This paper presents a novel video coding technique where most frames are represented as their projection onto a proper basis (eigenspace) computed using Principal Component Analysis (PCA). Since a video sequence contains regions with high time variations, a learning procedure is used to obtain an adequate basis. We also introduce the idea of bidirectional predicted frames to denote those frames that can be estimated from the nearest past and future PCA coefficients. Experimental results show the high quality/compression achieved using the new scheme with different eigenspace updating algorithms.

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 (eigenspace) [3, 9]. Recently, Piqué and Torres [7] have argued that PCA is a promising technique for coding faces in video sequences. The idea is to predict the actual frame by calculating their projection onto the eigenspace calculated from previous frames. The coefficients are therefore coded and transmitted. Full frames are only coded when a poor representation is obtained and, in this case, the eigenspace is updated using the algorithm proposed in [1].

In this paper, we propose a novel scheme for coding frames in a video sequence using PCA. Similarly to other video standards [8], we use three types of frames: intra frames (I-frames), predicted frames (P-frames) and bidirectional predicted frames (B-frames). I-frames are coded without reference to other frames in the video sequence, they are access points but achieve only low compression. From the I-frames, the encoder (and the decoder) computes the eigenspace which will be used to project the other frames. P-frames are represented as the PCA coefficients obtained by projecting the frames onto this eigenspace. Finally, B-frames are coded using zero coefficients and they are reconstructed by averaging the PCA coefficients corresponding to the the nearest past and future P-frames.

A key aspect in the performance of the new scheme is the ability to learn from the variations in the video sequence. Towards this aim, we will consider the eigenspace update algorithms described in [1] and [6]. Both algorithms have been initially proposed for face recognition where the training set is formed by different poses and faces. In Section 2 we will present a short revision of both methods and a comparison for predicting frames in a video sequence.

The remaining of the paper is organized as follows. In Section 3 we will propose the new video coding scheme. Section 4 provides the results of several simulations which show the high compression achieved with the proposed scheme and compares the performance obtained using two eigenspace updating algorithms. Finally, Section 5 presents the main conclusions and some ideas about further work.

2. FRAME PREDICTION USING PCA

We will interpret a video sequence as a sequence of 2-D frames of $M$ pixels which we will treat as 1-D vectors of dimension $M \times 1$ denoted by $x_i$, $i = 1, 2, ...$. In order to represent each frame with its PCA coefficients, we project $x_i$ onto the eigenspace computed from previous frames by using

$$\omega_{pi} = u_p^T x_i, \quad p = 1, ..., M$$

where $u_p$ is the $p$-th eigenvector of the data eigenspace. Evaluation of expression (1) requires $M$ multiplications and $M - 1$ additions for each coefficient, i.e., $M + (M - 1)$ flops. However, it is well known that PCA packs the maximum average energy in few eigenvalues. For this reason, we will retain only the eigenvectors corresponding to the $k$ largest eigenvalues. From these $k$ PCA coefficients, the frame is reconstructed by using

$$\hat{x}_i = \sum_{p=1}^{k} \omega_{pi} u_p$$
Since video sequences are dynamically acquired, the system may learn each time a new frame is captured. Towards this aim, many existing algorithms can be used to update the eigenspace according to changes in the sequence [1, 2, 5, 6]. Among them, we have studied the performance of the algorithm proposed by Chandrasekaran, Manjunath, Wang, Winkler, and Zhang in [1] (henceforth named CMWWZ) and the proposed by Liu, Chen, and Thornton [6] (henceforth named LCT).

2.1. The CMWWZ algorithm

The CMWWZ algorithm computes the set of eigenvectors $U = [u_1, ..., u_M]$ by performing the Singular Value Decomposition (SVD) on the image matrix, instead of the covariance matrix. The set of images $X_{i-1} = [x_1, ..., x_{i-1}]$ is represented as $X_{i-1} = U_{i-1} \Delta_{i-1} V_{i-1}^T$ where $\Delta_{i-1}$ contains the eigenvalues, and $U_{i-1}$ and $V_{i-1}$ are the eigenvector matrices. For a new captured image, $x_i$, the SVD is recalculated as

$$[U_{i-1} \Delta_{i-1} V_{i-1}^T x_i] = U_i \Delta_i V_i^T$$

(3)

The size of the matrices can be reduced by taking only the largest eigenvalues and the corresponding eigenvectors. Figure 1 contains more details about the algorithm. Note also that we do not include the steps needed to compute the matrix $V_i$ because it is not used in the video coder scheme presented in Section 3.

2.2. The LCT algorithm

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

$$m_i = \alpha_m m_{i-1} + (1 - \alpha_m)x_i$$

$$\hat{C}_i = \alpha_v \hat{C}_{i-1} + (1 - \alpha_v)(x_i - \hat{m}_i)(x_i - \hat{m}_i)^T$$

(4)

where $\alpha_m$ and $\alpha_v$ are the decay parameters. Note that the covariance matrix $\hat{C}_{i-1}$ can be approximated by using the $k$ retained eigenvectors (and eigenvalues) at time $i-1$, i.e.,

$$\hat{C}_i \approx \alpha_v U_{i-1} \Delta_{i-1} U_{i-1}^T + (1 - \alpha_v)(x_i - \hat{m}_i)(x_i - \hat{m}_i)^T = B_i B_i^T$$

(5)

where $B = [\sqrt{\alpha_v}U_{i-1} \Delta_{i-1}^{1/2} \sqrt{1 - \alpha_v}(x_i - \hat{m}_i)]$. The matrices $U_{i-1}$ and $\Delta_{i-1}$ contain, respectively, the $k$ eigenvectors and eigenvalues retained at time $i-1$. Once $\hat{C}_i$ has been estimated, the SVD is used to find the $k$ largest eigenvectors which represent the new eigenspace.

Figure 2 summarizes the algorithm (the initialization step has not been included). Note that the last step is needed because we compute the eigenvectors of matrix $A = B_i^T B_i$ instead of (5).

Given the frame $x_i$, the eigenvector matrix $U_c$, the eigenvalue matrix $\Delta_c$, and the estimated mean $m$ compute

- $m \leftarrow \alpha_m m + (1 - \alpha_m)x_i$
- $B \leftarrow [\sqrt{\alpha_v}U_{i-1} \Delta_{i-1}^{1/2} \sqrt{1 - \alpha_v}(x_i - m)]$
- $A \leftarrow B_i^T B_i$
- Compute the eigenvectors, $U'$, and eigenvalues, $\Delta$, of $A$
- Let $U'_c$ equal the first $k$ columns of $U'$
- Let $\Delta_c$ equal the leading $k \times k$ principal submatrix of $\Delta$
- $U_c \leftarrow B U'_c \Delta_c^{-1}$.

Fig. 2. The LCT algorithm.

2.3. Performance comparison

This subsection is dedicated to compare 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 [4]. This preprocessing is needed to guarantee a high correlation in the input data.

First, we will show the importance of the decay parameter $\alpha = \alpha_m = \alpha_v$ used in the LCT algorithm. For that, we have computed the eigenspace for several values of $\alpha$ using a training sequence formed by the first 10 frames. The frames #1 to #20 have been predicted from $k = 6$ eigenvectors. Figure 3 shows the Peak Signal Noise Ratio (PSNR) between the original and the reconstructed frames

$$PSNR(x_i, \hat{x}_i) = 10 \log_{10} \frac{255^2}{E[(x_i - \hat{x}_i)^2]}$$

Note that a good prediction has been obtained when $\alpha > 0.6$. It is also apparent that the performance decays for frames far to the training set.

The second experiment compares the performance of both LCT and CMWWZ algorithms to predict a frame from $k$ eigenvectors (or PCA coefficients). Figure 4 shows the PSNR obtained using LCT with different values of $\alpha$ and using CMWWZ. It can be seen that LCT achieves the optimum PSNR with less eigenvectors. This result is more apparent for $\alpha = 0.6$ where the optimum value is achieved with only $k = 4$ coefficients.

3. VIDEO CODING SYSTEM

The basic idea of our video coding system is to represent a frame using its projection on the eigenspace computed from previous frames (PCA coefficients). However, this representation can be poor for frames corresponding to strong variations in the video. In this case, the full image may be transmitted and the eigenspace updated. On the contrary, when there exists a small difference between a frame and the nearest past and future frames, the PCA coefficients take very similar values and it is reasonable to think that a good performance will be obtained by averaging the past and the future PCA coefficients.

Taking into account these considerations, we propose to divide the video sequence in groups of frames (GOFs) formed by three types of frames: I-frames coded using JPEG, P-frames coded with the PCA coefficients and B-frames coded with zero coefficients. In the decoder, B-frames are reconstructed by averaging PCA coefficients.

Figures 5 and 6 summarize the encoder and the decoder steps, respectively. The $j$-th GOF begins with an I-frame, $x_{j,1}$, which is coded without reference to other frames and produces an eigenspace update. For the even frames in the GOF, $x_{j,i}$, $i = 2, 4, \ldots$, the decoder computes the PCA coefficients using

$$\omega_{pj} = u_p^T x_i, \quad p = 1, 2, \ldots, k$$

where $k$ is the number of retained eigenvectors. In the next step, the image is reconstructed from the PCA coefficients and the PSNR is calculated, $\epsilon = PSNR(x_{j,i}, \hat{x}_{j,i})$ where
Given the video sequence ..., \(x_{i-1}, x_i, x_{i+1}, \ldots\) do:

1. Compute the PCA coefficients (7) for the frame \(x_i\).
2. Compute \(\hat{x}_i\) using (8).
3. Evaluate the \(\epsilon = PSNR(x_i, \hat{x}_i)\).
4. If \(\epsilon < \sigma\)
   (a) If \(x_{i-1}\) has not been code yet, set \(i = i - 1\) and go to step 1.
   (b) Else
      i. Set the number of frames of the \(j\)-th GOF to one, \(N_j = 1\), and \(x_{j,N_j} \leftarrow x_i\).
      ii. Code \(x_{j,N_j}\) as an I-frame and update the eigenspace.
      iii. Set \(i = i + 1\) and go to step 1.
5. Else
   (a) Set \(N_j = N_j + 1\) and \(x_{j,N_j} \leftarrow x_i\).
   (b) Code the PCA coefficients \(\omega_{p,N_j}\).
   (c) Set \(i = i + 2\) and go to step 1.

\[
\hat{x}_{j,i} = \sum_{p=1}^{k} \omega_{pi} u_p
\] (8)

If the frame has been predicted with enough quality (\(\epsilon \geq \sigma\)), only the PCA coefficients will be coded. On the contrary, when \(\epsilon < \sigma\), the system decides that a P-frame is a bad representation and set the end of the GOF. In this case, it is needed to code the previous frame as a P-frame.

In a GOF, a B-frame is used to represent a frame between two P-frames. No computations are performed in the encoder for this kind of frame. In the decoder, a B-frame is reconstructed using (8) where the PCA coefficients are obtained by averaging the PCA coefficients corresponding to the nearest past and future P-frames. Although more sophisticated methods could be used, we have seen that a good estimation is obtained by averaging the PCA coefficients, i.e., \(\omega_{pi} = (\omega_{p(i-1)} + \omega_{p(i+1)})/2\).

**4. EXPERIMENT RESULTS**

This Section presents some results obtained using the proposed video coding scheme. We have considered the video sequence “Miss America” described in Subsection 2.3 which is formed by 140 frames. The eigenspace dimension has been truncated to \(k = 6\) eigenvectors and it has been updated in order to achieve a minimum PSNR of 30 dB (\(\sigma = 30\)). I-frames have been coded using JPEG with a quality of 83% and P-frames have been coded using a fixed length code of 16 bits per coefficients (i.e., 96 bits per frame). We have compared the performance of our coding scheme with that obtained using only I-frames and P-frames, and with the optimal one where all frames are coded with JPEG (i.e., only I-frames are used).

In the first experiment we have used the CMWWZ algorithm to update the eigenspace when \(\sigma < 30\). Figure 7 shows the PSNR between the original and the reconstructed frames. We also show the frame type corresponding to each point. Note that the performance is not degraded when B-frames are used instead of P-frames.

Figure 8 shows the results obtained using the LCT algorithm. Comparing with Figure 7, we can conclude that the achieved PSNR is similar. However, the video sequence (see also Table 1) indicates that the CMWWZ algorithm uses 8 I-frames more than the LCT algorithm. This is an important drawback because it affects not only to the compression but also to the computational cost. Recall that the eigenspace is update each time an I-frame is transmitted.

Table 1 presents the mean PSNR and the bits per frame for several strategies. Note the considerable reduction in the bit rate of PCA with respect to JPEG. It is also apparent that the inclusion of B-frames allow to reduce the compression rate without affecting the PSNR.

Finally, Figure 9 shows the frame #81 and the recovered frame when it is compressed as an I-type, a P-type
Fig. 7. PSNR obtained using the proposed approach when the eigenspace is updated with the CMWWZ algorithm. For comparison purposes it is also plotted the results obtained using only I-frames, and using both I- and P-frames.

Fig. 8. PSNR obtained using the proposed approach when the eigenspace is updated with the LCT algorithm. For comparison purposes it is also plotted the results obtained using only I-frames, and both I- and P-frames.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>I-frames</th>
<th>Number of P-frames</th>
<th>B-frames</th>
<th>Avg. PSNR (dB)</th>
<th>Avg. bits per frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCT (I, P &amp; B)</td>
<td>39</td>
<td>68</td>
<td>33</td>
<td>33.4645</td>
<td>2,382.51</td>
</tr>
<tr>
<td>LCT (I &amp; P)</td>
<td>39</td>
<td>101</td>
<td>–</td>
<td>33.4811</td>
<td>2,405.14</td>
</tr>
<tr>
<td>CMWWZ (I, P &amp; B)</td>
<td>47</td>
<td>66</td>
<td>27</td>
<td>33.7116</td>
<td>2,864.63</td>
</tr>
<tr>
<td>CMWWZ (I &amp; P)</td>
<td>47</td>
<td>93</td>
<td>–</td>
<td>33.7260</td>
<td>2,887.26</td>
</tr>
<tr>
<td>Only I frames</td>
<td>140</td>
<td>–</td>
<td>–</td>
<td>37.7388</td>
<td>8,392.97</td>
</tr>
</tbody>
</table>

Table 1. Results obtained using the proposed scheme with the LCT and the CMWWZ algorithms. For comparison purposes, we also present the values when B-frames are not used and when only I-frames are coded.
and a B-type. The eigenspace has been computed using the LCT algorithm with $\alpha = 0.9$. It is apparent that the visual quality is very similar for the three frames types.

5. CONCLUSIONS AND FURTHER WORK

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 (PCA coefficients). A good prediction allows to represent the frame using only these PCA coefficients (about six coefficients per frame). We introduce also the idea of B-frame applied to PCA to represent those frames coded using zero coefficients because they can be estimated using the nearest past and future PCA coefficients. Finally, I-frames (coded with JPEG) are used when a poor representation is obtained using PCA.

An important module of our scheme is the algorithm used to update the eigenspace when an I-frame is transmitted. We have compared two eigenspace updating algorithms: the LCT algorithm proposed in [6] and the CMWWZ algorithm proposed in [1] (also used in [7]). Experimental results show that the LCT algorithm achieves the same PSNR with a higher compression.

Further work deals with improving our scheme by encoding the PCA coefficients using, for instance, the AVC (Advanced Video Coding) standard. A second work under development is to use Independent Component Analysis (ICA) instead of PCA. We hope that ICA will not require to align the frames because high correlation is not needed in the input data.

6. REFERENCES


