Content Analysis of Video Using Principal Components

Emile Sahouria Avideh Zakhor
EECS Department, University of California, Berkeley, CA 94720
email: {emile,avz}@eecs.Berkeley.EDU

Abstract

We use principal component analysis (PCA) to reduce the dimensionality of features of video frames for the purpose of content description. This low dimensional description makes practical the direct use of all the frames of a video sequence in later analysis. The PCA representation circumvents or eliminates several of the stumbling blocks in current analysis methods, and makes new analyses feasible. We demonstrate this with two applications. The first accomplishes high level scene description without shot detection and key frame selection. The second uses the time sequences of motion data from every frame to classify sports sequences.

1 Introduction

The essential goal of video content analysis is to represent the visual data in video in a way that allows meaningful and efficient classification, indexing and retrieval of objects in a video database. An increasingly important goal has been to develop analysis techniques uniquely suited to the time-varying nature of video, rather than relying on the ad-hoc application of still image techniques. We demonstrate a representation based on Principal Component Analysis (PCA) that allows one to fully use the temporal dimension. We also describe the application of this representation to high level structure analysis and characterization of long video segments via motion.

The most common techniques for describing video rely on first detecting transitions between shots, and then selecting a single representative frame from each shot and applying some still picture analysis methods to it. Others have clustered these key frames to provide the user

with a hierarchical representation of a database of video clips that is amenable to browsing. In particular [19] used graphs to represent the temporal relationships between scenes. The camera motion motion and gross statistics of motion in a given shot have also been used as descriptions of shots [8].

Object-based representations such as [2], in which one applies some sort of segmentation and/or tracking to find potentially moving objects, are ways of more fully utilizing the spatial-temporal nature of video. [2], [4] and[13] all detected and indexed the trajectories of objects within shots. These approaches are low-level and local in nature, however, and only provide indexing capabilities for rather precise queries. Also, they provide access to video clips of a very small time scale. A few recent methods have actually provided facilities for analyzing video on a larger time scale for the purposes of classification and navigation of databases of longer pieces of video. [8] provide a global, probabilistic description of video, while [17] and [7] use local motion and shot length to characterize long segments of video.

A common element in all of these approaches is the reduction of the massive amount of data present in video to a manageable form, either by use of motion analysis, shot cuts and key frames, or more ad-hoc and limited approaches. These methods discard or simplify much of the potentially useful data in video, and in doing so, reduce the power of the specific description technique. This work generalizes these efforts by using the classical method of principal components analysis (PCA) to describe video in a very low dimensional space. Such a description can then be used to build useful analysis, indexing and classification applications. This formalization of data reduction allows massive reductions in computational and storage complexity that previously rendered more sophisticated analysis schemes untenable, while retaining the spatio-temporal data that ad-hoc techniques may inadvertently throw away.

This paper begins with a brief history of PCA in the analysis and classification of data. We use this to motivate our use of the method as a way of describing video in section 3. In section 4 we describe the two applications that use this description: one implements high level structure analysis without shot detection and key frame selection, and the second uses time sequences of motion data to classify sports sequences.

2 PCA and Content Analysis

PCA has long been used in the field of pattern recognition. Applications to content-based databases include the use of eigenfunctions for face representation in the *Photobook* project [11] and for feature space reduction in image databases . PCA has also found use in more traditional fields–creating "feature spaces" for text databases. Latent semantic indexing (LSI) used the singular value decomposition (SVD) to find a low dimensional basis for the space of histograms of keyword occurrences in text documents [5]. "Pattern-space" gesture recognition has also been investigated [10]. PCA and its approximations have even been used in shot change detection [15, 9].

The PCA method reduces the dimension of the feature space and reveals relationships between objects that facilitate searches by similarity. Many intuitive explanations and descriptive examples of this result have been given [5]. Essentially the method assumes a relationship between data similarity and concentrations of energy in the eigenspace that spans the data.

While the validity of the last statement is arguable, the dimensionality reduction property alone motivated us to apply the technique to video for the purposes of content representation. Aside from formalizing the data reduction, the consideration of an entire video sequence in forming a description of that sequence may reveal long-range or subtle relationships that are potentially missed by other methods.

3 Describing Video via PCA

The strategy for describing video using PCA is to condense local spatial information using the SVD, and to preserve the temporal information by keeping all such reduced spatial information for all frames. A low-level feature vector is derived from each data unit, which may be a frame or small group of frames in the video. These vectors are stacked to form a matrix, and the SVD is used to find the small dimensional subspace that best represents the vectors. Further, the features to which PCA is applied should be "aligned," in that they should have similar statistical behavior across time. For instance, the pixels in the video frames themselves do not meet this alignment criteria, because small camera motions will decorrelate pixels at the same screen location in different frames, even if the frames are in the same video shot.

These criteria led us to use single frames as our data units, and color histograms and motion vectors as features. Further, the video used as input to the system is compressed using a standard MPEG-1 compressor. To simplify processing, only P-type, or forward motion predicted, frames are used in the analysis. Motion vectors are easily extracted by inverting the entropy code and parsing the MPEG bitstream. Color histograms can be obtained either by fully decompressing the bitstream, or by using subsampling techniques to reduce computational complexity, such as producing DC sequences [18] and extracting the histograms from them.

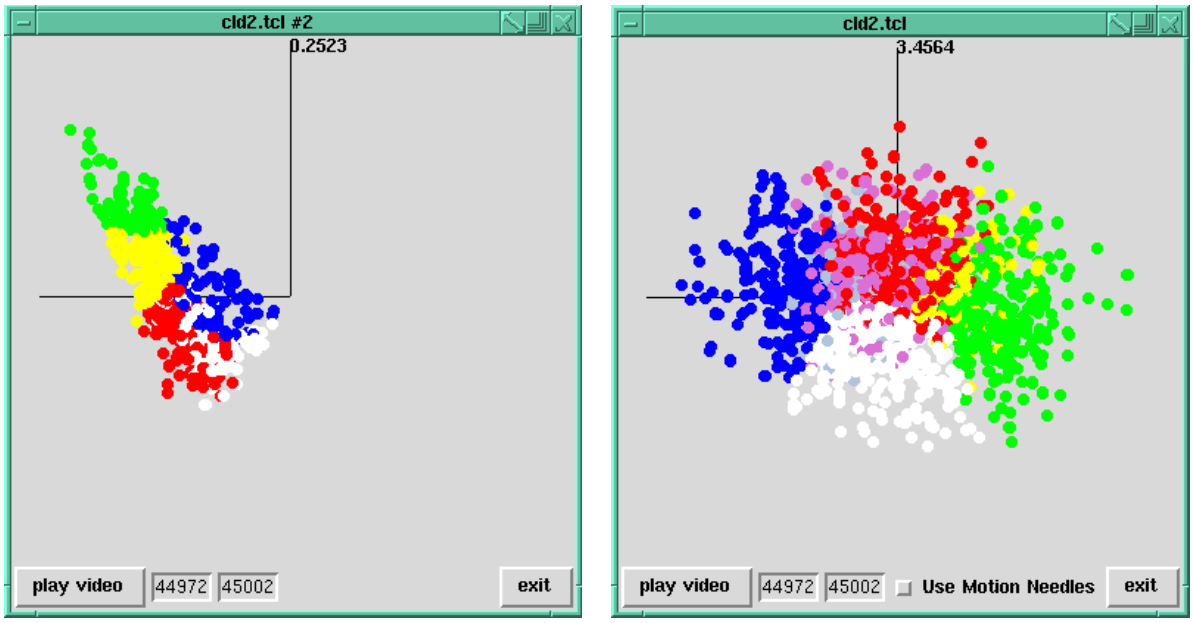
Once the features are extracted from the video data units, they are reordered into row vectors and stacked into a matrix. The matrix has one row for each video frame, with the row size equal to the size of the frame feature. The matrix is decomposed using the SVD, and a small number of basis functions is chosen to represent this feature space, which is the same as the row space of the matrix. These are the first k right singular vectors, where k is the chosen dimension of the basis. Finally, the representation of each frame is its projection onto this basis. For the purposes of the applications described below, this small number of coordinates, along with the basis, fully describes the frame.

The matrices of feature vectors are large enough that taking their SVD might be too computationally complex. The CLAPACK software package [3] contains algorithms for computing the full SVD, but with large memory and time demands. An alternative to this is to use Lanczos subspace algorithms, such as the one found in the freely available SVDPACK [1], for computing only a few singular vectors of sparse matrices. Motion fields and color histograms tend to produce such sparse feature matrices, so those methods are applicable. One may also randomly subsample the original video sequence to obtain a subset of the feature vectors on which to train the basis; this is the approach used in the applications described below. Once the basis is found, all frames of the sequence may be projected onto it.

Kobla et al. [9] used a similar subspace decomposition to analyze shot transitions. However, a random dimensionality reduction technique was used to limit complexity. The resulting "random basis" limits the use of the representation. One shortcoming is that two sequences with slightly different features may give rise to very different bases, which do not necessarily span the same space. The eigenspaces for such sequences should span nearly the same space.

While the coordinate description of a video frame is merely an intermediate step in the full

analysis of a video sequence, observing the results of the description is insightful. Figure 1 shows the coordinates of 2000 video frames from a recent movie along the first two principal components found by the technique described above. The left figure is a description based on bin color, while the right is one based on motion. The color features are 256 bin color histograms derived from DC sequences; a 9×7 array of motion vectors per frame is used as the motion feature. 500 randomly selected frames were used to compute the basis. Each point represents a frame, and the color of each point is a psuedocolor representing cluster assignment by the ISODATA [16] algorithm. Similar segments of video, either in the color or motion sense, lie close to each other in these spaces. Clips from the upper left of the color space in Figure 1 are very dark, while those a the bottom are bright outdoor scenes, for instance.



(a) (b)

Figure 1: *(a) Color and (b) Motion coordinates.*

The principal component representation of the frames versus time provides another interesting view of the data. A "time-line" of frames and their relationships to each other according to an ISODATA clustering appears in Figure 2. A high level temporal structure is evident, and will be discussed more in the next section.

Figure 2: *Frame cluster membership versus time in a video sequence. Note the higher level "story" structure.*

4 Applications of the Principal Components Description

Two applications that demonstrate the power and versatility of the PCA technique are now presented. The first implements scene analysis, a problem that is currently treated using the techniques of shot change detection and key frame selection. The second application classifies sports sequences based on motion.

4.1 Scene analysis

Scene change detection is the process of finding transitions between video *scenes*. Shots are sets of contiguous frames between cuts, fades, wipes or large camera motions, and scenes are groups of shots which exhibit some consistency in the context of the plot of the video. They may be composed of many different shots, or alternate randomly between a few shots. Yeung and Yeo [19] demonstrated a scene analysis method based on Scene Transition Graphs. It uses the time-constrained clustering algorithm to cluster the frames in a video sequence such that shots greatly separated in time fall into different clusters, even if they have similar spatial or other local descriptions. By considering the time sequence of these clusters as they appear in the video, one may construct a directed graph that consists of a chain of transient classes. Each transient class is separated from the others by a cut edge of the graph, and generally corresponds to a change in scene. Yeung and Yeo characterized each shot by the color signatures of one or more key frames, and applied the clustering to the shots.

We propose to bypass the shot detection and key frame selection by describing each frame by its coordinates in principal component space. We represent each frame by the projection of a 256 bin color histogram, derived from a DC sequence, onto a four dimensional subspace. We then apply the time-constrained clustering directly to the projections. This is illustrated in Figure 3. A sample scene transition graph, representing an hour of video, that was generated by this new approach appears in Figure 4. Several of the cut edges are marked. Since shot

transitions are not explicitly detected, the algorithm is able to recognize scene structure despite gradual transitions and large camera motions by using temporally disparate data. Forcing a shot segmentation in these regions of gradual change may obscure data necessary to the clustering, while the present technique essentially delays all decisions until all data can be used simultaneously. This is especially appropriate if one is interested not in precise shot boundaries, but in large-scale structure. The results in Figure 5 demonstrate operation on material with a great deal of camera motion and gradually changing scene content. The figure shows scene cuts determined by a human viewer and those detected by the algorithm in one half hour of a movie. Two of 23 scene changes are missed, and fifteen false positives exist due the fact that humans are able to group together perceptually similar shots, even if the graph of these shots contains many cut edges.

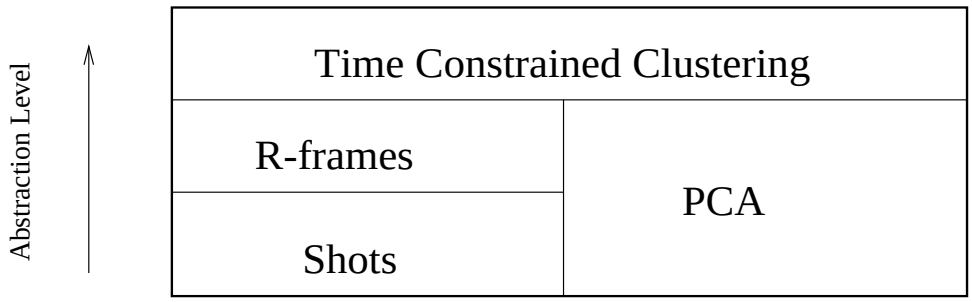


Figure 3: *Alternate routes to scene change detection.*

We also note that the performance is comparable to the method of[19]. A comparison between the two techniques for a subset of the test data appears in Figure 6. The figure shows scene cuts detected by both methods, and the cuts determined by a human viewer for 10,000 frames, or approximately 11 minutes of the movie in Figure 5. For this time period, our proposed scheme results in 2 misses, 7 hits and 7 false alarms, while the system in [19] results in 3 misses, 6 hits and 6 false alarms.

4.2 Sports Classification by Motion

A second application under development characterizes entire videos by motion. Motion is a highly desirable feature in that it possesses a substantial degree of invariance across sequences to color and lighting, and to a lesser extent, to scale. As an illustrative example, we have

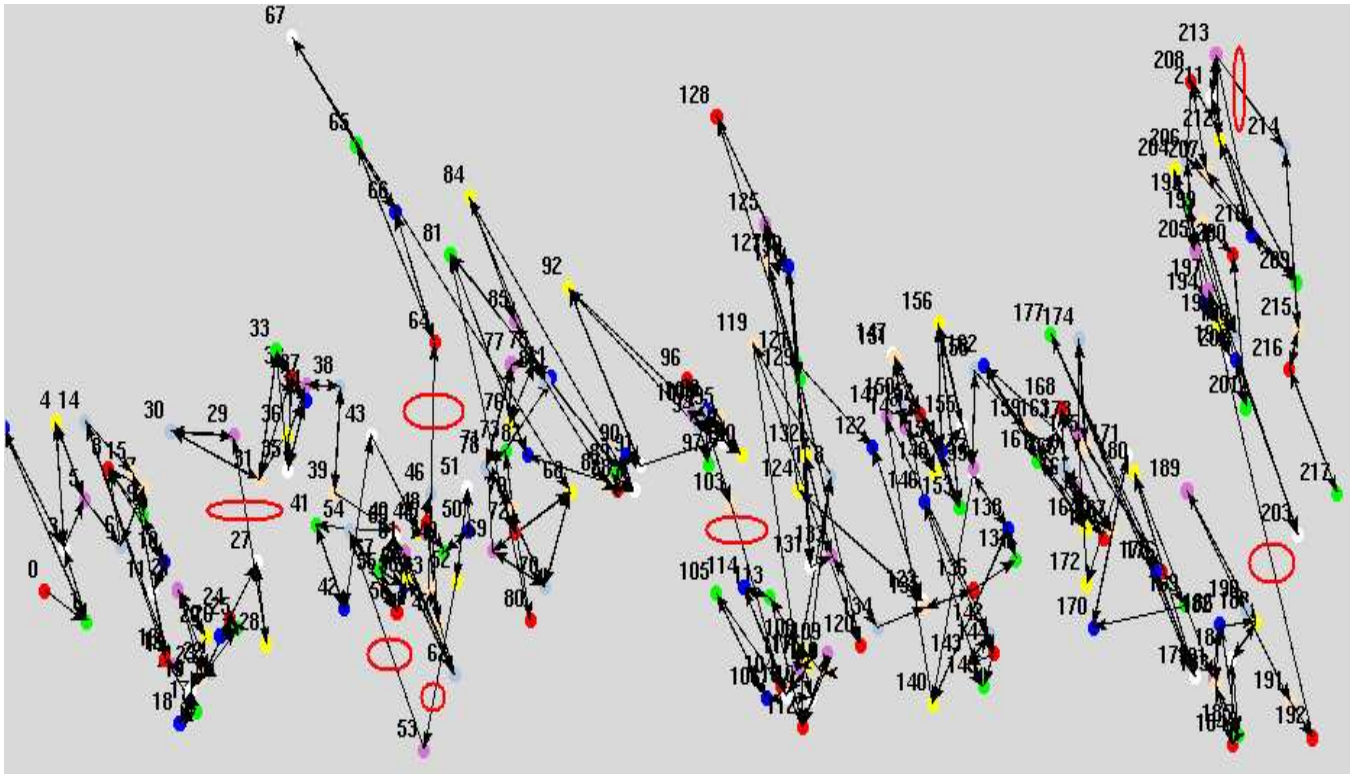


Figure 4: *A sample scene transition graph. Several cut edges are circled.*

implemented a simple sports classification system. Three classes are assumed: basketball, ice hockey and volleyball. While one may argue that cues other than motion could more easily differentiate these classes, we claim that these other techniques are not as generally applicable as motion, and that some of them may not be as easily used in the compressed domain. Hence, only MPEG-1 motion vectors from P-frames are used in the classification. The edge vectors of a 20×15 macroblock frame are discarded and the resulting motion field decimated by two in each dimension, leaving 63 vectors per frame. These motion vectors are projected onto a "typical" basis, resulting in a sequence of eigenspace coordinate vectors for each video sequence. These reduced sequences are then classified by two techniques–one which considers only time-average behavior of motion fields, and another which also considered temporal relationships between elements of the observed sequence.

While in the previous application, a new basis was constructed for each video, it suffices to use a single basis to characterize the frames of motion vectors in sports sequences. We reached

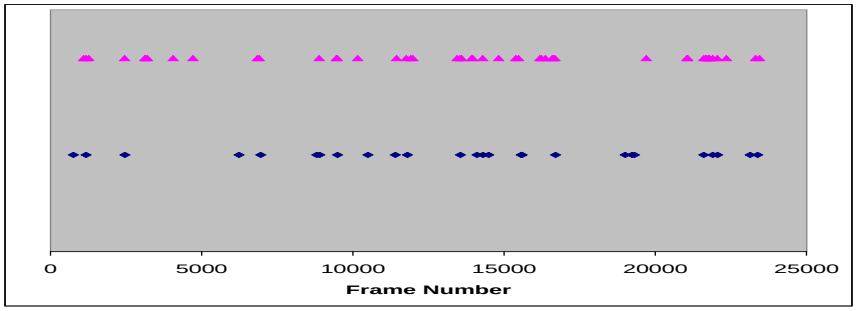


Figure 5: *Scene change evaluation. Round markers are the cuts determined by human; the x's show how the machine sees it.*

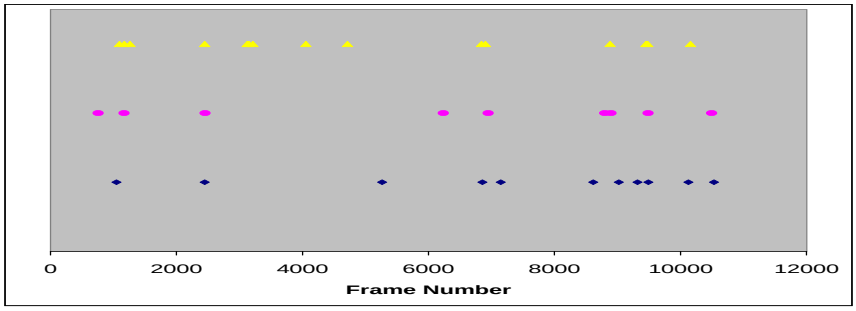


Figure 6: *Scene change method comparison. Top: Current method. Middle: "ground truth". Bottom: Method of Yeung and Yeo.*

this conclusion by observing the bases designed for several sports sequences. Two things were apparent: the sequences contained mainly camera motion, and the "principal motions" were very similar across the bases of the different sports. Figure 7 shows the three largest principal components of one such basis. Note the simple interpretation in terms pan, zoom and tilt. The projections onto this basis effectively provide approximations to the dominant camera motions between two frames of video. For each of the techniques below, a two-dimensional eigenspace was used to perform the classification. A quick examination of the projections of the sports sequences onto this basis shown in Figure 8 indicates a clear distinction in the statistics of each class. As an example, hockey shows rapidly changing motions mostly of small amplitude with

periods of extended motion, while volleyball exhibits short duration, large magnitude motions in one dimension.

The techniques were tested on two data sets. Set 1 comprises 116 sequences, 50 taken from two basketball games, 16 from two ice hockey games and 50 from two soccer games. Further, the sequences were edited to contain only the play of the sports. For instance, no crowd scenes or timeouts are included. This is a somewhat limited test set, although it proves the functionality of the methods. Set 2 contains more realistic examples. It consists of a total of fourteen news highlight sequences of the three sports, almost evenly divided between the sports. Each sequence includes non-relevant segments such as crowd scenes, fights and graphics overlays. For each of the two data sets, some sequences from each sports class were used to train the classifier, and the rest were used only for validation.

The first technique used to classify the sequences characterizes the long term motion statistics by vector quantization. The 10-bin minimum mean-square energy quantizer was found for each training set. Every sequence in the sample sets was then quantized by the quantizer for each class, and the average distortion was recorded as a distance measure from the class. The quantizer with the lowest distortion was determined to be the class to which the sequence belonged.

The second classifier uses the local temporal statistics in addition to average magnitudes by using Hidden Markov Models (HMMs) [12]. HMMs are used to analyze correlated sequential data; in particular, they have been used with great success in speech and gesture recognition [12, 14]. Each element of the data sequence is considered to be a random function of the state of an underlying Markov chain. The standard problems in HMM analysis are to estimate the markov models, evaluate the likelihood that an observed sequence was generated by a given model, and estimate the most likely sequence of states that produced an observation sequence. Solutions to these problems allow, among other things, the classification and recognition of data sequences. A primary concern in HMM analysis is computational complexity; the complexity of the problems outlined above depends linearly on the dimension of the observation space, for instance.

PCA on motion fields has been used to generate the observed features for HMM analysis in gesture recognition [6]. More recently, [7] used very simple HMMs to classify video sequences

based on shot length and average motion activity. The rich, yet compact description afforded by PCA offers hope of using HMMs to more precisely analyze the video. In such a scheme, the coordinates of each frame in the principal component space are the observations of the Markov process. In the present application, one continuous observation density markov model is trained for each sports class. Each sequence in the test set is evaluated against each model, and the sequence is assigned to the class with the highest likelihood of having produced the sequence.

Table 1 shows the results for each classifier on the two sample sets. The table lists the total number of sequences for all sports in each test set, and the number of sequences used for training. The last four columns indicate the numbers of wrong classifications in the training sets and remaining sequences, for each of the two classification methods. The performance of the VQ approach is worse for the both sets. This implies that while the marginal, i.e. time average, statistics were very similar between the classes, the dynamics of the observation process were sufficiently distinct to allow a better classification.

Set		Total Size	Training Set Size	VQ		HMM	
				Train	Test	Train	Test
1	bb	50	4	2	21	0	5
	nhl	16	4	0	1	0	0
	wc	50	4	0	5	0	1
2	bb	5	2	2	2	0	1
	nhl	5	2	0	0	0	0
	vb	4	2	0	0	0	1

Table 1: *Set sizes and misclassifications.*

5 Discussion

We have demonstrated a way to fully use the temporal nature of video for content analysis. The analysis depends on having a practical description of time varying data for all frames; we use principal components analysis to achieve this goal. Once one has this description, applications can be developed which overcome the limitations of other approaches, and which make new applications possible. We provide two example applications as evidence of this

claim. An important outcome of their development is the realization that time-series analysis is potentially a very useful tool in content-based analysis. As noted above, the use of these methods is gradually filtering in from other fields such as speech and gesture recognition. The use of more sophisticated time-series methods will undoubtedly add power to the methods described above. A prime example is the treatment of background or noise segments of the signal, exemplified by crowd scenes in sports sequences. The sequence analysis need not be limited to motion, or even a single feature alone. Only further development will reveal the limits of such methods.

References

[1] M. W. Berry. Svdpack. http://www.netlib.org/svdpack.

[2] S.-F. Chang et al. Visual information retrieval from large distributed online repositories. *Comm. ACM*, 1(1), 1996.

[3] J. Demmel et al. Lapack. http://www.netlib.org/lapack/.

[4] N. Dimitrova and F. Golshani. Motion recovery for video content classification. *ACM Trans. on Information Systems*, 13(4):408–439, 11 1995.

[5] S. Dumais et al. Indexing by latent semantic analysis. *J. Am. Soc. Information Science*, 41:391–407, 1990.

[6] Y. Iwai, T. Hata, and M. Yachida. Gesture recognition based on subspace method and hidden markov model. In *IROS 97*, volume 2, pages 960–966, September 1997.

[7] G. Iyengar and A. Lippman. Models for automatic classification of video sequences. *SPIE*, vol. 3312, pp. 216–227, 1994.

[8] G. Iyengar and A. B. Lippman. Videobook: An experiment in characterization of video. *ICIP*, vol. 3, pp. 855–58. 1996.

[9] V. Kobla, D. Doermann, and C. Faloutsos. Developing high-level representations of video clips using videotrails. In *SPIE*, volume 3312, pages 81–92, 1998.

[10] S. Nagaya, S. Seki, and R. Oka. A theoretical consideration of pattern space trajectory for gesture spotting recognition. In *Proc. Second Intl. Conf. on Automatic Face and Gesture Recognition*, pages 72–7, 1996.

[11] A. Pentland, R. Picard, and S. Scarloff. Photobook: Tools for content-based manipulation of image databases. In *SPIE Conference on Storage and Retrieval of Image and Video Database II*, volume 2185, 1994.

[12] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[13] E. Sahouria and A. Zakhor. Motion indexing of video. In *ICIP*, volume 2, pages 526–529, 1997.

[14] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proc. Intl. Symposium on Computer Vision*, pp. 265–270, November 1995.

[15] A. Tewfik and K. Han. Eigen-image based video segmentation and indexing. In *ICIP*, volume 2, pages 538–541, 1997.

[16] C. Therrien. *Decision, Estimation and Classification*. Wiley, 1989.

[17] N. Vasconcelos and A. Lippman. Towards semantically meaningful feature spaces for the characterization of video content. In *ICIP*, volume 1, pages 25–29. IEEE, 1997.

[18] B.-L. Yeo and B. Liu. A unified approach to temporal segmentation of motion jpeg and mpeg compressed video. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 81–88, May 1995.

[19] M. M. Yeung and B.-L.Yeo. Time-constrained clustering for segmentation of video into story units. In *Proceedings of the ICPR*, pages 375–380, 1996.

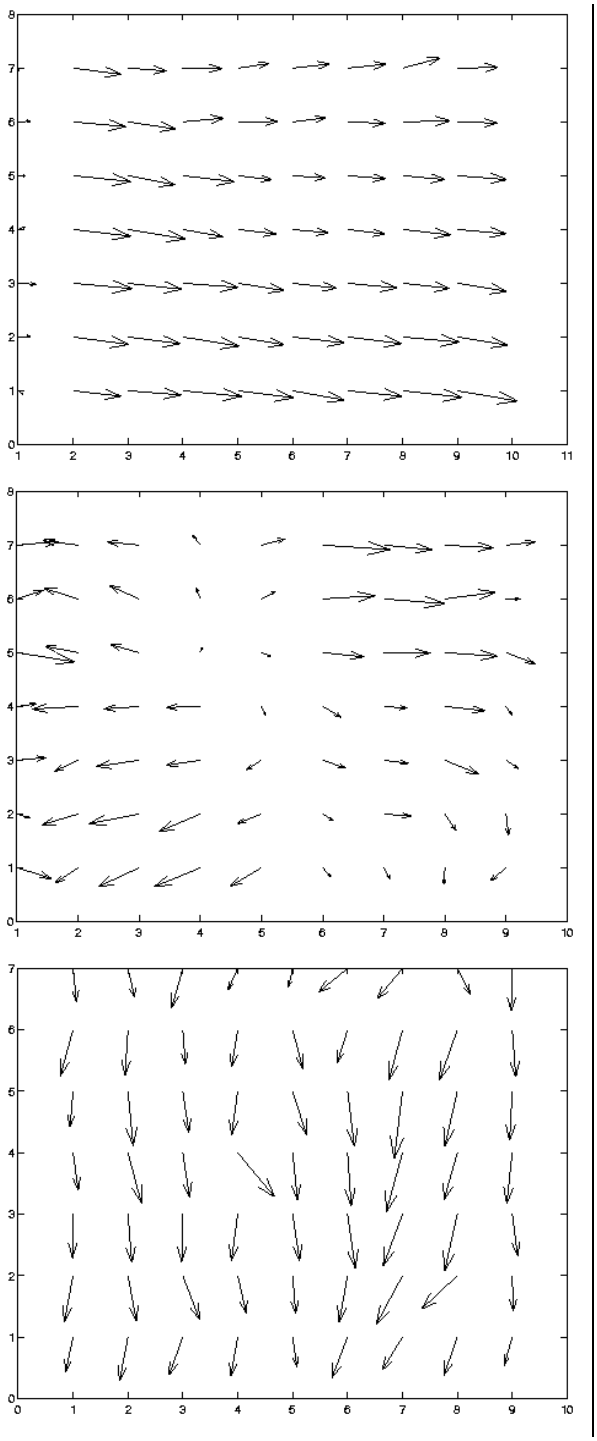


Figure 7: *The top three principal components for a sports sequence.*

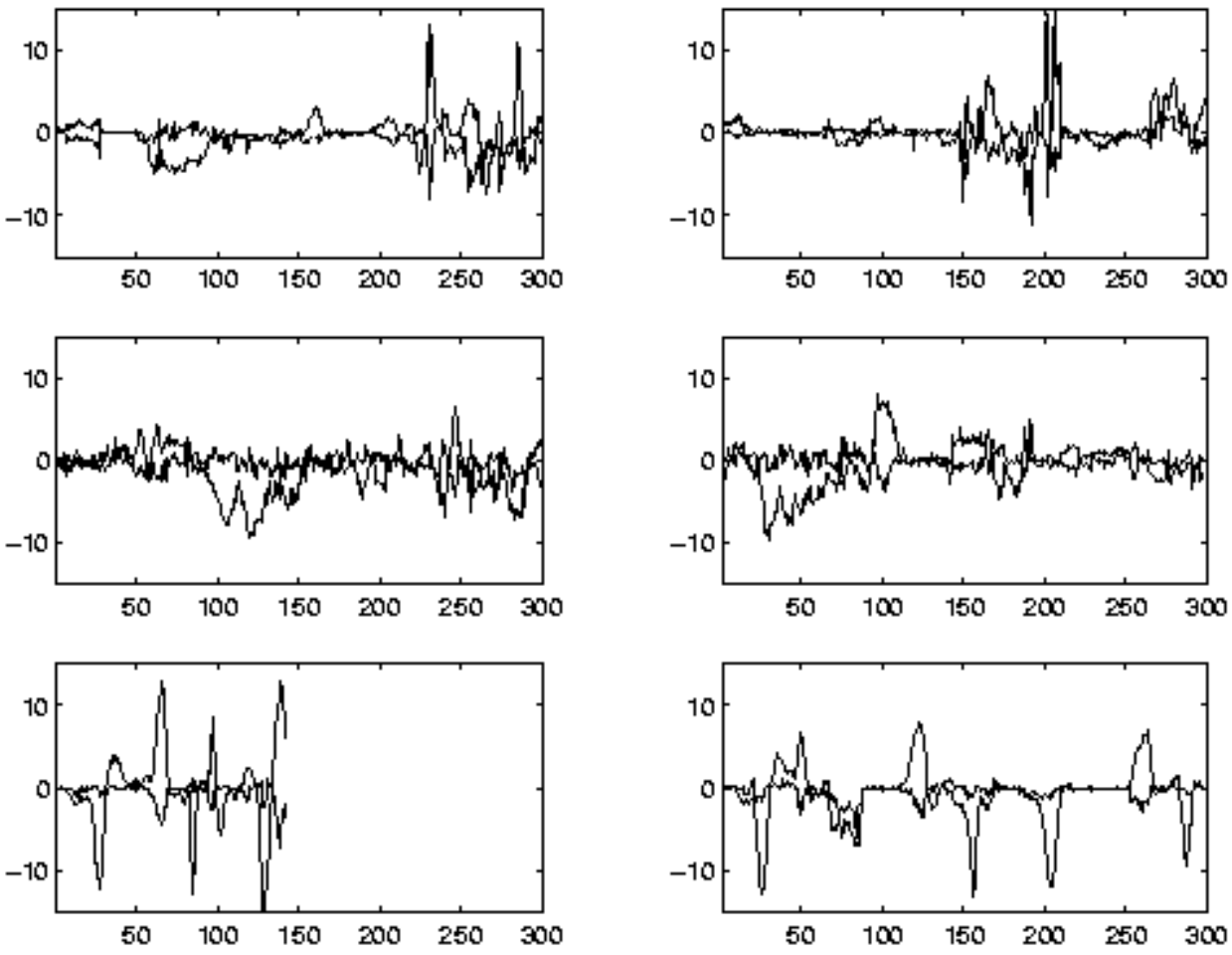


Figure 8: *Top two eigenspace coordinates versus time for two sequences of each sports class. Top: basketball. Middle: hockey. Bottom: volleyball.*