Video Skimming and Summarization Based on Principal Component Analysis

Dan Lelescu[1] and Dan Schonfeld[2]

[1] Compression Science, Inc.,
901 Campisi Way, Campbell, Ca., U.S.A.
dan@compressionscience.com
[2] University of Illinois at Chicago,
851 S. Morgan St., Chicago, IL., U.S.A.
ds@eecs.uic.edu

Abstract. An increasing number of applications such as content-based multimedia retrieval in a distributed system and low-bitrate video communications, require the efficient processing and transmission of video information. In content-based video retrieval, video segmentation produces video shots characterized by a certain degree of visual cohesiveness. The number of relevant video shots returned by the system can be very large, thereby requiring significant transmission bandwidth. In this paper, we present a new algorithm for the representation of visual information contained in video segments. The approach is based on Principal Component Analysis and takes advantage of the characteristics of the data in video shots, and the optimal energy compaction properties of the transform. The algorithm can use additional information about video sequences provided by a video analysis and retrieval system, such as a visual change estimator, and a video object tracking module.

1 Introduction

In this paper, we present a video representation approach that utilizes a structuring of the video data and determines the optimal energy compaction of the resulting video entities in order to enable their efficient representation and compression. Although the emphasis of the paper is placed on the video retrieval context, the extension of the approach to the video compression context is also discussed.

Content-based multimedia processing and retrieval systems are created to offer the means to automate the selective access to information, as expressed in a query. Ideally, the retrieval system should find and return to the user only the information that is relevant for a particular query. The information obtained as a result of processing video shots, such as key frames, video objects, trajectories, can be used for immediate response to a query, or it can be indexed for later retrieval. The detail level at which the user initially receives visual information from the processing system as a result of a query is hierarchical in that the user may be given one or multiple key frames per relevant video shot, a text

E.S. Al-Shaer and G. Pacifici (Eds.): MMNS 2001, LNCS 2216, pp. 128–141, 2001.

description of the video shot contents, or the full-resolution shot. Text or key frames are limited in their capacity to provide dynamic information about a video shot. For a large video archive the number of video shots returned by the system may be significant. A full resolution video shot might have to be separately decoded and then re-encoded by itself for transmission. An alternative for providing the user with sufficient video information to allow for a decision about the full resolution video shot is to efficiently transmit a lower resolution representation of it.

In this paper, we discuss a method for the efficient representation of video shots. This approach can be applied hierarchically both in terms of resolution, and scene contents (i.e., full frame, or objects). For this purpose, we utilize the optimality properties of the Principal Component Analysis transformation in terms of energy compaction. Also, information provided by other components of the multimedia processing system, such as the video segmentation and video tracking modules, is used to facilitate the operation of the PCA-based algorithms at the corresponding hierarchical levels mentioned above. We propose an approach that utilizes the notion of visual activity in video sequences to facilitate important savings in the number of transform coefficients required to represent each video frame, while maintaining a reasonable level of distortion of the reconstructed video sequence. The information about the activity in a video segment is provided by a statistical estimator of visual changes in the video scene. The granularity of the video segmentation process can be adjusted to ensure a sufficient degree of visual consistency and locality in video shots. Given this new structuring of video information, a local PCA-based representation at the level of video shots becomes very efficient. This representation can be applied at the full frame level, or at the video object level. For the latter case, moving video objects can be extracted and tracked through the video sequence, for the purpose of enabling their PCA representation and separate encoding.

The PCA-based video representation approach presented can also represent a basis for developing video compression algorithms. The temporal segmentation of the original image sequence is achieved through the use of video segmentation, to obtain the video segments which are processed by the algorithm. The same video segmentation approach mentioned above can be applied for full or lower-resolution image sequences. The determined video segments can be encoded at various resolution levels, taking into account the increasing computation that comes with increasing image resolution. The hierarchical operation of the algorithm in terms of scene contents is also applicable through the separate encoding of video objects. This can be made possible by object recognition and tracking in the image sequence. Also, in this paper, we investigate the PCA-based transformation of the video data for the purpose of compressed representation. The issues of optimal quantization and entropy coding are beyond the scope of this presentation.

The paper is organized as follows. Section 2 reviews related work. In Section 3, algorithms for PCA are discussed. Section 4 presents the PCA-based repre-

sentation of video sequences. Section 5 contains simulation results. The paper is concluded in Section 6.

2 Related Work

Content-based multimedia processing, retrieval and transmission naturally shares topics with the low bit rate video compression area. Content-based video retrieval and indexing [1], [2], [3], [4], has the potential to automate to a large degree the multimedia retrieval process. The objective of such systems is to provide selective access to multimedia data, by finding query-relevant video segments and presenting them to the user. Once video segments have been found as relevant, various levels of information regarding the shots must be transmitted to the user. This information can include a text description of the shot, one keyframe per shot, visual storyboards consisting of multiple keyframes, time-compressed video shots, and low or full resolution versions of the video shot. Dynamic video information is the most preferable to be transmitted to the user.

Next, video coding techniques are briefly discussed. Waveform coding methods (i.e., transform and subband coding) encode data using the pixel structure of an image, whether at full frame level, or block level. Although these methods are well-established at high bit rate encoding, they have known limitations at low bit rates and introduce artifacts to which the human visual system is very sensitive. Block-based transform techniques have been included in video compression standards such as the MPEG family and the H.26x. Subband coding [5],[6], utilizes a decomposition of the image as produced by an analysis filter bank which is followed by down-sampling. The resulting subbands can represent the input to other stages of analysis/downsampling, producing a hierarchical, multi-resolution description of the image data.

Second generation (object level) coding techniques utilize the scene contents to improve the low bit rate coding performance and eliminate artifacts of the waveform coding techniques. However, contour and texture may require complex encoding operations and a significant bit rate allocation. These techniques attempt to provide better performance by considering the visual scene contents and its structure, as opposed to an artificial partition of an image into blocks. Object-based coding [7], [8], depends on the ability to recognize objects in the scene and represent them by shape, texture, and motion. The extraction of objects from the video scene is difficult and currently can be achieved satisfactorily if there are not many moving objects in the scene, the object motion is dominant and moderate, and camera motion is limited. The possibility that object/region based encoding may fail for particular types of video sequences or portions of a video sequence has prompted the creation of hybrid coders that use block based coding when the object based techniques become unfeasible.

The use of PCA for optimal dimension reduction and transform coding for still images has received renewed attention [9], [10]. In [9], the resource allocation using local linear models for non-linear PCA is discussed. In the local PCA model, the data is partitioned into regions and PCA is performed in each

region. The local PCA model is applied to image dimension reduction and transform coding. The allocation of PCA representation and coding to different image regions permits the adjustment of the dimension locally, while the average dimension is constrained to a particular value. In [10], the problem of finding an appropriate data partition for the application of the local PCA is investigated. In this paper, we use the PCA transform as the basis for encoding video data. We introduce a video sequence representation approach that can facilitate important savings in the number of transform coefficients required to represent each video frame. Using a visual activity-based structuring of video sequences and video object tracking in scenes, a PCA-based representation becomes suitable for use in encoding video segments, with additional computational methods to make such algorithms efficient.

3 Algorithms for Efficient Principal Component Analysis

Let $\mathbf{P}$ be an $N_c \times L$ *data matrix* corresponding to a set of L data vectors with dimension $N_c \times 1$. In the following, we use deterministic estimates of statistical variables, by taking the matrix $\mathbf{C} = \mathbf{P}\mathbf{P}^T$ to be an estimate of the correlation matrix of the data. Let X_k denote a data vector (column) of matrix $\mathbf{P}$. The Principal Component Analysis (PCA) performs a partial KL transform by finding the largest $M < L$ eigenvalues and corresponding eigenvectors of $\mathbf{C}$. The new representation Y_k of an original data vector X_k is $Y_k = \mathbf{\Phi}_M^T X_k$, where $\mathbf{\Phi}_M$ is the eigenmatrix formed by selecting only the M eigenvectors corresponding to the largest M eigenvalues.

Assuming that $N_c >> L$ is very large for the case of interest, the size of matrix $\mathbf{C}$ is also large, which would result in computationally intensive operations using the direct PCA computation. As described in [11], an efficient approach for negotiating this problem is to consider the implicit correlation matrix $\tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{P}$. The matrix $\tilde{\mathbf{C}}$ is of size $L \times L$, which is much smaller than the size of $\mathbf{C}$. The $M \leq L-1$ *largest* eigenvalues λ_i, and corresponding eigenvectors e_i of $\mathbf{C}$ can be *exactly* found from the $M \leq L-1$ largest eigenvalues and eigenvectors of $\tilde{\mathbf{C}}$ as follows [11]: $\lambda_i = \tilde{\lambda}_i$, $e_i = \tilde{\lambda}_i^{-\frac{1}{2}}\mathbf{P}\tilde{e}_i$, where $\tilde{\lambda}_i$, $\tilde{e}_i$ are the corresponding eigenvalues and eigenvectors of $\tilde{\mathbf{C}}$. The eigenvectors $\tilde{e}_i$ of $\tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{P}$ are given by the right singular vectors of $\mathbf{P}$, determined using the SVD.

An efficient iterative approach for computing *approximations* of the M largest eigenvalues and corresponding eigenvectors of $\mathbf{C}$ was also proposed in [11]. It is assumed that the data vectors are processed sequentially. The algorithm is initialized by direct computation of the M most significant eigenvectors of an initial set of $(M+1)$ data vectors. In the following iterative procedure only the M eigenvectors corresponding to the largest eigenvalues are retained at every stage of the iteration. For every new input vector, the M eigenvectors computed in the previous step are refined. Let us denote by $\{x^{(i)}\}_{i=1\ldots L}$ the L data vectors. Assume that only the M most significant eigenvectors $\{\psi_L^{(i)}\}_{i=1\ldots M}$, obtained after all L data vectors have been processed, will be retained. Let $\mathbf{A}_k$ be an $(M+1) \times (M+1)$ correlation matrix formed at iteration k. Its corresponding

eigenvectors and eigenvalues are $\{a_k^{(i)}\}_{i=1\ldots M}$ and $\{\lambda_k^{(i)}\}_{i=1\ldots M}$. The iterative PCA algorithm proceeds as follows:
Step 0:
Set $k = M + 1$.
Read the (M+1) data vectors $\{x^{(i)}\}_{i=1\ldots M+1}$.
Determine the matrix $\mathbf{A}_k$
Calculate eigenvectors $\{a_k^{(i)}\}_{i=1\ldots M}$ and $\{\lambda_k^{(i)}\}_{i=1\ldots M}$ by direct calculation.
Compute the initial set of M eigenvectors $\{\psi_k^{(i)}\}_{i=1\ldots M}$ as follows:

$$\psi_{M+1}^{(i)} = (a_{M+1}^{(i)})_1 x^{(1)} + (a_{M+1}^{(i)})_2 x^{(2)} + \ldots + (a_{M+1}^{(i)})_{M+1} x^{(M+1)}. \tag{1}$$

where $(a_k^{(i)})_I$ is the I^{th} component of the i^{th} eigenvector $a_k^{(i)}$, at iteration k.
Step 1:
$k = k + 1$.
As the new vector $x^{(k)}$ is processed, re-compute the $(M + 1) \times (M + 1)$ matrix $\mathbf{A}_k$ from $x^{(k)}$ and the previous principal vectors $\{\psi_k^{(i)}\}_{i=1\ldots M}$ as follows ($(\mathbf{A}_k)_{i,j}$ represents the $(i, j)^{th}$ entry in matrix $\mathbf{A}_k$):

$$\begin{aligned}
(\mathbf{A}_k)_{i,j} &= \frac{k-1}{k}(\lambda_{k-1}^{(i)}\lambda_{k-1}^{(j)})^{1/2}\delta_{ij}; i, j = 1 \ldots M, \\
(\mathbf{A}_k)_{i,M+1} &= (\mathbf{A}_k)_{M+1,i} = \frac{1}{k}\psi_{k-1}^{(i)T} x^{(k)}; i = 1 \ldots M, \\
(\mathbf{A}_k)_{M+1,M+1} &= \frac{1}{k} x^{(k)T} x^{(k)}.
\end{aligned} \tag{2}$$

The updated principal eigenvectors $\{\psi_k^{(i)}\}_{i=1,M}$ and their corresponding eigenvalues $\{\lambda_k^{(i)}\}_{i=1\ldots M}$ are then obtained by finding the eigenvalues and eigenvectors of matrix $\mathbf{A}_k$ and using the following formula:

$$\psi_k^{(i)} = (a_k^{(i)})_1 \psi_{k-1}^{(1)} + (a_k^{(i)})_2 \psi_{k-1}^{(2)} + \ldots + (a_k^{(i)})_M \psi_{k-1}^{(M)} + (a_k^{(i)})_{M+1} x^{(k)}. \tag{3}$$

Step 2:
Repeat Step 1 until $k = L$ (all vectors have been processed).
At $k = L$, normalize the M retained eigenvectors $\{\psi_L^{(i)}\}_{i=1\ldots M}$ by $1/(\sqrt{L\lambda_L^{(i)}})$ to obtain the normalized principal vectors of the data set $\{x^{(k)}\}_{k=1\ldots L}$.

4 PCA-Based Representation of Video Sequences

The video representation and compression algorithms presented in this section are intended for use in two contexts that require the efficient coding of video sequences. In the case of a distributed multimedia processing and retrieval system, the query-relevant video shots must be transmitted to the user's location. Within computational considerations, the proposed compression approach can represent the basis for encoding video data at very low bit rates. In both cases, the use of video segmentation, visual change estimation, and object tracking,

enables the operation of the PCA-based video representation algorithm. A video segmentation algorithm can be applied to compressed or raw video data. The object extraction and tracking information, however, is obtained differently depending on the context. For example, for motion-compensated video, objects can be extracted and tracked as presented in [3].

4.1 Video Shot Representation in a Multimedia Retrieval System

In the case of a distributed multimedia processing and retrieval system, the query-relevant video shots must be efficiently transmitted to the user. As mentioned before, the information presented to the user can be hierarchical in nature (a single key frame, multiple key frames, story boards (tree structures) of key frames, lower-resolution versions of the video shot, time-compressed video shots, or the full-resolution video shots).

We shall illustrate the operation and characteristics of the proposed video representation algorithm for the case of a video shot extracted from a compressed MPEG-2 video sequence. The preliminary operations of video segmentation and video object tracking are assumed to have already taken place. Thus, we have access to the video shot of interest in compressed form. Additionally, as produced by the other modules of the retrieval system operating on compressed bitstream ([3], [4]), there exists information regarding the visual changes in the video shot, and objects of interest in the shot. To introduce the algorithm, its functionality is presented for the case of low resolution, frame-level operation. For efficiency purposes, a low resolution version of the original video shot can be directly extracted from the bistream by considering only the DC coefficients of the blocks in each picture of the shot. One of the important characteristics of video shots is that they are relatively visually-coherent, that is, the visual information in the scene is expected to vary within certain limits.

For the case considered, the video segment to be encoded consists of a sequence of DC images. Each of these images are lexicographically ordered, resulting in the corresponding sequence of DC vectors. Let us denote these vectors by $\{X_k\}$, having dimensionality $N_c \times 1$. Let us also assume that the number of vectors in the video segment is L. We are interested in approximating the original vector space by a much smaller number M of eigenvectors. In order to use a PCA-based representation of the video segment, a number of options are available. The PCA-based representation algorithms can use the *training sample* approach, or the *iterative* approach, as described in Sect. 3.

For the training sample PCA approach (TS-PCA), a training subset τ of sample vectors taken from the entire vector set can be used. The M largest eigenvalues and the corresponding eigenvectors of τ can be found using the procedure outlined in Sect. 3. The retained eigenvectors can be utilized to represent the original DC vectors X_k in the video segment by the corresponding transformed vectors Y_k of dimensionality $M \times 1$:

$$Y_k = \mathbf{\Phi}_M{}^T X_k. \tag{4}$$

where $\mathbf{\Phi}_M$ is the corresponding eigenmatrix.

Additionally, two options are available for the selection of the training set τ. The first one is to select this set randomly from the vectors forming the video segment. The second option is to use information provided by modules of the retrieval system. Specifically, one by-product of the video segmentation approach presented in [4] is related to the fact that the activity or visual changes in a video shot are indicated by the test statistic g_k that is used for the detection of scene changes. The test statistic can be used as an estimator of activity in the video shot in order to improve the performance of the PCA representation, by means of selecting a training sample from the shot. The samples can be assigned in a non-uniform manner, by taking more samples in portions of high activity in the shot. The simulations conducted show that the performance of the TS-PCA algorithm is dependent on the degree of visual change that takes place in the video shot. Also, as expected, taking a training sample that spans the entire video segment results in improved performance compared to selecting the training set only from the first images in the shot. The alternative to using a training sample for the PCA video representation, is to use the iterative PCA algorithm (I-PCA) presented in Sect. 3. Because this approach uses all the data vectors in the set in determining their PCA representation, the corresponding algorithm is expected to provide an improvement in the quality of the representation. This is confirmed by the simulations presented in Sect. 5.

For the lossy reconstruction of the images in the video shot, the information needed consists of the eigenspace description as provided by the retained eigenvectors in matrix Φ_M, and the coordinates of each image in this space, represented by its corresponding vector Y_k. Formally, using the orthogonality of the transform, a reconstructed vector (image) is obtained as follows:

$$\hat{X}_k = \Phi_M Y_k. \tag{5}$$

Up until now, we have considered the case of applying the PCA-based representation algorithm for low-resolution video shots and at the frame level. Its functionality can be extended to a hierarchical level based on video scene contents. In general, tracking can provide us with information about the position and size a specific object through the frames of the shot. This information can be used to extract query-relevant object 'streams' in a video shot, by recording the images of the object as it appears in each frame of the shot. Furthermore, this enables the diversification of video representation and encoding at both full image and object levels. Simulations conducted show that the separate PCA-representation of moving objects in the scene results in superior quality of the reconstructed object images, compared to the exclusive frame-level PCA representation, for small to moderate affine changes of the object inside the tracking rectangle.

4.2 Video Compression

The functionality of the PCA-based video representation and compression can constitute a basis for developing algorithms for encoding raw video data. The

original video data can be assigned new structure through the use of video segmentation, change estimators, and video object tracking.

The video segmentation approach can be applied to images at a given resolution. For example, a subband transform can be used for generating a multiresolution representation of the original images. As we know, low-resolution images are sufficient for the detection of scene changes in the video. The resulting video segments are represented using the PCA algorithm with a similar technique to that described in Sect. 3. This is shown in the block diagram in Fig. 1. From a practical point of view, two challenges arise. One is related to the fact that as the resolution of the images increases, the visible distortion introduced by the PCA representation can increase for video sequences with large motion in the scene. The locality of the data has to be ensured by selecting the granularity (threshold) of the video segmentation process. Also, for video frames at full resolution the dimensionality increases, and therefore the computation cost associated with the PCA algorithm increases as well. An alternative to the PCA

Fig. 1. Hierarchical video compression for different image resolutions.

representation of the actual image sequence is to encode the sequence of corresponding error images. If the image with index k in the sequence is denoted by X_k, then the error image E_k is obtained as $E_k = X_k - X_{k-1}$. The vectors corresponding to the sequence E_k are represented by the PCA algorithm discussed in this section. The most significant M eigenvectors of the residual sequence and the PCA representation of each residual image, along with the first image in the video sequence must be encoded and transmitted. Through the inverse transformation, the residual images are reconstructed, and using the first image for the initial step, all other frames of the video segment are obtained, i.e., $\hat{X}_k = \hat{X}_{k-1} + \hat{E}_k$. One additional area that is currently investigated consists of the use of a PCA-based approach in conjunction with motion compensation (MC). For example, the motion fields corresponding to frames from a video shot can be modeled and transmitted using a PCA-based technique.

The PCA-based video representation can also be applied hierarchically based on video scene contents, similarly to the case of compressed content-based processing. Tracking allows for the registration of the objects inside the tracking rectangle and contributes to the locality of video data at the object level. The objects are extracted and tracked inside a determined video segment, obtained through the prior video segmentation of the image sequence. The PCA transform can be applied to the sequence of object images. For video object tracking in video segments two options are available. Firstly, objects can be extracted and tracked through the frames of the raw video sequence. The images of the

objects as delimited by the tracking bounding box form the object streams that are processed by the PCA representation algorithm, as shown in Fig. 2.

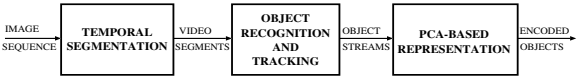

Fig. 2. Object level video compression.

Secondly, block-level motion compensation can be applied to the original image sequence so that the object recognition and tracking can use the generated motion information. This information can be used to extract the object streams in a video segment, composed of the images of the tracked object. This process is illustrated in Fig. 3.

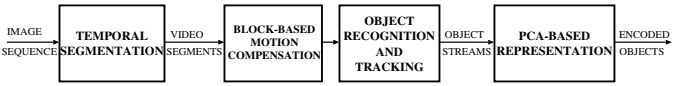

Fig. 3. Object level video compression on motion compensated video sequence.

One additional observation is that whenever object tracking is used for separate encoding, the resulting sequence of object images through the frames of the video segment may have variable size due to variation in the object size. If this variation is small and the tracking frame is fixed in size, the object bounded by its tracking frame can be directly processed by the PCA. In the case where there exists variation in the size of the tracking frame, and thus in that of the object images, additional processing is required to prepare the resulting images for the input to the PCA-representation module. If the variation is relatively small the largest size of the tracked object can be used to determine the dimensionality of the data vectors to be processed by the PCA.

For all cases presented, the PCA-based compression of video sequences offers the advantage of a high compression ratio due to the optimality of energy compaction for a given number of transform coefficients. The effectiveness of this approach is dependent on the structure (locality) of video data obtained as a result of temporal segmentation and object tracking, as well as on the image resolution at which the algorithm operates.

5 Simulation Results

The original video sequences utilized in the simulations were encoded using the MPEG-2 video compression standard. The size of the video frame was 240 x 352 pixels. The encoding pattern was 'IPBBP', with a GOP of length 12.

The objective of simulations of the PCA-based video representation was to evaluate the allocation of transform coefficients for each frame of the video segments determined as discussed in the previous Sections, and the objective quality of the reconstructed frames of the video sequences. The use of PCA representation in a motion compensation context is part of on-going work. Following the presentation in Sect. 4, a low resolution version of video segments was used for simulations. For simplicity, only the luminance part of the image data was retained. These low resolution video sequences were created by extracting the DC coefficients of blocks from the compressed video frames. Thus, the dimensionality of the resulting DC vectors is 1320×1 (corresponding to DC images of size 30×44). As described in Sect. 3 and 4, the original set of DC vectors comprising the low-resolution video shot is processed by the PCA-based algorithm. The DC images can also be seen as a raw image sequence that are subject to a transform. The video shots utilized fall into three categories ranging from low (shot 1), moderate (shot 2), and high activity (shot 3).

Each DC image in the video shot was also represented using both frame level and block-level DCT transforms. A subset of the DCT coefficients of each DC image was retained either at the frame or block level. For example, for frame level DCT, there are 1320 DCT coefficients corresponding to a 30×44 DC image. A subset of these coefficients can be utilized for a lossy representation of the image, e.g., 900 coefficients. Similarly, at block DCT level, if blocks are taken to be of size 4×4, there are 16 DCT coefficients per block, of each one could retain only the first 9 in plane order. Blocks of size 4×4 yield the best representation performance for DC images (the spatial correlation present in the full resolution images is diminished in their DC representation). The lossy representations of each image in the video shot are then used for its reconstruction, through the inverse transformation.

In the first part of the simulations we use the PCA-based representation of the DC images in a video segment by utilizing a training set of samples. The information needed for reconstruction of the images comprises the M eigenvectors of the representation space, and the M-dimensional representation (coordinates) of each DC image in the determined space. As presented in Sect. 3, 4, a training sample of $M + 1$ DC vectors in each video segment must be selected to enable the TS-PCA representation. If additional information is available about the visual changes taking place in a video shot such as given by a statistical change estimator, the sample can be selected accordingly (see Sect. 4). Once the PCA-representation of the video shot is obtained, the images in the video shot are reconstructed as presented in Sect. 3.

Figs. 4 - 5 show the SNRs of reconstructed DC images in the video shots considered. These images were encoded using a TS-PCA-based representation, and lossy frame and block DCT-based representations of each DC image (identified by the abbreviations DCT and BDCT). The number of coefficients retained for each representation, as well as the total number of coefficients per frame are also indicated in the figures.

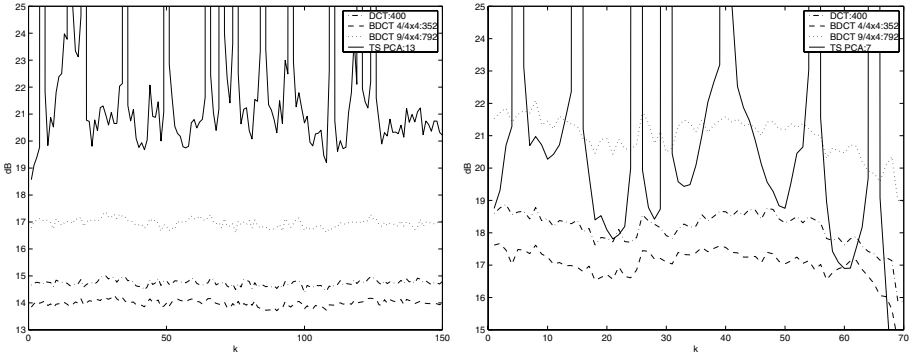

Fig. 4. Reconstructed sequence SNR: Training set PCA - Video shot a) 1, b) 2.

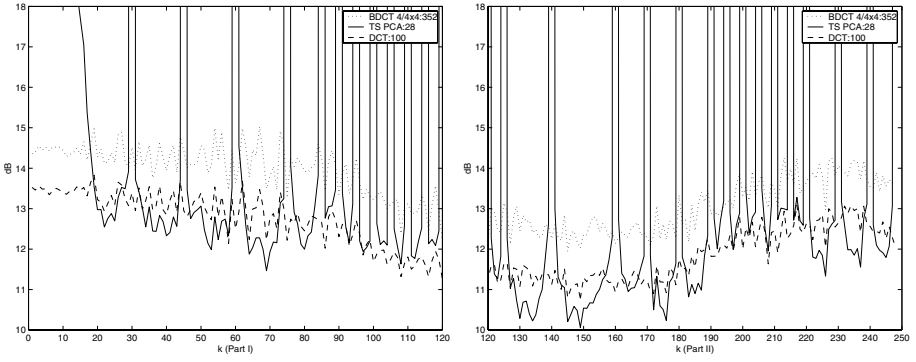

Fig. 5. Reconstructed sequence SNR: Training set PCA - Video shot 3.

A characteristic of the training sample-based PCA (TS-PCA) is that the images that were part of the training sample used for the representation are perfectly reconstructed, and thus are present in the figures as spikes in the SNR. For the video shots with low to moderate visual activity, the performance of the PCA representation is comparable to that of the DCT-based representations using a much larger number of coefficients, as seen in Fig. 4. The video shot in Fig. 5 is more difficult to represent for a TS-PCA-based representation (especially if processed as one shot), due to significant and continuous changes in the visual information. Two approaches can be used to improve performance. The most effective remedy is provided by the appropriate selection of the video segments to be encoded, i.e., the original video shot should be split into two sub-shots which are much more consistent visually. The second method to improve representation performance is to increase the dimensionality of the subspace, with the cost of increasing the number of eigenvectors that have to be transmitted.

An alternative to the training sample methodology is made computationally efficient by the iterative algorithm discussed in Sect. 3. An *initial* estimate of the representation space having the desired dimensionality M is obtained by

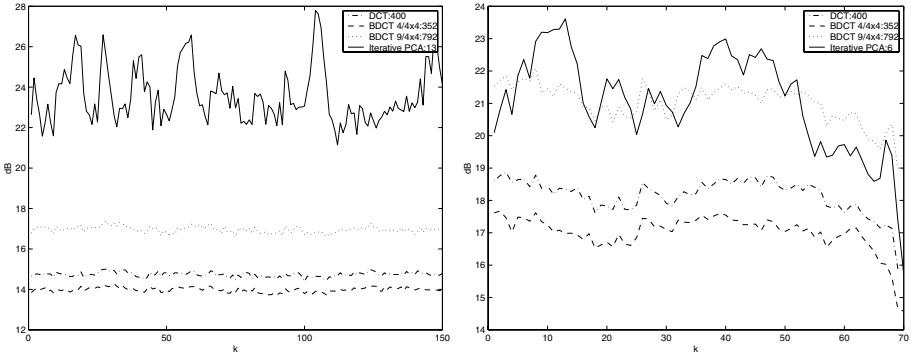

Fig. 6. Reconstructed sequence SNR: I-PCA - Video shot a) 1, b) 2.

computing the most significant M eigenvectors of $M + 1$ images from the video shot by the direct method presented in Sect. 3. Once the initial representation is determined, the algorithm refines the representation space iteratively, with each new image from the video sequence, until all images have been processed. At the end of its operation, the iterative PCA algorithm (I-PCA) produces approximations of the most significant M eigenvectors corresponding to the *entire set* of images.

The SNRs of reconstructed images corresponding to the PCA-based representation using the I-PCA algorithm and the DCT-based representation are shown in Figs. 6-7. The images used to compute an initial estimate of the space are selected to be the first ones in the video shot. Other choices would be a random selection, or a selection based on an activity estimator, similarly to the training sample methodology. As expected, there is a marked improvement in the reconstructed video quality in the case of I-PCA, reflecting a more precise description of the image sequence space. The performance of the representation is also improved for the third video shot in Fig. 7(a), although the previous considerations regarding additional gain that can be obtained through separate processing of its two component sub-shots remain valid. The performance of the I-PCA computed using a spread sample initialization instead of selecting the first frames in the shot for the initial estimation of the eigenspace, is shown in Fig. 7(b) for the last video shot considered. As expected in this case, the SNR is slightly improved compared to Fig. 7(a), using a selection of the sample that spans the entire video shot. For the case of the PCA-based encoding, similarly to intra-coded images in a motion compensation context, the M eigenvectors must be intra-coded for transmission. For all other images in the video sequence, only the $M \times 1$-dimensional vector representing coordinates of the corresponding image in the subspace must be transmitted. For the I-PCA algorithm, M was chosen so as not to exceed the number I of intra-coded images needed for a motion compensated sequence.

As discussed in Sect. 4, by taking into account information regarding the scene contents, the proposed algorithm can be extended to operate at object

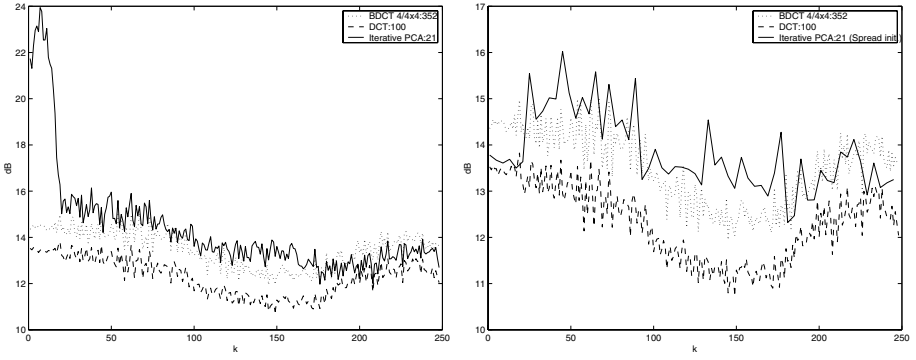

Fig. 7. Rec. SNR (Shot 3): a) I-PCA, b) spread sample I-PCA (at higher SNR resolution.)

level. Through extraction and tracking of moving objects, their PCA-based representation has better performance due to the virtual registration of the objects inside a tracking bounding box. In Fig. 8(a), sample frames show two objects marked by their corresponding tracking frames, which were tracked through the frames of a video shot by the algorithm presented in [3]. Thus, these objects can be extracted and encoded separately using the PCA-based algorithm. Sample original object images and their reconstructed counterparts are also shown in Fig. 8(b). Even though while being tracked the objects suffer a moderate degree of change inside the tracking box, their PCA-based representation and reconstruction is very good.

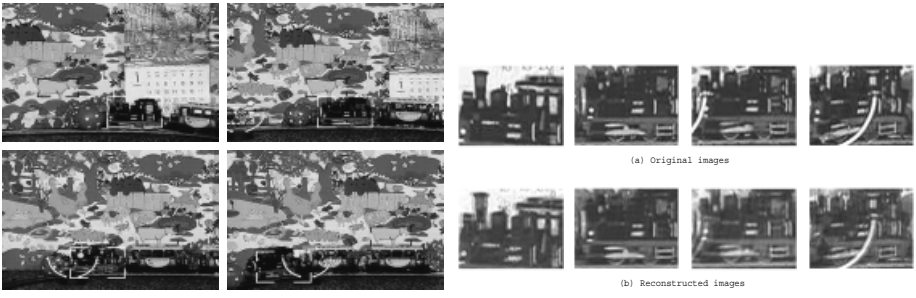

Fig. 8. Snapshots of video tracking for retrieval of multiple templates.

6 Conclusion

In this paper, a new algorithm for the efficient representation of video segments based on Principal Component Analysis was presented. This approach can be used in the context of content-based multimedia processing and retrieval, as well

as a basis for developing video data compression algorithms. For both application domains, the PCA algorithms can be hierarchically applied at both resolution and scene contents level. Simulations of the PCA algorithms show the potential for a very economical representation of video segments, using a small average number of transform coefficients for the images of the sequence. The performance of the approach can be maintained by a temporal segmentation of the video sequence. With increasing resolution of the image sequence, the computational cost of the algorithm increases as well. Depending on the nature of the video segment, a variable number of eigenvectors may be used for representation. It is of interest to determine the optimal number of eigenvectors to be used for each video segment subject to a constrained number of eigenvectors used for the entire video sequence. Also, a hybrid encoder can utilize different encoding methods (PCA-based, motion compensation) depending on the resolution of the image sequence, and degree of visual changes in a video segment.

References

1. Wactlar H., "Informedia-Search and Summarization in the Video Medium," *Proceedings of Imagina 2000 Conference,*Monaco, 2000.
2. Smith J.R., Chang S.F., "VisualSEEk: A Fully Automated Content-Based Image Query System," *Proceedings of ACM Multimedia '96, ACM Press,* Boston, November 1996.
3. Schonfeld D., and Lelescu D., "VORTEX: Video Retrieval and Tracking from Compressed Multimedia Databases-Multiple Object Tracking from MPEG-2 Bitstream," *Journal of Visual Communications and Image Representation(JVCIR'2000),* 2000 Vol.11, No.2
4. Lelescu D., and Schonfeld D., "Real-Time Scene Change Detection on Compressed Multimedia Bitstream Using Statistical Sequential Analysis," *IEEE International Conference on Multimedia and Exposition (ICME2000),*2000 New York.
5. Vetterli M., "Multi-Dimensional Subbband Coding:Some Theory and Algorithms," *Signal Processing,* 1984, Vol.6, No.2, pp.97-112.
6. Woods J.W., "Subband Image Coding," *Kluwer Academic Publishers,* Boston, 1991.
7. Salembier P., Marques F., and Gasull A., "Coding of Partition Sequences," *Video Coding: The Second Generation Approach (Torres L. and Kunt M., eds.) Kluwer Academic Publishers,* Boston, 1996.
8. Katsaggelos A. K., Kondi L.P., Meier F.W., Ostermann J., Schuster G.M., "MPEG-4 and Rate Distortion Based Shape-Coding Techniques," *IEEE Proceedings Special Issue on Multimedia Signal Processing,* 1998, Vol.86, No.6, pp.1126-1154.
9. Archer C., Leen T.K., "Optimal Dimension Reduction and Transform Coding with Mixture Principal Components," *Proceedings of the IEEE International Joint Conference on Neural Networks,*, Washington, 1999.
10. Meinicke P., Ritter H., "Local PCA Learning with Resolution-Dependent Mixture of Gaussians," *Proceedings of 9th International Conference on Artificial Neural Networks (ICANN'99),* 1999, Edinburgh, UK, pp 497-502.
11. Murakami H. and Kumar V., "Efficient Calculation of Primary Images from a Set of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1982, Vol. PAMI-4, No. 5, pp. 511-515.