PREDICTING FACES IN VIDEO SEQUENCES USING EIGENSPACE UPDATE ALGORITHMS

Héctor J. Pérez-Iglesias and Adriana Dapena

Departamento de Electrónica y Sistemas, Facultad de Informática
Universidad de La Coruña, Campus de Elviña s/n, 15071, La Coruña, Spain
email: {hperez,adriana}@udc.es

ABSTRACT

A fundamental module in modern video coders is the *frame predictor* which provides the data needed to code frames from previous ones. In PCA-based predictors, the frames are represented as their projection in a proper basis (eigenspace) obtained from the convariance matrix.

In this paper, we investigate the performance of several algorithms in order to obtain an adequate eigenspace. Experiment results show that the best performance is obtained when the eigenspace is updated taking into account the non-stationary nature of face images. The technique offers a competitive alternative to P-predictive and B-predictive frames.

1. INTRODUCTION

Principal Component Analysis (PCA) is a well-known statistical processing technique that allows to reduce the redundancy of the input data by projecting the data over a proper basis. In particular, PCA has proved to be a powerful technique for representing a set of face images using a reduced number of coefficients. Each face image in the training set is represented exactly in terms of a linear combination of the eigenfaces calculated by identifying the eigenvectors of the covariance matrix.

Recently, Piqué and Torres [6] have also argued that PCA is a promising technique for coding faces in video sequences and offers a very competitive alternative to B-predictive frames. The idea is to predict the frame by calculating the projection into the eigenspace calculated from previous faces. The coefficients are therefore coded and transmitted. Full faces are only coded when a poor representation is obtained and, in this case, the eigenspace is updated using the algorithm proposed in [1].

We focus our attention on two aspects of the coding scheme proposed in [6]: the update algorithm and the compression degree. We will prove that the eigenspace update algorithm proposed in [5] is able to predict the frames by using less coefficients than the algorithm used in [6]. As a result, a higher compression is achieved when this algorithm is included in the face coding scheme.

This paper is structured as follows. Section 2 shows the video coding scheme and presents a short revision of the two eigenspace update algorithms considered in the paper. Section 3 provides the results of several simulations which allow to compare the performance of the update algorithms when they are used to predict faces in a video sequence. Finally, Section 4 is devoted to the conclusions.

This work has been supported by Ministerio de Ciencia y Tecnolog´ıa of Spain and FEDER funds from the European Union, under grant number TEC2004-06451-C05-01.

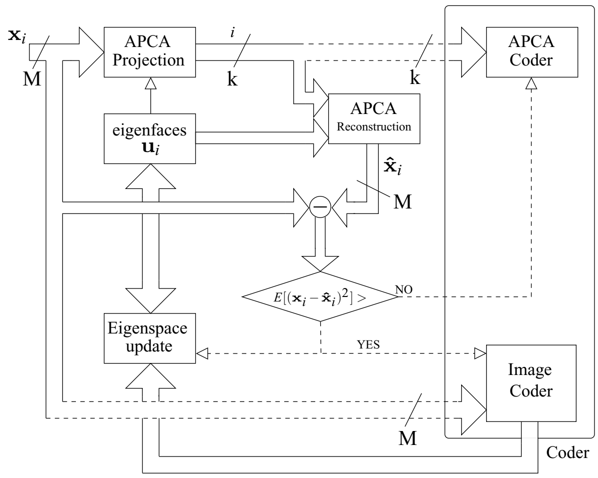


Figure 1: Face coding scheme.

2. FACE CODING SCHEME

We consider the video coding scheme shown in Figure 1 which is formed by a frame predictor followed by a image/coefficients coder. The 2-D image of M pixels is treated as an 1-D vector of dimension denoted by $\mathbf{x}_i$, $i = 1, 2, \ldots$ The first image, $\mathbf{x}_1$ is full coded and it is used to obtain the first eigenface, $\mathbf{u}_1$.

The other frames in the sequence, $\mathbf{x}_i$, $i = 2, \ldots$, are projected over the eigenspace formed by the k eigenvectors, $\mathbf{u}_1, \ldots, \mathbf{u}_k$, obtained from previous frames. That is, for the i-th image, we compute k coefficients given by

$$_{pi} = \mathbf{u}_p^T \mathbf{x}_i, \quad p = 1, 2, \ldots, k \tag{1}$$

Note that to evaluate the expression (1) requires M multiplications and $M-1$ additions, i.e., $k \times (M + (M-1))$ flops. In [6], the number k is equal to the number of eigenvectors computed from the data. However, it is well known that PCA packs the maximum average energy in few eigenvalues. For this reason, we propose to retain only the eigenvectors corresponding to the k largest eigenvalues. This retained eigenvectors are afterwards used to calculate (1).

In the next step, the image is reconstructed from the retained coefficients and the Mean Square Error (MSE) is calculated,

$$= E[(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2] \tag{2}$$

where $\hat{\mathbf{x}}_i$ is the predicted frame

$$\hat{\mathbf{x}}_i = \quad {}^{k}_{p=1} \quad {}_{pi}\mathbf{u}_p \tag{3}$$

If the frame has been predicted with enough quality ($\leq$), only the coefficients will be coded. On the contrary, when $>$, the image is coded as an intra-frame and the eigenspace is updated. The parameter is determined taking into account the desired PSNR (Peak Signal Noise Ratio).

Since the frames are dynamically captured, we are interested in updating the eigenfaces each time a new frame is acquired. Among the existing eigenspace update algorithms [1, 2, 4, 5], we have studied the performance of the algorithm proposed by Chandrasekaran, Manjunath, Wang, Winkler, and Zhang in [1] (henceforth CMWWZ) and the proposed by Liu, Chen, and Thornton [5] (henceforth LCT).

2.1 The CMWWZ algorithm

The CMWWZ algorithm computes the eigenspace by performing the Singular Value Decomposition (SVD) on the image matrix, instead of the covariance matrix. The set of images $\mathbf{X}_{i-1} = [\mathbf{x}_1, ..., \mathbf{x}_{i-1}]$ is represented as $\mathbf{X}_{i-1} = \mathbf{U}_{i-1}\mathbf{\Delta}_{i-1}\mathbf{V}_{i-1}^T$ where $\mathbf{\Delta}_{i-1}$ contains the eigenvalues, and $\mathbf{U}_{i-1}$ and $\mathbf{V}_{i-1}$ are the eigenvector matrices. For a new captured image $\mathbf{x}_i$, the SVD is recalculated

$$[\mathbf{U}_{i-1}\mathbf{\Delta}_{i-1}\mathbf{V}_{i-1}^T\mathbf{x}_i] = \mathbf{U}_i\mathbf{\Delta}_i\mathbf{V}_i^T \tag{4}$$

The size of the matrices can be reduced by taking only the largest eigenvalues and the corresponding eigenvectors. Figure 2 contains more details of the algorithm. Note also that we do not include the steps related with matrix $\mathbf{V}$ because it is not used in the face coder scheme presented in Section 2.

Given the frame $\mathbf{x}_i$, the eigenvector matrix $\mathbf{U}_c$ and the eigenvalues matrix $\mathbf{\Delta}_c$, *compute*

1. $\mathbf{y} \leftarrow \mathbf{U}_c^T\mathbf{x}_i$
2. $\mathbf{a}' \leftarrow \mathbf{x}_i - \mathbf{U}_c\mathbf{y}$
3. $\mathbf{a} \leftarrow \frac{\mathbf{a}'}{\|\mathbf{a}'\|}$
4. $\mathbf{A} \leftarrow \begin{bmatrix} \mathbf{\Delta}_c & \mathbf{y} \\ 0 & \mathbf{a}^T\mathbf{x}_i \end{bmatrix}$
5. Compute the eigenvectors, $\mathbf{U}'$, and eigenvalues, $\mathbf{\Delta}$, of $\mathbf{A}$.
6. $\mathbf{U} \leftarrow [\mathbf{U}_c \ \ \mathbf{a}]\mathbf{U}'$.
7. Let $\mathbf{U}_c$ equal the first k columns of $\mathbf{U}$.
8. Let $\mathbf{\Delta}_c$ equal the leading $k \times k$ principal submatrix of $\mathbf{\Delta}$.

Figure 2: The CMWWZ algorithm.

2.2 The LCT algorithm

The LCT algorithm is based on estimating the covariance matrix taking into account the variations on the first and second-order statistics of the image signals. The mean and the covariance are estimated using the following expressions

$$\begin{aligned} \hat{\mathbf{m}}_i &= {}_m\hat{\mathbf{m}}_{i-1} + (1 - {}_m)\mathbf{x}_i \\ \hat{\mathbf{C}}_i &= {}_v\hat{\mathbf{C}}_{i-1} + (1 - {}_v)(\mathbf{x}_i - \hat{\mathbf{m}}_i)(\mathbf{x}_i - \hat{\mathbf{m}}_i)^T \end{aligned} \tag{5}$$

where ${}_m$ and ${}_v$ are the decay parameters. Since the matrix $\hat{\mathbf{C}}_i$ is calculated at time i, we can perform the SVD to obtain the eigenvectors. Then, we keep only k eigenvectors. Note as well that the covariance matrix $\hat{\mathbf{C}}_{i-1}$ can be approximated by using the k retained eigenvectors (and eigenvalues) at time $i-1$, i.e.,

$$\begin{aligned} \hat{\mathbf{C}}_i &\approx {}_v\mathbf{U}_{i-1} \ {}_{i-1}\mathbf{U}_{i-1}^T + (1 - {}_v)(\mathbf{x}_i - \hat{\mathbf{m}}_i)(\mathbf{x}_i - \hat{\mathbf{m}}_i)^T \\ &= \mathbf{B}_i\mathbf{B}_i^T \end{aligned} \tag{6}$$

where $\mathbf{B} = [\sqrt{\ {}_v}\mathbf{U}_{i-1}\mathbf{\Delta}_{i-1}^{1/2} \quad \sqrt{1 - {}_v}(\mathbf{x}_i - \hat{m}_i)]$. The matrices $\mathbf{U}_{i-1}$ and $\mathbf{\Delta}_{i-1}$ contain, respectively, the k eigenvectors and eigenvalues retained at time $i-1$. Figure 3 summarizes the algorithm (the initialization step has not been included). Note that the last step is needed because we compute the eigenvectors of matrix $\mathbf{A} = \mathbf{B}_i^T\mathbf{B}_i$ instead of (6).

Given the frame $\mathbf{x}_i$, the eigenvector matrix $\mathbf{U}_c$, the eigenvalues matrix $\mathbf{\Delta}_c$ and the estimated mean $\mathbf{m}$ *compute*

- $\mathbf{m} \leftarrow {}_m\mathbf{m} + (1 - {}_m)\mathbf{x}_i$
- $\mathbf{B} \leftarrow [\sqrt{\ {}_v}\mathbf{U}_c\mathbf{\Delta}_c^{1/2} \quad \sqrt{1 - {}_v}(\mathbf{x}_i - \mathbf{m})]$
- $\mathbf{A} \leftarrow \mathbf{B}^T\mathbf{B}$
- Compute the eigenvectors, $\mathbf{U}'$, and eigenvalues, $\mathbf{\Delta}$, of $\mathbf{A}$.
- Let $\mathbf{U}_c'$ equal the first k columns of $\mathbf{U}'$.
- Let $\mathbf{\Delta}_c$ equal the leading $k \times k$ principal submatrix of $\mathbf{\Delta}$.
- $\mathbf{U}_c \leftarrow \mathbf{B}\mathbf{U}_c'\mathbf{\Delta}_c^{-\frac{1}{2}}$.

Figure 3: The LCT algorithm.

3. EXPERIMENT RESULTS

This section compares the performance of the eigenspace update algorithms explained above to predict faces in a video sequence. The results presented here have been obtained using the "Miss America" video sequence formed by frames of $M = 3,072$ pixels and 25 frame/sec but similar results have been observed considering other video sequences. As a previous step, the faces have been aligned using the technique required in the extraction of MPEG-7 face recognition descriptors [3].

The first set of experiments is oriented to show the importance of the decay parameter $= {}_m = {}_v$ used in the LCT algorithm. Using a training sequence formed by the first 10 frames of the video sequence, we have computed the eigenspace for several values of . The frame #11 has been predicted from k retained eigenvectors ($k = 2,3,..,9$). Figure 4 shows the PSNR between the original frame and the predicted one. Note that a good PSNR has been obtained when the decay parameter is greather than 0.6. On the contrary, a small value of produces an important reduction on the quality. This bad behavior is more apparent when the number of retained eigenvectors is small.

For the case of $k = 6$ eigenvectors, Figure 5 shows the PSNR obtained when 20 frames are predicted using the initial eigenspace which has been computed from the first 10 frames. The eigenspace has not been updated for the other frames. Once again a good prediction has been obtained when > 0.6. Note as well that the performance decays for frames far to the training set.

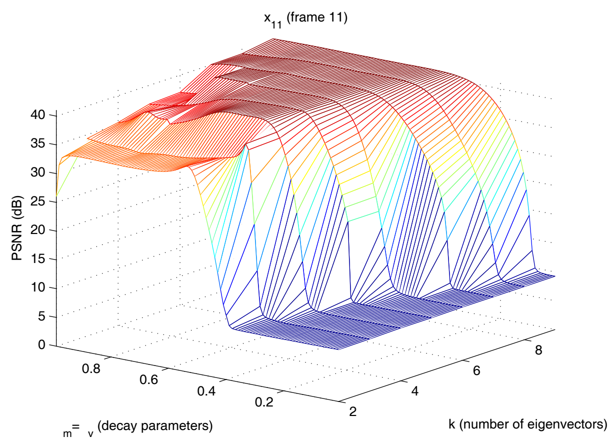


Figure 4: PSNR obtained by predicting the frame #11 using k eigenvectors obtained using the LCT algorithm with decay parameters $= \,_m = \,_v$.

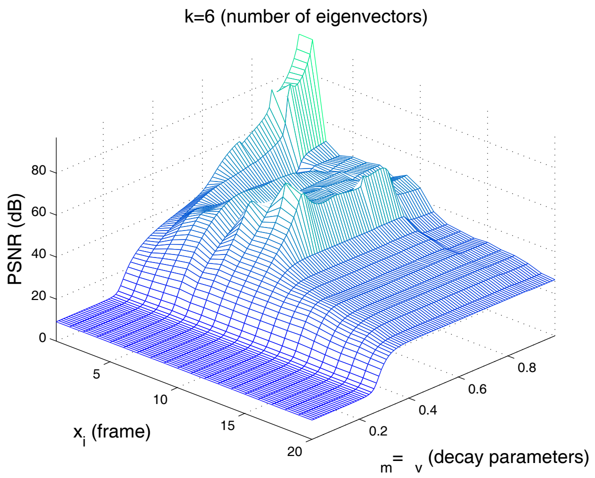


Figure 5: PSNR obtained by predicting 20 frames from $k = 6$ eigenvectors obtained using the LCT algorithm.

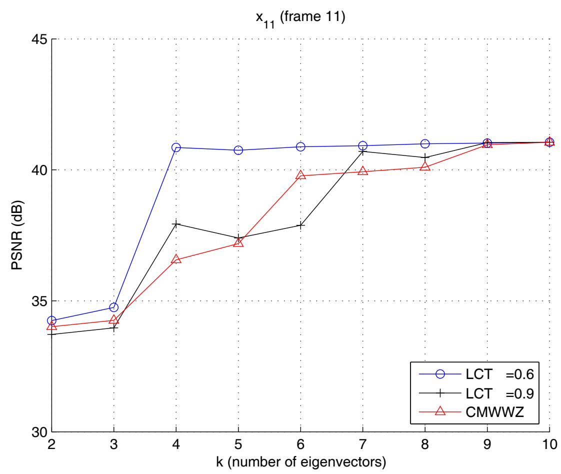


Figure 6: Comparison between the LCT and the CWWMZ algorithm when the frame #11 is predicted from k eigenvectors.

Algorithm	Number of updates	Coefficients
CMWWZ	47	$k = 6$
LCT, $= 0.9$	39	$k = 6$
LCT	30	$k = max = 1,\ldots,30$

Table 1: Number of updates performed by the CMWWZ and the LCT algorithms to code a video sequence of 140 frames. Note that each update implies to transmit an intra-frame.

In the next experiment, we have compared the prediction performance of both LCT and CMWWZ algorithms. Figure 6 shows the PSNR obtained when the frame #11 is predicted from k eigenvectors. Note that the LCT converges faster to the optimum value. This result is more apparent for $= 0.6$.

The last experiment is oriented to show the behavior of the video coding scheme presented above. The initial eigenspace has been updated in order to achieve a minimum PSNR of 30 dB ($= 65$). Figure 7 shows the PSNR obtained by predicting the 140 frames on the sequence using $k = 6$ eigenvectors: part (a) has been obtained using the CMWWZ algorithm and part (b) corresponds to the LCT algorithm with $= 0.9$. We have marked the points corresponding to intra-frame coded by using baseline JPEG with a quality of 83%. Although the achieved PSNR is similar, note that there exists an important difference in the number of transmitted intra-frames. From Table 1 we conclude that the CMWWZ algorithm needs to update the eigenspace 8 times more than the LCT algorithm.

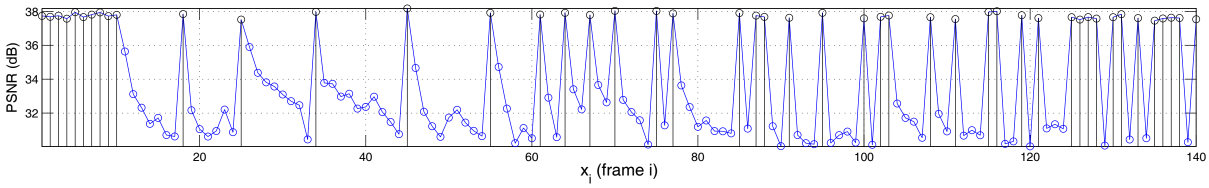
For comparison purpose, Figure 7 (c) plots the PSNR obtained using the LCT algorithm when k is equal to the maximum number of eigenvectors, i.e., the number k increases one unit at each update. In this case, any PCA algorithm provides the same performance. Comparing this results with the obtained for $k = 6$ (see also Table 1), we can say that the LCT algorithm needs to update the eigenspace 9 times more for $k = 6$ than for $k = max$ (i.e, from $k = 1$ to $k = 30$). This increase is compensated by a reduction in the computational cost of the frame predictor in both the coder and the decoder size. Recall that to evaluate (1) requires $k \times (M + (M - 1))$ flops.

Finally, Figure 8 shows the frame #81 of the video sequence and the reconstructed frames obtained using JPEG with a quality of 83%, the LCT algorithm with $= 0.9$ and $k = 6$, and the CMWWZ algorithm with $k = 6$. It is apparent that the best quality has been obtained using the LCT algorithm.

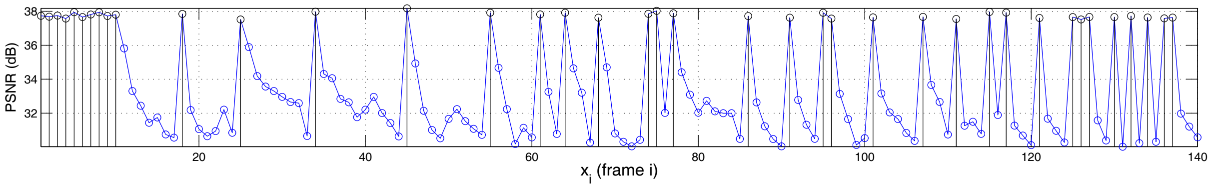
4. CONCLUSIONS

This paper presents a PCA-based video coding technique where the frames are represented as its projection in the space formed by the eigenvectors of the covariance matrix. Since the frames are dynamically acquired, we have used the algorithm proposed in [5] to update the eigenspace according to changes in the first and second-order statistics. Comparing the simulation results with the obtained using [1] (also used in [6]), we can say that the LCT algorithm needs less coefficients to achieve the same PSNR (about seven coefficients by frame). This result implies higher compression with less computational cost.

(a) CMWWZ algorithm with $k = 6$



(b) LCT algorithm with $k = 6$ and $= 0.9$.



(c) LCT algorithm with $k = max$

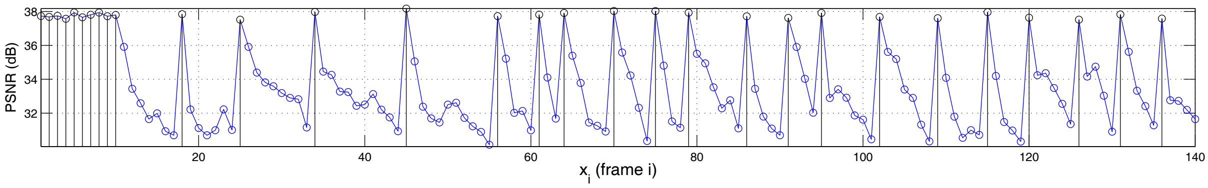


Figure 7: PSNR obtained using eigenspace update algorithms to predict 140 frames. Marked points correspond to intra-frames.

ACKNOWLEDGMENTS

We would like to thank Roger Piqué and Luis Torres for their help with video coding using PCA.

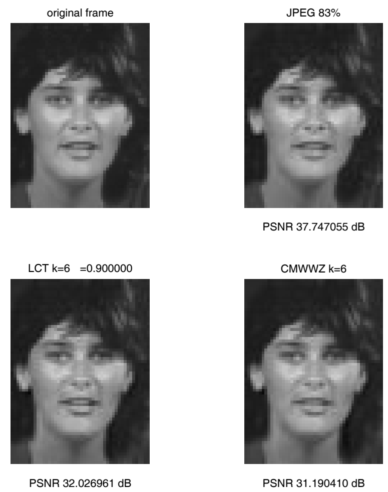


Figure 8: Visual quality obtained using JPEG and the eigenspace update algorithms with $k = 6$.

REFERENCES

[1] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkler, and H. Zhang, "An eigenspace update algorithm for image analysis", *Graphical Models and Image Processing*, vol. 59. no. 5, 1997.

[2] P. M. Hall, D. Marshall, R. Martin, "Incremental Eigenanalysis for Classification", Department of Computer Science, University of Wales Cardiff, UK, Research Report Serious No: 98001, May 1998.

[3] ISO/IEC 15938-3: 3002, Information technology – Multimedia content description interface – Part 3: Visual.

[4] A. Levy, M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its applications to Images", IEEE Transactions on Image processing, vol. 9, no. 8, pp. 1371-1374, 2000.

[5] X. Liu, T. Chen, S. M. Thornton, "Eigenspace updating for non-stationary process and its application to face recognition", accepted in *Patter Recognition*, Special issue on kernel and subspace methods for computed vision, 2002.

[6] R. Piqué, L. Torres, "Efficient face coding in video sequences combining adaptive principal component analysis and a hybrid codec approach", *Proc. International Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 629-32, 2003.

[7] M. A. Turk and A. P. Pentland "Face recognition using eigenface," *Proc. IEEE Conf. on computer vision and pattern recognition*, pp. 586–591, 1991.