$ where $-8 \leq \Delta u, \Delta v \leq 8$. Each row of $16 \times 16$-blocks is divided into $8 \times 8$-blocks, that are interleaved into two packets as in [17], see Fig. 1.

*Motion Vectors for Error Concealment:* The error concealment schemes are evaluated in the case of correctly received motion vectors that are protected in a high priority layer, and in the case of lost motion vectors that are estimated by the median of the motion vectors of the available neighboring blocks [10].

*Video Data:* We use the luminance component of 124 MPEG-1 movies from [32] that have a frame rate of 29.97 frames per second and an image size of $352 \times 240$ pixels. The movies are divided into two independent sets, one for offline parameter optimization, 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. For visual comparison, a test movie from [33] is used.

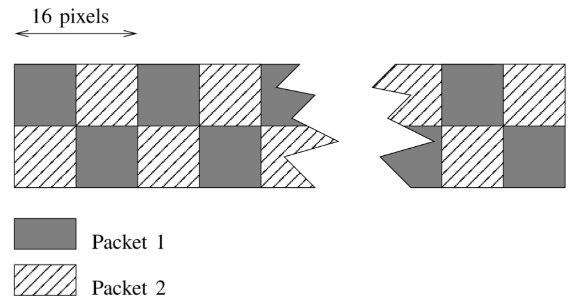


Fig. 1. Block interleaving. One row of $16 \times 16$-blocks is separated into two packets.

The proposed estimator uses the following setting:

*Pixel Employment:* Each lost $8 \times 8$-block is repaired by splitting it into four $4 \times 4$-blocks whose pixels are represented by a vector $x$, see Fig. 2. The vector $x$ is estimated using vectors of available surrounding pixels $y_{\mathrm{C}}$ and $y_{\mathrm{P}}$, cf. (7) and Fig. 2. We choose to work with few mixture components since we strive for low online 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 to the north and to the west in $y_{\mathrm{C}}$ 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 neighbors in the time direction in $y_{\mathrm{C}}$ are included in the exponent. A mixture with $M = 5$ components that focus on pixel differences in the south-west, west, north-west, north, and time directions is also investigated. Likewise, a mixture with $M = 9$ components that focus on pixel differences in the south-south-west, south-west, west-south-west, west, west-north-west, north-west, north-north-west, north, and time direction is examined, as well.

Surrounding pixels may not be available, because the block to be estimated is close to a border, or because several neighboring blocks are lost. Separate estimators (7) are obtained and stored for four cases, namely the case where all pixels in $Y_{\mathrm{C}}$ in Fig. 2 are present, the case where all pixels in $Y_{\mathrm{C}}$ over the lost block in frame $t$ and over the motion-compensated position of the lost block in frame $t-1$ are present, the case where all pixels in $Y_{\mathrm{C}}$ to the left of the lost block in frame $t$ and to the left of the motion-compensated position of the lost block in frame $t-1$ are present, and the case where only the part of $y_{\mathrm{P}}$ that is in frame $t-1$ in Fig. 2 is present. In the last case, there is no $y_{\mathrm{C}}$ for classification, and, therefore, the estimator in (7) is reduced to the linear LS estimator by setting $M = 1$.

*Offline Parameter Estimation:* The parameters $\rho_m$ and $c_m$, $m = 1,\ldots,M$ are initialized by setting them equal to 1, and the parameters $A^{(m)}$ are initialized by the linear MMSE estimator that is obtained by setting $M = 1$ in (7) and solving (11).

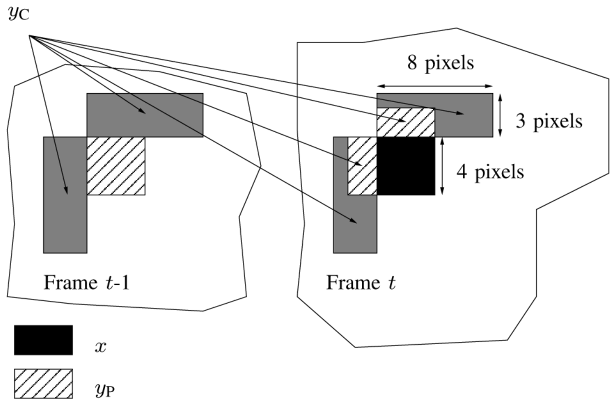


Fig. 2. Illustration of variables to be used with the proposed estimator. Lost blocks of size $8 \times 8$ are divided into four $4 \times 4$-blocks $x$ and estimated separately. The vector $x$ is estimated using vectors of available surrounding pixels $y_C$ and $y_P$.

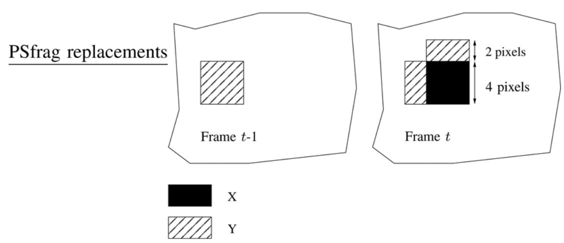


Fig. 3. Illustration of variables to be used with the GMM-based estimator as in [23]. Lost blocks of size $8 \times 8$ are divided into four $4 \times 4$-blocks $x$ that are estimated separately. The vector $x$ is estimated using a vector of surrounding pixels $y$.

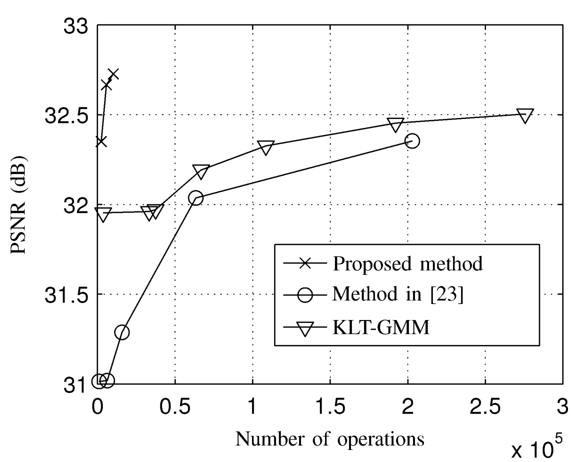


Fig. 4. Performance in PSNR, and computational complexity in terms of the number of online floating point operations, is visualized for the proposed method, GMM, i.e., the method in [23], and KLT-GMM. All pixels surrounding the lost block are received, and the motion vectors are lost and estimated on the receiver side. From left to right, the number of mixture components $M$ is 2, 5, and 9 for the proposed method, 1, 2, 5, 20, 64 for GMM, and 1, 2, 3, 10, 20, 40, 60 for KLT-GMM.

For the update of $\rho^{(m)}$ and $c^{(m)}$, we choose $\alpha = 0.1$. In each update, 1500 000 realizations of $z^{\mathrm{T}} = [x^{\mathrm{T}}, y^{\mathrm{T}}]$ are used. In each of 10 first iterations, ten updates are performed for $\rho_m$ and $c_m$, respectively, as well as one update per $A^{(m)}$. In ten final iterations, only the $A^{(m)}$ are updated. Separate parameter estimates $\hat{\theta}$ are obtained for the case of motion vectors that are protected in a high priority layer, and the case where lost motion vectors

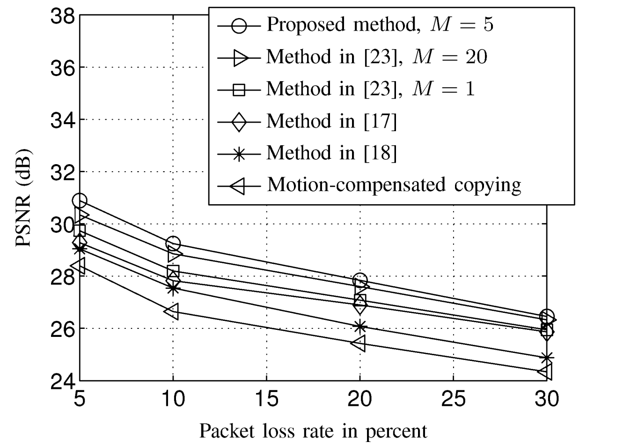


Fig. 5. Comparisons of different error concealment schemes in the case when the motion vectors are unavailable and replaced by the median of the motion vectors of the available surrounding blocks.

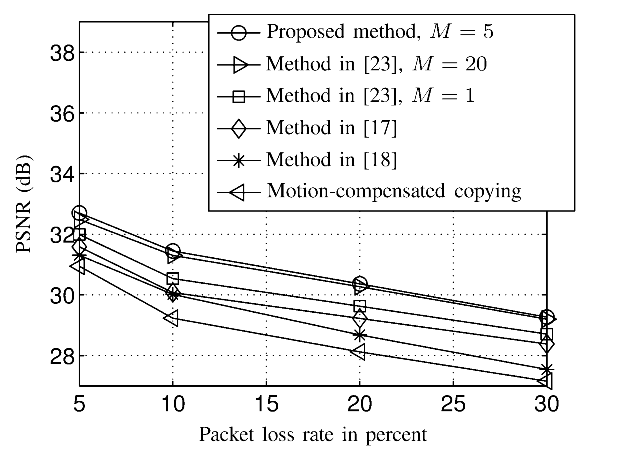


Fig. 6. Comparisons of different error concealment schemes in the case when the motion vectors are available.

are estimated by the median of the motion vectors of the available neighboring blocks.

### B. Results

Two sets of experiments are conducted. First, we investigate the performance in PSNR versus the computational complexity of different mixture-based estimator solutions to the error concealment problem. Thereafter, our solution is compared to general state-of-the-art error concealment schemes.

*Mixture-Based Estimator Comparison:* The purpose of this first experiment is to see that our estimator strategy yields higher performance in PSNR for the same computational complexity, as well as lower computational complexity for the same performance in PSNR, compared to the GMM-based estimator [23]. Moreover, we suggest a new version of the method in [23], namely a GMM where the Gaussian components have diagonal covariance matrices, that is trained using Karhunen Loéve transform (KLT)-rotated vectors.

The GMM uses the same information as in [23] for estimating $x$, namely $y$ as illustrated in Fig. 3. The KLT-GMM algorithm uses the union of $y_C$ and $y_P$ seen in Fig. 2 for forming an estimate, which is considerably more information than what is used

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 7. Visual comparison of restorations of a movie frame from [33] by means of the different error concealment methods in the case of a previous frame without errors, and lost motion vectors that are estimated by the median of the motion vectors of the neighboring blocks. (a) Original whole frame; (b) whole frame with error pattern; (c) motion-compensated copying of frame detail that is the woman's right arm; (d) method in [18]; (e) method in [17]; (f) method in [23] with $M = 1$; (g) method in [17] with $M = 20$; (h) proposed method with $M = 5$; (i) proposed method with $M = 5$ applied to whole frame.

in the GMM approach. All pixels surrounding the loss are received, and the motion vectors are lost and estimated on the receiver side. For evaluation in this experiment, 500 000 randomly drawn realization vectors $z^{\mathrm{T}} = [x^{\mathrm{T}}, y^{\mathrm{T}}]$ from the evaluation movies were used.

Fig. 4 shows performance in PSNR and computational complexity in terms of the number of online floating point operations, for the proposed method, the method in [23], and KLT-GMM. The proposed method clearly has the best tradeoff between performance in PSNR and computational complexity: For example, the proposed method with $M = 5$ mixture components increases the performance in PSNR by 1.6 dB compared to the method in [23] operating at the same complexity. Also, the proposed method with $M = 5$ increases performance by 0.3 dB compared to the method in [23], at 35 times lower computational complexity. In the case when the motion vectors are received, the PSNR values are higher, but the proposed method still has the best tradeoff between performance in PSNR and computational complexity.

*Error Concealment Comparison:* The proposed scheme is compared to previously proposed methods that mix spatial and temporal information given the motion vectors: the method in [18], the method in [17], and GMM-based error concealment [23], as well as to motion compensated copying [10]. In the experiments, the packets are assigned as lost in an independent and identically distributed random manner, and pixels surrounding a lost $8 \times 8$-block are not guaranteed to be available. The errors are distributed and propagate in a few tens of frames in each movie. Simulations are run for packet loss probabilities ranging from 0.05 to 0.3.

Fig. 5 presents the results for the case when the motion vectors are unavailable and replaced by the median of the motion vectors of the available surrounding blocks. In Fig. 6, we see the results in the case when the motion vectors are available. The proposed method gives best performance in PSNR in all tested cases. A GMM-based method with $M = 20$ gives a comparable result, but this comes at a cost of 11 times higher online computational complexity than the proposed method.

Visual performance is compared in Fig. 7(a)–(i). Fig. 7(a) shows the whole original frame and Fig. 7(b) shows the applied error pattern. Fig. 7(c)–(h) shows error concealment results for the woman's right arm obtained by motion compensated copying [10], the method in [18], the method in [17], the method in [23] with $M = 1$ and $M = 20$, and the proposed method with $M = 5$ respectively. The whole frame restored by the proposed method with $M = 5$ is shown in Fig. 7(i).

## IV. Conclusion

In this paper, a mixture model- and LS-based estimation technique for solving the packet video error concealment problem is presented. The proposed scheme improves performance in PSNR compared to other state-of-the-art error concealment schemes, for a wide range of stationary packet loss probabilities. For the same computational complexity, the proposed mixture-based method outperforms a previously proposed GMM-based technique in terms of PSNR. Also, for the same performance in PSNR, the proposed scheme has considerably lower computational complexity than the GMM approach. Other more sophisticated mixture-based estimators and parameter estimation techniques may yield further improvements and constitute interesting extensions of this work.

## References


[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[2] Y. Wang, Q.-F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

[3] W. Zhu, Y. Wang, and Q.-F. Zhu, "Second-order derivative-based smoothness measure for error concealment in DCT-based codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 713–718, Oct. 1998.

[4] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 470–477, Apr. 1995.

[5] J. Park, D.-C. Park, I. Marks, R. J. , and M. El-Sharkawi, "Recovery of image blocks using the method of alternating projections," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 461–474, Apr. 2005.

[6] O. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-Part I: Theory," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 539–554, Mar. 2006.

[7] O. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-Part II: Adaptive algorithms," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 555–571, Mar. 2006.

[8] P. Salama, N. Shroff, and E. Delp, "Error concealment in encoded video streams," in *Signal Recovery Techniques for Image and Video Compression and Transmission*, N. P. Galatsanos and A. K. Katsaggelos, Eds. Boston, MA: Kluwer, 1998 [Online]. Available: http://citeseer.ist.psu.edu/108531.html, [Online]. Available

[9] Y.-K. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26l test model," in *Proc. Int. Conf. Image Processing*, May 2002, vol. 2, pp. 729–732.

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

[11] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. ICASSP*, Apr. 1993, pp. 417–420.

[12] Y. Zhang and K.-K. Ma, "Error concealment for video transmission with dual multiscale Markov random field modeling," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 236–242, Feb. 2003.

[13] J. Zheng and L.-P. Chau, "Efficient motion vector recovery algorithm for H.264 based on a polynomial model," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 507–513, Jun. 2005.

[14] Y.-L. Huang and H.-Y. Lien, "Temporal error concealment for MPEG coded video using a self-organizing map," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 676–681, May 2006.

[15] S. Garg and S. Merchant, "Interpolated candidate motion vectors for boundary matching error concealment technique in video," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 10, pp. 1039–1043, Oct. 2006.

[16] M.-C. Hwang, J.-H. Kim, D. T. Duong, and S.-J. Ko, "Hybrid temporal error concealment methods for block-based compressed video transmission," *IEEE Trans. Broadcasting*, vol. 54, no. 2, pp. 198–207, Jun. 2008.

[17] 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, no. 3, pp. 248–258, Jun. 1993.

[18] 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, no. 6, pp. 1122–1128, Jun. 2000.

[19] D. Agrafiotis, D. Bull, and C. Canagarajah, "Enhanced error concealment with mode selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 8, pp. 960–973, Aug. 2006.

[20] L. Tang, "Combined and iterative form of spatial and temporal error concealment for video signals," *IEEE Trans. Broadcasting*, vol. 52, no. 3, pp. 356–361, Sep. 2006.

[21] Y. Chen, Y. Hu, O. Au, H. Li, and C. W. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," *IEEE Trans. Multimedia*, vol. 10, no. 1, pp. 2–15, Jan. 2008.

[22] D. Turaga and T. Chen, "Model-based error concealment for wireless video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 483–495, Jun. 2002.

[23] D. Persson, T. Eriksson, and P. Hedelin, "Packet video error concealment with Gaussian mixture models," *IEEE Trans. Image Process.*, vol. 17, no. 2, pp. 145–154, Feb. 2008.

[24] J. Zhang and D. Ma, "Nonlinear prediction for Gaussian mixture image models," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 836–847, Jun. 2004.

[25] K. Popat and R. Picard, "Cluster-based probability model and its application to image and texture processing," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 268–284, Feb. 1997.

[26] R. Martin, C. Hoelper, and I. Wittke, "Estimation of missing LSF parameters using Gaussian mixture models," in *Proc. IEEE ICASSP*, May 2001, vol. 2, pp. 729–732.

[27] J. Lindblom and J. Samuelsson, "Bounded support Gaussian mixture modeling of speech spectra," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 1, pp. 88–99, Jan. 2003.

[28] D. Persson and T. Eriksson, "A minimum mean square error estimation and mixture-based approach to packet video error concealment," in *Proc. IEEE ICASSP*, Apr. 2007, vol. 1, pp. 749–752.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[30] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, pp. 195–239, 1984.

[31] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[32] *The Prelinger Archives*, [Online]. Available: http://www.archive.org/details/prelinger

[33] *Xiph.Org Foundation*, [Online]. Available: http://downloads.xiph.org/websites/media.xiph.org/video/derf/



**Daniel Persson** was born in Halmstad, Sweden, in 1977. He graduated from Ecole Polytechnique, Paris, France, in 2002, received the M.Sc. degree in engineering physics from the Chalmers University of Technology, Göteborg, Sweden, in 2002, and the Ph.D. degree in electrical engineering from the Chalmers University of Technology in 2009.

He is currently a postdoctoral researcher in the Communication Systems Group, Linköping University, Sweden. His research interest is joint source-channel coding.



**Thomas Eriksson** was born in Skövde, Sweden, on April 7, 1964. He received the M.Sc. degree in electrical engineering and the Ph.D. degree in information theory from the Chalmers University of Technology, Göteborg, Sweden, in 1990 and 1996, respectively.

He was with AT&T Labs-Research from 1997 to 1998, and in 1998 and 1999, he was working on a joint research project with the Royal Institute of Technology and Ericsson Radio Systems AB. Since 1999, he has been an Associate Professor at the Chalmers University of Technology, and his research interests include vector quantization, speaker recognition, and system modeling of nonideal hardware.