Fast and Reliable PCA-Based Temporal Segmentation of Video Sequences

Jiří Filip
School of Math. and Computer Sciences
Heriot-Watt University
EH14 4AS Edinburgh, Scotland
filipj@macs.hw.ac.uk

Michal Haindl
Institute of Information Theory and Automation
of the ASCR
182 08 Prague 8, Czech Republic
haindl@utia.cas.cz

Abstract

With significantly increasing number of archived movie sequences a need of their automatic indexation and annotation is raising. Robust and fast temporal segmentation of video sequences is one of the challenging research topics in this area. In this paper we propose a new temporal segmentation method of the video sequences based on PCA approach. Contrary to standard approaches based on histogram or motion field analysis the proposed method does not require any such a complex analysis. The method starts with sparse greyscale sampling and eigen-analysis of input sequence. A sum of absolute derivatives of temporal mixing coefficients of main eigen-images is then used as cuts detection feature, while dissolve transitions are detected by means of coefficients' specific behaviour. The functionality of the method was successfully tested on number of sequences ranging from artificial set of similar dynamic textures to professional documentary movies. Although, the results may not be unexpected, we believe that proposed method provides novel, very fast and reliable way of movie cuts detection.

1 Introduction

Recent increasing accessibility of tools for creating and editing of digital movies, results in tremendous increase of stored digital video data. Furthermore, a vast number of digitalized movies are stored in archives. To allow us to effectively search in these massive data, methods of automatic indexation, annotation, and retrieval becomes essential. Automatic video indexation method should be able to divide the video sequence into a set of scenes an further to individual shots, while providing user with corresponding representative information. Two video shots are typically divided either by simple cuts (abrupt change of two shots) or means of more complex gradual effects as dissolves of two shots (see Fig. 1), fades, wipes etc. As this topic is very timely today there is significant research effort involved in solving of this task as it is apparent from available survey papers [6, 7]. The core of all published methods is a temporal segmentation algorithm, by which we can roughly group them into several categories. The first of them is based on some kind of pixel-wise inter-frame difference [13]. As this approach is susceptible to noise, its block-based modifications were implemented. The second group of methods is based on inter-frame histogram comparison [13]. As this approach does not take into account any spatial distribution of pixels over the frame, its modification were suggested dividing each frame into several regions for local difference computation [8]. While the above mentioned methods are popular and relatively computationally efficient in cuts detection, they often fail in detection of gradual transitions in the sequence To solve this a next group of method emerged based on motion detection in the analysed sequence. These method were based either on block-wise motion detection [11, 10] or on motion field analysis [2]. While the former compute block-wise pixel difference of motion compensated blocks the latter use inverse of motion smoothness as a frames disparity measure. There are also approaches available based on unsupervised temporal clustering [4], on examining spatial distribution of edge pixels [12], or on video model-based techniques [5]. The motion and model based methods have generally better performance than the pixel or histogram-based methods mainly in detection of gradual changes, however, their computational demands are due to using of motion compensation much higher. On the other hand, motion-based methods should not misinterpret a slow camera motion as a shot transition. In this paper we propose novel, fast, and reliable PCA-based approach for temporal segmentation of movie sequences. The principle of the method is outlined in Section 2. Section 3 shows results of the method and discuses its advantages and limitations. Section 4 concludes the paper.

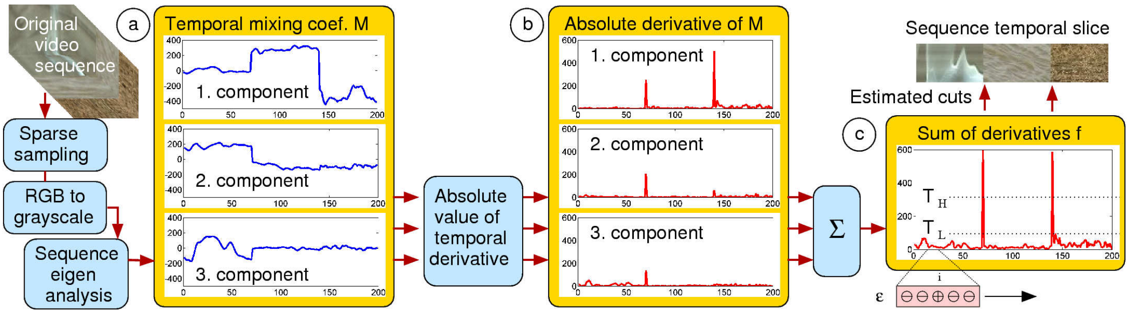


Figure 2. Scheme of the proposed cuts detection method.

Figure 1. Example of dissolve transition.

2 PCA-Based Temporal Segmentation

The scheme of the proposed segmentation method is shown in Fig. 2. The method starts with sparse spatial sampling of individual frames from the input sequence, since our experiments have shown that only a very limited number of pixels is required for the proper functionality of the method.

Eigen-analysis: Selected RGB pixels of individual images from the video sequence are converted to greyscale and arranged into normalized column vectors forming matrix $\mathbf{C}$ $(n \times t)$ where n is number of pixel values $n = MN$ depending on the image resolution $M \times N$, t is a number of colour frames and $\mu_{\mathbf{C}}$ is a mean image of the sequence. Matrix $\mathbf{C}$ is a subject of eigen-analysis that follows. From the matrix $\mathbf{C}$ a covariance matrix $\mathbf{A}$ $(t \times t)$ is created including spatial and spectral correlation of the sequence according to $\mathbf{A} = \mathbf{C}^T\mathbf{C}$. The resulted matrix $\mathbf{A}$ is decomposed using singular value decomposition. $\mathbf{A} = \mathbf{UDU}^T$ [3] where $\mathbf{U}$ is orthogonal matrix of eigenvectors and $\mathbf{D}$ is diagonal matrix of corresponding eigenvalues sorted in ascending order. From matrix $\mathbf{U}$ only a k of eigenvectors $(k = 3)$ are preserved in the matrix $\tilde{\mathbf{U}}$ corresponding to eigenvalues bearing the most of the information. Using $\hat{\mathbf{U}} = \mathbf{C}\tilde{\mathbf{U}}\tilde{\mathbf{D}}$, where $\tilde{\mathbf{D}} = diag\{\sigma_1^{-\frac{1}{2}}, \ldots, \sigma_k^{-\frac{1}{2}}\}$, we obtain the matrix of eigen-images $\hat{\mathbf{U}}$ ordered into k columns of the length n. Finally the progress of temporal mixing coefficients of individual eigen-images $\hat{\mathbf{U}}$ for all frames from the original sequence is computed using $\mathbf{M} = \hat{\mathbf{U}}^T\mathbf{C}$. Only the matrix $\mathbf{M}$ $(k \times t)$ is a subject of the further processing.

Scene Cut Detection: The coefficients of first k principal components contain most of the information about sequence dynamics and contents change (see Fig. 2-a). The main temporal changes in these coefficients, i.e., cut points, can be obtained by absolute value of coefficients' derivative approximated as temporal difference (see Fig. 2-b). Finally, the absolute values of the derivative coefficients are summarised (see Fig. 2-c) to obtain the proposed discriminative feature f as follows

$$f(i; i = 1 \ldots t) = \sum_{j=1}^{k} |\mathbf{M}(j, i+1) - \mathbf{M}(j, i)| \ . \quad (1)$$

To reliably determine cuts in the sequence we suggest to use two thresholds

$$T_H = c_H \max_{i=1\ldots t} f(i) \qquad T_L = c_L \max_{i=1\ldots t} f(i) \qquad (2)$$

where c_H, c_L are two user-defined constants from an interval $(0 \ldots 1)$. We mark frame i directly as a cut point if $f(i) > T_H$. On the other hand, we disqualify the point i from being a cut point when $f(i) < T_L$. While T_H guarantee that none important coefficient change is missed, T_L prevent relatively minor changes to be considered as a cut. If a value $f(i)$ lies in-between T_L and T_H, a contextual neighbourhood ε (see Fig. 2-c) at the point i is evaluated using

$$\varepsilon(i) = f(i) - f(i-1) - f(i+1) - f(i-2) - f(i+2) \ . \quad (3)$$

Such a point is marked as the cut when $\varepsilon(i) > 0$, i.e., when feature f has a distinctive peak at position i. Although an optimal setting of the values c_H, c_L can depend on a temporal dynamics of the analysed sequence, all results in this paper were obtained using $c_H = 0.5$ and $c_L = 0.15$.

The proposed method works well for the sequence analysed in one stroke, however, almost every movie sequence contains tens thousands of frames. The reliable temporal segmentation of sequences of such a length using the proposed approach would require enormous processing times and a high number of the preserved principal components. To avoid this situation we suggest a moving window extension as it is shown in Fig. 3. The principle of the extension rests on subdivision of analysed sequence into a set of overlapping windows. The contents of the windows are analysed independently, possibly simultaneously, in a way described above. The only addition is that the feature f of the

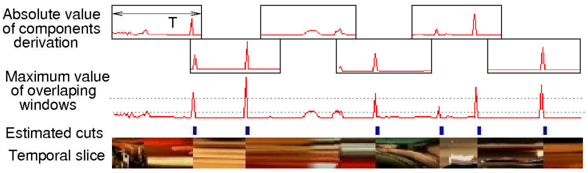


Figure 3. The proposed moving window extension.

whole sequence is obtained as maxima of features from the overlapped parts of the windows (see Fig. 3). The cuts are obtained from the feature f in the same way as explained above. The proposed approach allows us to gain significant increase in processing speed as the whole sequence is processed linearly in time.

Dissolve Transition Detection: An analysis of the mixing coefficients in individual temporal windows has revealed that a typical dissolve shots transition (see Fig.1) is accompanied by a smooth monotone change in first component's coefficient in significantly higher span than in the remaining components' coefficients (see Fig.4). The candidate of dissolve transition v is found as a maximum variance ρ_p of first temporal coefficient $M(1,.)$ in local neighbourhood p over the size of temporal window of length T, i.e., $v = \max_{i=1...T} \rho_p(M(1,i))$. This candidate is marked as a dissolve transition when $s_p > c_p s_1$, $s_1 > c_s s_2$, $s_1 > c_s s_3$ and $M(1,.)$ shows significant monotonicity in neighbourhood p at position v, i.e., $\max(dM^+/dM^-, dM^-/dM^+) > c_m$, where dM^+, dM^- are sums of positive and negative values in temporal derivative $d(i) = M(1,i+1) - M(1,i)$. Values $p = 20$ frames, $c_m = 10$, $c_p = 0.7$, and $c_s = 1.5$ were used throughout our experiments.

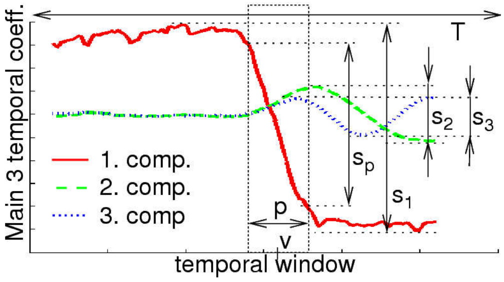
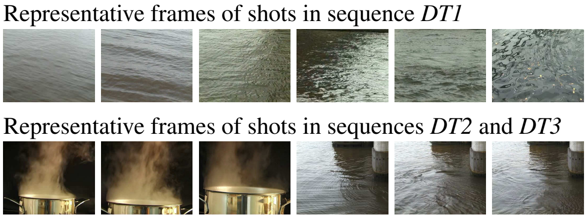


Figure 4. Temporal window with typical dissolve transition in the middle.

3 Results and Discussion

The performance of the proposed segmentation technique was tested on a range of artificial and real sequences. From each frame only a 1200 pixels were sampled giving us a resolution 40 × 30. All the test sequences were converted to a greyscale prior to the analysis and only three main principal components were used during the floating window eigen-analysis (i.e., $k = 3$), although the optimal number may be set dependently on analysed sequence. Length of temporal window T was 100 frames with 25 frames overlap.

For basic test of our method three artificial sequences were created as a sequence of similar dynamic textures (DT) from DynTex database [9]. Fig. 5 shows representative frames if individual "shots" in three dynamic textures *DT1* (different water surfaces), *DT2* (different views of the same steaming pot), and *DT3* (identical water surface in different time). Although the sequences were created from DTs exhibiting very similar temporal dynamics the method found reliably all 9 cuts.


Representative frames of shots in sequence *DT1*

Representative frames of shots in sequences *DT2* and *DT3*

Figure 5. Examples of test dynamic texture sequences.

The results on DTs were encouraging so we moved to tests on five TV advertisement clips (*MV1*, *MV2*, *MV3*, *MV4*, and *MV5*) containing 5192 frames with 94 cuts and 4 dissolves. Estimated cut transitions in these sequences are shown in Fig. 6. Each video sequence is represented by its time-slice and corresponding ground-truth (GT) set of cuts in blue. Below in red are shown cuts estimated by the proposed method together with the graph of the proposed discrimination feature f, with marked levels of used T_L and T_H. Method found all but one cuts (*MV3*), however, it missed two dissolve effects present in *MV1* and *MV2*.

Eventually, we tested our method on five sequences from TREC retrieval database [1]. The sequences contained eight NASA documentary movies of total length 45 minutes. A total of 268 cuts and 200 dissolve transitions were identified in these sequences by a human observer. Performance of our method on this dataset is summarised in Tab. 1, where N_T is a total number of transitions, N_C, N_M, N_F are numbers of correctly detected, missed, and falsely detected shots transitions, respectively. Corresponding recall $R = N_C/(N_C + N_M)$ and precision $P = N_C/(N_C + N_F)$ retrieval rates are show in last two columns.

transition	N_T	N_C	N_M	N_F	**R**	**P**
cuts	268	268	0	19	**100.0**	**93.7**
dissolves	200	157	43	81	**78.5**	**66.0**

Table 1. Performance on TREC dataset.

The speed of the method is due to the introduced floating window extension very high. Processing of 1000 frames sequence, already loaded to memory, using our C++ implementation takes approximately 3.5 seconds on PC AMD Athlon 2GHz. Thus the temporal segmentation of one hour movie with framerate 25 frames/second would take approximately 315 seconds. If needed the method's performance can be even improved if more frame pixels, principal components k or/and all RGB components are used. Nevertheless, some these changes may have impact on method's final speed.

4 Conclusions

We have presented a novel PCA-based approach to temporal segmentation of video sequences. The method provides reliable detection of cuts and promising detection of dissolve transitions in video sequences. The method is extremely fast due to subdivision of a sequence into a set of independently processed overlapping blocks, and due to very sparse greyscale sampling of movie frames. Robustness of the method was successfully tested on challenging set artificial and real movie sequences. We believe that the proposed method offers performance comparable with current state-of-the-art, while benefits from its exceptional speed.

5 Acknowledgments

The first author was supported by the EC Marie-Curie IEF No.41358. This research was also supported by grants 102/08/0593, 1ET400750407, 2C06019.

References

[1] TREC video retrieval test collection. URL:http://www.open-video.org/collection_detail.php?cid=7, 2001.

[2] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba. Video indexing using motion vectors. In *SPIE Conf. on Visual Communication and Image Processing*, volume 1818, pages 1522–1530, 1993.

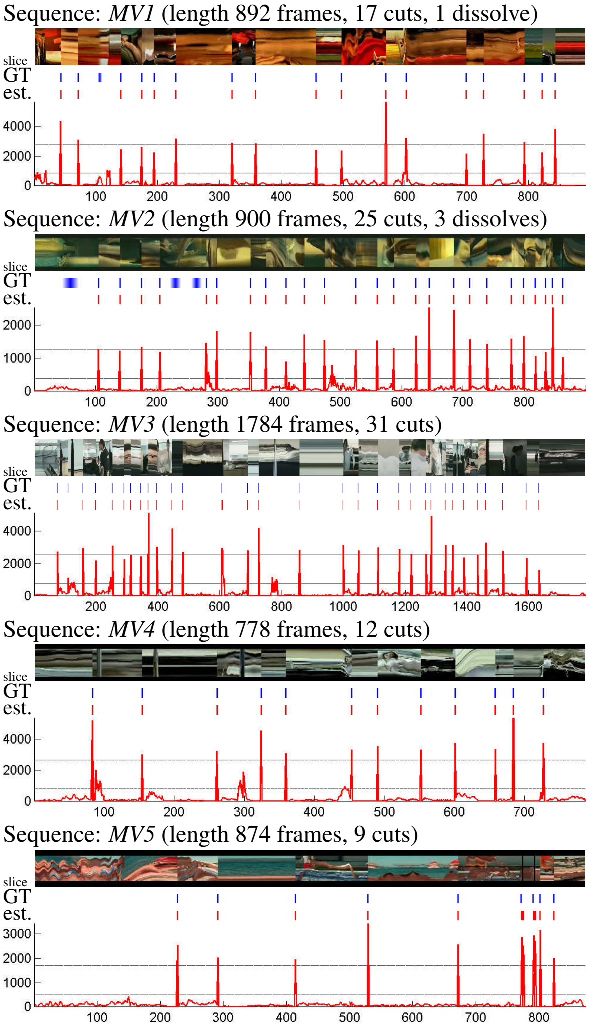
[3] G. H. Golub and C. F. Van Loan. *Matrix Computations, (3rd ed.)*. Johns Hopkins University Press, 1996.

[4] B. Günsel, A. Ferman, and A. M. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, 7(3):592–604, 1998.

[5] A. Hampapur, R. Jain, and T. Weymouth. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1(1):9–46, 1995.

[6] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing Image Communication, Elsevier Science*, 16:477–500, 2001.

[7] R. Lienhart. Reliable transition detection in videos: A survey and a practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486, 2001.


Sequence: *MV1* (length 892 frames, 17 cuts, 1 dissolve)

Sequence: *MV2* (length 900 frames, 25 cuts, 3 dissolves)

Sequence: *MV3* (length 1784 frames, 31 cuts)

Sequence: *MV4* (length 778 frames, 12 cuts)

Sequence: *MV5* (length 874 frames, 9 cuts)

Figure 6. Performance of the cuts detection on the tested movie sequences.

[8] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. *Visual Database Systems*, 2:113–127, 1992.

[9] R. Péteri and M. Huiskes. Dyntex - a comprehensive database of dynamic textures. URL:http://www.cwi.nl/projects/dyntex/.

[10] S. V. Porter, M. Mirmehdi, and B. T. Thomas. Temporal video segmentation and classification of edit effects. *Image and Vision Computing*, 13-14(21):1097–1106, December 2003.

[11] B. Shahraray. Scene change detection and content-based sampling of video sequences. In *SPIE Conf. on Digital Video Compression: Algorithms and Technologies*, pages 2–13, 1995.

[12] R. Zahib, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia systems*, 7(2):119–128, 1993.

[13] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of full-motion video. *Multimedia systems*, 1(1):10–28, 1993.