A STATISTICAL APPROACH TO PACKET LOSS CONCEALMENT FOR VIDEO

Daniel Persson and Per Hedelin

Chalmers University of Technology
Department of Electrical Engineering
412 96 Göteborg
Sweden

ABSTRACT

We have developed a statistical prediction technique to compensate for lost pixel blocks during real time transmission of video over packet-switched networks. The approach could as well be used for standard inter-frame coding. The method is based on the joint modeling of pixel statistics by Gaussian mixtures. Different EM variants for decreasing computational complexity and storage space are proposed. In the case of 4×4 luminance blocks, our method increases performance by 2.3 dB when augmenting the number of mixture components from 1 to 64. We show that these results are statistically significant.

1. INTRODUCTION

Video communication over the Internet increases in popularity. The problem of delivery of a Quality of Service in packet-switched networks is treated at various levels throughout the ATM and IP protocol stacks [1, pp 579–594]. As of today, Intserv and Diffserv models have not yet found their way out to the public. Though one may reasonably expect a continuation of the improvement of the physical layer, an increase in access possibilities is usually followed by an increase in demand. Under such circumstances, there will be a continued need for error concealment. Another area of application of error concealment techniques is the compensation for frame erasures in cellular networks.

In [2, 3], error concealment is treated in conjunction with neural network based prediction and end user quality. We have developed an interpolation scheme based on Gaussian mixtures to compensate for lost blocks. For the evaluation of our scheme, motion vectors are recalculated at the receiver end. This means that our method will work even under severe conditions when motion vectors, encoded with high redundancy, are lost. This also means that the method is able to compensate lost I and P frames as well as B frames. The system can be used for both uni- and bi-directional interpolation. In [4], it has been shown that classification of pixels in lost blocks reduces blocking artifacts when interpolating dropped frames in very-low-bit-rate coding. Our model-based predictor can be seen as a soft classifier that tries to understand the current situation in order to make the best possible proposal for the replacement of the lost block. The EM algorithm discussed in [5] has been widely used on statistical problems throughout many disciplines. In the field of communication, it has been applied for fitting a Gaussian basis in order to improve prediction [6], quantizing [7] and classification [8].

In this paper, we focus on the statistics of the training of the Gaussian mixture model for 4×4 blocks. Our idea is to replace larger lost blocks by a combination of smaller blocks in the case of a real scenario. By the usage of small blocks, computational complexity is decreased. Using spatial information from the surrounding of each 4×4-block helps avoiding blocky artifacts in the larger lost blocks. An advantageous feature of our predictor is that it can provide a replacement for varying available contexts at the receiver side.

This paper is organized as follows. In section 2, our method is described. Two versions of the EM algorithm are presented in section 3. Section 4 gives the numerical values of the parameters that are used in the simulations. The results are presented in section 5. The paper is concluded in section 6.

2. METHOD

2.1. Modeling the source

Let x denote the pixel luminance values of a lost block and let y denote a context to the loss that is still available at the receiver. Given these preliminaries, one may pose the problem of error concealment as to find a predictor

$$\hat{x} = E[x|y]. \tag{1}$$

We model the statistical dependencies between x and y as a Gaussian mixture, that is a weighted sum of M Gaussian densities,

$$f_{\text{GM}}(z) = \sum_{m=1}^{M} \rho_m f_m(z|\mu_m, C_m) \tag{2}$$

where $z^T = (x^T, y^T)$, the $f_m(z|\mu_m, C_m)$ constitute a Gaussian basis and the ρ_m are the weights or a priori probabilities. The conditional pdf

$$f_{\text{GM}}(x|y) = \sum_{m=1}^{M} \pi_m(y) f_m(x|y, \mu_{x|y,m}, C_{x|y,m}) \tag{3}$$

where the $\pi_m(y)$ are

$$\pi_m(y) = \frac{\rho_m f_m(y|\mu_{y,m}, C_{yy,m})}{\sum_{l=1}^{M} \rho_l f_l(y|\mu_{y,l}, C_{yy,l})}. \tag{4}$$

is obtained from equation (2). One easily derives

$$\hat{x} = \sum_{m=1}^{M} \pi_m(y)(\mu_{x,m} + C_{xy,m} C_{yy,m}^{-1}(y - \mu_{y,m})). \tag{5}$$

One advantage of this approach is that we are free to form x and y in any way from z. This is valuable in the case of error bursts where there is a high probability of several missing neighboring blocks. If only a sub-set $\tilde{y} \in y$ is available at the receiver, we form a new predictor

$$\tilde{\hat{x}} = E[x|\tilde{y}]. \tag{6}$$

The EM algorithm is used to optimize the model in equation (2) for a given database. The standard EM update-equations are

$$\rho_m^{i+1} = \frac{1}{N}\sum_{n=1}^{N}\gamma_{n,m}^i \tag{7}$$

$$\mu_m^{i+1} = \frac{1}{N\rho_m^{i+1}}\sum_{n=1}^{N}\gamma_{n,m}^i z_n \tag{8}$$

$$C_m^{i+1} = \frac{1}{N\rho_m^{i+1}}\sum_{n=1}^{N}\gamma_{n,m}^i (z_n-\mu_m^{i+1})(z_n-\mu_m^{i+1})^T \tag{9}$$

where z_n denote the N training vectors and i is the iteration count. The a posteriori probabilities $\gamma_{n,m}^i$ are

$$\gamma_{n,m}^i = \frac{\rho_m^i f_m^i(z_n|\mu_m^i, C_m^i)}{\sum_{l=1}^{M}\rho_l^i f_l^i(z_n|\mu_l^i, C_l^i)}. \tag{10}$$

The computational complexity of the algorithm is approximately M determinants, M matrix inversions and $3MND^2$ additions and multiplications per iteration where D is the number of elements in a vector z_n. The form of a more exact expression for the complexity is dependent on the values of M, N and D. The value of the above expression can be made smaller by observing that we deal with symmetric matrixes. The training is performed off-line.

2.2. Organization of pixels into vectors

We work on the luminance components of the pixels. The vector z in equation (2) contains the pixel luminances of a square block z^t of side length Δ and its spatio-temporal surroundings. The reader should distinguish the index t which signifies time from the index n that enumerates the vectors in the database. The situation is depicted in figure 1. The past motion vector is calculated by finding the best match in frame $t-2$ to the block in frame $t-1$ that is situated at the same spatial position as the lost block. The future motion vector is calculated by finding the best match in frame $t+2$ to the block in frame $t+1$ that is situated at the same spatial position as the lost block. This means that our scheme is independent of motion vectors calculated at the transmitter end. Two frames have to be buffered in order to utilize motion-compensated future information. We set

$$x = z^t \tag{11}$$

$$y = \begin{pmatrix} z^{t-1} \\ z^{t+1} \\ z_{\mathcal{F}}^t \end{pmatrix}. \tag{12}$$

2.3. Evaluation

The evaluation is done using PSNR and the log-likelihood for the complete data z. The log-likelihood can be related to an upper bound on the entropy and is evaluated as

$$L(\Theta) = \frac{1}{ND}\sum_{n=1}^{N}\log_2\Big(\sum_{m=1}^{M}\rho_m f_m(z_n|\mu_m, C_m)\Big) \tag{13}$$

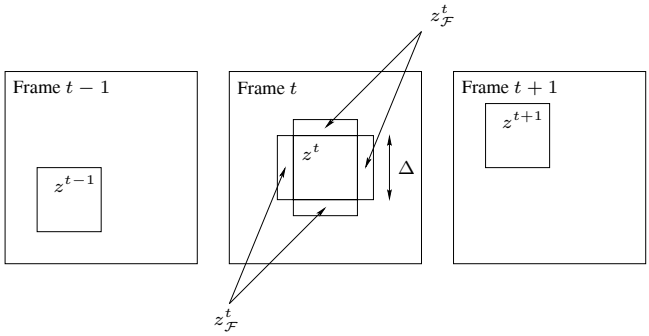

Fig. 1. Organization of pixels into vectors.

where D is the number of elements in z and $\Theta = \{\mu_m, C_m\}$.

3. TWO VARIATIONS ON THE APPROACH

3.1. Karhunen-Loéve diagonalization

Neighboring video-pixels are highly correlated and tend to have the same value. This means that the source has the form of a hyper-ellipsoid in color space. We can exploit this feature by applying the Karhunen-Loéve transform

$$z' = Vz \tag{14}$$

where V is obtained by use of the conventional EM-algorithm with $M = 1$. We may now approximate that all of the elements outside the diagonals of the covariance matrices are zero and hereby decrease the number of variables to be estimated in the covariance matrixes from D^2 to D. Storage space is also saved. Through the diagonalization, the mixture components are allowed to adjust to local correlation by scaling but not by rotation. Lower performance is therefore expected in comparison with the standard method. The predictor (5) can now be used with

$$\mu_m = V\mu' \tag{15}$$

$$C_m = VC'_mV^T \tag{16}$$

where μ'_m and C'_m are the means and covariances obtained through training on the transformed data.

Another possible approach would be to use the DCT transform in order to decorrelate the components of z. The advantage of the DCT-approach is that no storage of the eigenvectors is needed. The decisive disadvantage of this method is though that it does not handle temporal correlation in a straight-forward way.

3.2. Deflection of frequent vectors

Another way to modify the EM-algorithm to make profit of the form of the source is to randomly deflect vectors that occur more often. In this way, given a fixed number N of processed vectors z_n, the EM-algorithm sees more of the distribution during each iteration. This might be valuable in a real application where an extensive database should be used for the training. We have selected the criterion to accept each vector with a probability

$$p(z) = \alpha\frac{d(z)^2+\beta_1}{d(z)^2+\beta_2} \tag{17}$$

where $\alpha > 0$ and $0 < \beta_1 < \beta_2$ are arbitrary constants and $d(z)$ is the scaled Euclidean distance from z to its projection on $\frac{1}{\sqrt{D}}(1, ..., 1)$

$$d(z) = \frac{\|z - \breve{z}\|_2}{\sqrt{D}}. \quad (18)$$

The vector $\breve{z}$ is defined as

$$\breve{z} = \frac{|z \cdot \frac{1}{\sqrt{D}}(1, ..., 1)|}{\sqrt{D}}(1, ..., 1). \quad (19)$$

A consequence of this approach is that the algorithm now sees a somewhat skewed distribution and we have to modify the equations in order to avoid this. The compensated update equations are

$$\begin{aligned}
\rho_m^{i+1} &= \frac{1}{\sum_{k=1}^{\tilde{N}} \frac{1}{p(z_k)}\phi_k(p(z_k))} \sum_{n=1}^{\tilde{N}} \gamma_{n,m}^i \phi_n(p(z_n)) \\
\mu_m^{i+1} &= \frac{1}{\rho_m^{i+1} \sum_{k=1}^{\tilde{N}} \frac{1}{p(z_k)}\phi_k(p(z_k))} \sum_{n=1}^{\tilde{N}} \gamma_{n,m}^i \phi_n(p(z_n)) z_n \\
C_m^{i+1} &= \frac{1}{\rho_m^{i+1} \sum_{k=1}^{\tilde{N}} \frac{1}{p(z_k)}\phi_k(p(z_k))} \sum_{n=1}^{\tilde{N}} \gamma_{n,m}^i \phi_n(p(z_n)) \\
&\quad \times (z_n - \mu_m^{i+1})(z_n - \mu_m^{i+1})^T \quad (20)
\end{aligned}$$

where $\tilde{N} > N$ is the number of scanned vectors in the database out of which N vectors are are extracted for the training. The index function $\phi_n(p(z_n))$ assumes the values 0 or 1 depending on whether z_n is chosen for the training or not. The parameter $\gamma_{n,m}^i$ is

$$\gamma_{n,m}^i = \frac{\rho_m^i f_m^i(z_n|\mu_m^i, C_m^i)}{p(z_n) \sum_{l=1}^{M} \rho_l^i f_l^i(z_n|\mu_l^i, C_l^i)}. \quad (21)$$

4. SIMULATION PREREQUISITES

The search for the motion vectors is performed within a square block of side length 17. The size of the lost block $\Delta = 4$.

The database is generated from 124 colour and B&W MPEG1 movies having a framerate of 29.97 frames per second and an image size of 352×240 pixels. The movies are all taken from [9]. These movies are randomly divided into two sets. One of these sets consists of 35 movies and is used to extract the model. The other set consists of 89 movies and is used for the evaluation. In this way, our evaluation is open in the strict sense. From each of the two sets described above, vectors are drawn in a uniformly random manner and in such a way that no two vectors coincide. In this way, a training and an evaluation database consisting of 1470 000 and 480 000 vectors respectively are built.

In the training, 147 000 randomly drawn vectors from the training database are used during each iteration. In the evaluation, the whole evaluation database is used. Ten EM iterations are run before each evaluation. In all our tests, this number of iterations has been enough to reach convergence. The algorithm is initiated by estimates of the mean and covariance of the source. An individual covariance matrix for each mixture component is created by adding a small random number to each of the diagonal elements of the estimate of the covariance matrix of the source.

For the deflection of vectors described in section 3.2, $M = 16$, α is chosen in order to extract a certain number of vectors during one scan through the database, $\beta_1 = 1$ and $\beta_2 = \frac{64^2}{3}$.

Method	Lower bound	Upper bound
Simple mean interpolation	29.77	29.91
$M = 1$	31.93	32.03
$M = 64$	34.23	34.35

Table 1. Bounds within which the PSNR lies with 98% probability.

5. RESULTS

In figure 2, we see how the log-likelihood increases as a function of the number of mixture components. Figure 3 depicts the increase in PSNR as a function of the number of mixture components. In addition to the PSNR provided by the full model, the PSNR curve supplied by the mixture component $f_m(z|\mu_m, C_m)$ having the largest $\gamma_m(y)$ is shown. Through using only this component, $M - 1$ matrix multiplications in equation (5) can be avoided. By augmenting M from 1 to 64, we increase PSNR by 2.3 dB. The computational complexity increases linearly with the number of mixtures M. Simple mean linear interpolation of the motion compensated information in z^{t-1} and z^{t+1} is used as a benchmark. This method gives a PSNR of 29.8 dB. Thus, our method increases PSNR by 4.5 dB. It should though be noted that the simple mean linear interpolation does not take the spatial surrounding $z_{\mathcal{F}}^t$ into account. This is an important cause for the difference in PSNR.

Significance tests have been performed assuming that the square Euclidean norm of the difference between x and $\hat{x}$ is distributed according to a normal distribution with parameters that we can estimate with good precision. The bounds on the PSNR that are seen in table 1 tell us that the result is 0.98-significant.

In figure 4, the Karhunen-Loéve diagonalization approach and the standard EM are compared. The PSNR saturates faster using the Karhunen-Loéve diagonalization. This is in agreement with our expectations.

As was already pointed out in the section 3.2, the method of deflection of frequent vectors is valuable when we want the algorithm to get as much information as possible out of a comprehensive database, without necessarily feeding the algorithm the whole content of the database. In order to investigate this scenario, the PSNR is plotted as a function of vectors per iteration in figure 5. As can be seen from the picture, the deflection-approach tends to perform better in comparison to standard EM in trainings where a smaller fraction of the database is used to fit the Gaussians. One might expect that the trainings with less vectors are subject to larger variations in performance. Therefore, we have made sure that these simulations reproduced the same results a number of times.

Finally in figure 6, we compare what happens if we divide the 4×4 block into four 2×2 blocks and compensate these on an individual basis. One should note that the same amount of information has been used by both replacement strategies, but the joint effort performs better.

6. CONCLUSION

In this paper, a Gaussian mixture based approach to concealment of lost pixel blocks during transmission of video is explored. We find that Gaussian mixtures increase performance. Variants of the algorithm that speed up the training procedure and decrease stor-

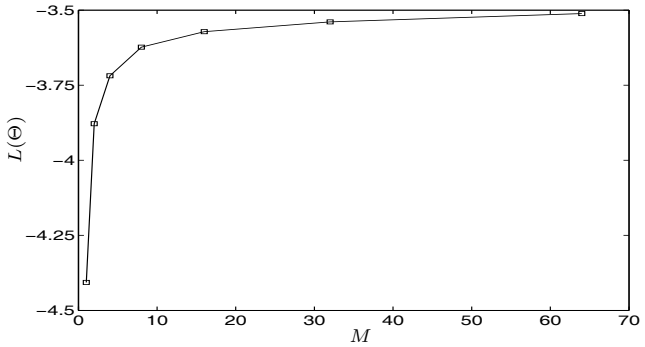

Fig. 2. Standard EM. Log-likelihood as a function of the number of mixture components.

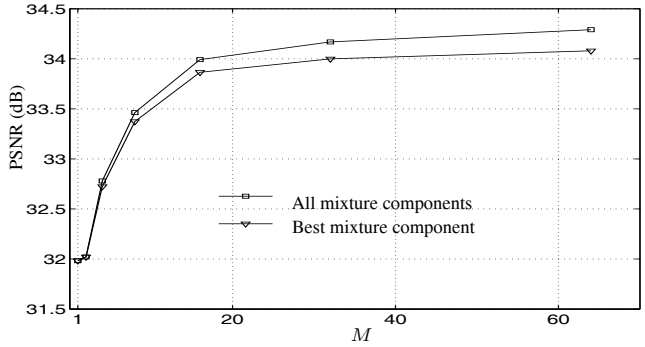

Fig. 3. Standard EM. PSNR as a function of the number of mixture components.

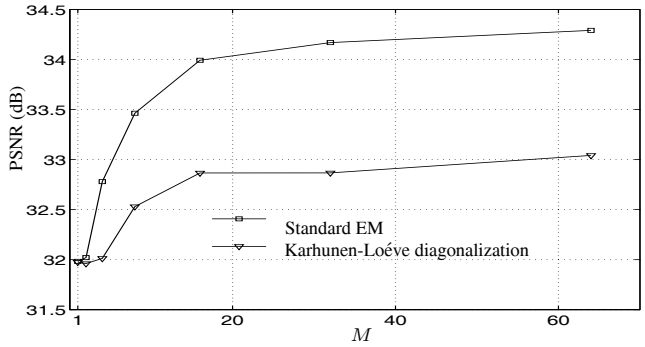

Fig. 4. Karhunen-Loéve diagonalization in comparison to standard EM. PSNR as a function of the number of mixture components.

age space are investigated. The importance of the spatial context for error concealment is observed. An investigation of how to handle error bursts giving rise to varying available contexts at the receiver end could be the topic of future research.

7. REFERENCES

[1] J. F. Kurose and K. W. Ross, *Computer Networking*, Addison-Wesley, 2002.

[2] C. E. Cramer and E. Gelenbe, "Video quality and traffic qos in learning-based subsampled and receiver-interpolated video sequences," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 2, pp. 150–167, Feb. 2000.

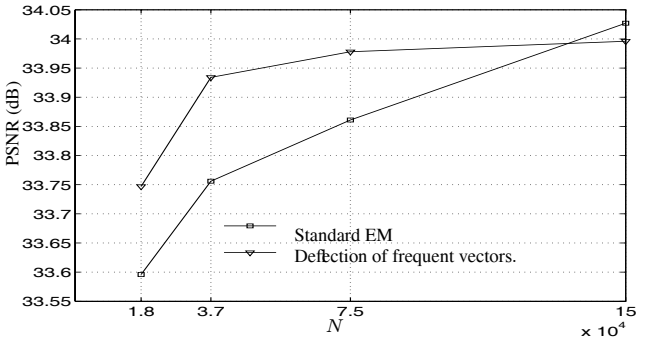

Fig. 5. Deflection of frequent vectors in comparison to standard EM. PSNR as a function of the number of vectors in the training.

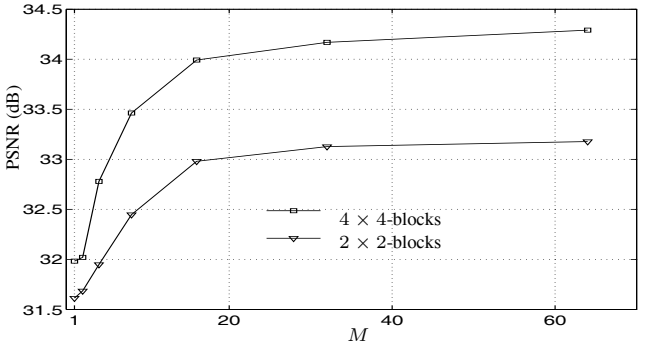

Fig. 6. Repair of lost 4×4-block by one 4×4-block and by four 2×2-blocks. PSNR as a function of the number of mixture components.

[3] G. Karlsson O. Verscheure, X. Garcia and J. Hubaux, "User-oriented qos in packet video delivery," *IEEE Network*, vol. 12, no. 6, pp. 12–21, Nov./Dec. 1998.

[4] O. C. Au C. Wong and C. Tang, "Modified motion compensated temporal frame interpolation for very low bit rate video," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1996, vol. 4, pp. 2327–2330.

[5] N. M. Laird A. P. Dempster and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of Royal Stat. Soc.*, vol. 39, pp. 1–38, 1977.

[6] J. Samuelsson and P. Hedelin, "Recursive coding of spectrum parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 492–503, Jul. 2001.

[7] P. Hedelin and J. Skoglund, "Vector quantization based on gaussian mixture models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 385–401, Jul. 2000.

[8] M. Aili and A. Nilsson, "Algorithms for head-steering of computer," M.S. thesis, Lund Institute of Technology, 2003.

[9] "Prelinger archives," www.archive.org/movies/prelinger.php, Online resource.