Video shot segmentation using singular value decomposition

Conference Paper *in* Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing · August 2003

DOI: 10.1109/ICME.2003.1221613 · Source: IEEE Xplore

Zuzana Cernekova
Comenius University in Bratislava
22 PUBLICATIONS **417** CITATIONS
SEE PROFILE

C. Kotropoulos
Aristotle University of Thessaloniki
259 PUBLICATIONS **4,575** CITATIONS
SEE PROFILE

VIDEO SHOT SEGMENTATION USING SINGULAR VALUE DECOMPOSITION

Z. Černeková, C. Kotropoulos, I. Pitas

Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 541 24, GREECE
E-mail: (zuzana, costas, pitas)@zeus.csd.auth.gr

ABSTRACT

A new method for detecting shot boundaries in video sequences using singular value decomposition (SVD) is proposed. The method relies on performing singular value decomposition on the matrix **A** created from 3D histograms of single frames. We have used SVD for its capabilities to derive a low dimensional refined feature space from a high dimensional raw feature space, where pattern similarity can easily be detected. The method can detect cuts and gradual transitions, such as dissolves and fades, which cannot be detected easily by entropy measures.

1. INTRODUCTION

The indexing and retrieval of digital video is an active research area. Shot boundary detection is the first preprocessing step to further analyze the video content for indexing, browsing, searching, summarization, etc. [1].

Early work on shot detection mainly focused on abrupt cuts. A comparison of existing methods is presented in [2, 3]. The standard color histogram-based algorithm and its variations are widely used for detecting cuts [4, 5]. Abrupt cut detection algorithms detect the changes between shots by comparing the differences of the intensity histograms of consecutive video frames.

Gradual transitions, such as dissolves, fade-ins, fade-outs, and wipes are examined in [6, 7, 8]. These transitions are generally more difficult to be detected, due to camera and object motions within a shot. A *fade* is a transition of gradual diminishing (fade-out) or increasing (fade-in) visual intensity. A *dissolve* can be viewed as a fade-out and fade-in with some overlap. Transitions between shots are widely used in TV and their appearance generally signals a shot change. Therefore, their detection is a very powerful tool for shot classification and story summarization. Existing techniques for shot detection rely on twin thresholding [1] or grey level statistics [2] and have a relatively high false detection rate.

In a previous work, we used entropy measures for detecting abrupt cuts and fades [9]. In this paper, we develop a method for automated shot boundary detection using singular value decomposition. The method relies on performing singular value decomposition on the matrix **A** created by the 3D color histograms of single frames. We have used the SVD for its capabilities to derive a low dimensional refined feature space from a high dimensional raw feature space, where pattern similarity can easily be detected. We have motivated by the success of SVD in document clustering and retrieval, where very good results have been reported [10]. By using SVD we are able to detect dissolves which were not addressed in our previous work.

2. SINGULAR VALUE DECOMPOSITION

The singular value decomposition (SVD) is a powerful linear algebra technique. The SVD exposes the geometric structure of a matrix, an important aspect of many matrix calculations. A matrix can be seen as a transformation from one vector space to another. The components of the SVD quantify the resulting change between the underlying geometry of those vector spaces. The SVD is employed in a variety of applications, from least-squares problems to solving systems of linear equations.

Each of these applications exploits the key properties of the SVD, i.e., its relation to the rank of a matrix and its ability to approximate matrices of a given rank. Many fundamental aspects of linear algebra rely on determining the rank of a matrix, making the SVD an important and widely-used technique.

The SVD of an $M \times N$ matrix **A** whose number of rows M is greater than or equal to its number of columns N, is any factorization of the form $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where **U** is an $M \times N$ column-orthogonal matrix, **V** is an $N \times N$ column orthogonal matrix, and $\mathbf{\Sigma} = diag(\sigma_1, ..., \sigma_R)$ is a diagonal matrix with non-negative elements, with $\sigma_1 \geq ... \geq \sigma_R \geq 0$ and $R = \min(M, N)$. The values σ_i are the **singular**

This work has been supported by the European Union Research Training Network "Methods for Unified Multimedia Information Retrieval" (MOUMIR)

values, whereas the first R columns of $\mathbf{V}$ and $\mathbf{U}$ are called the **right singular vectors** and the **left singular vectors**, respectively.

3. SHOT DETECTION

In our approach, we calculated an M-dimensional feature vector $\mathbf{a}_i$ for each frame f_i, $i = 1, 2, \ldots, N$. Using $\mathbf{a}_i$ as a column, we obtained the matrix

$$\mathbf{A} = [\mathbf{a}_1|...|\mathbf{a}_N].$$

As feature vector we chose the color histogram of each frame. More specifically, we calculated the three-dimensional normalized histograms in the RGB color space with 16 bins, for each of the R, G, B color components. Thus, the dimensionality of feature vectors is $M = 16^3 = 4096$.

Using such a feature vector as a column we created the $M \times N$ feature matrix $\mathbf{A}$. Each feature is associated with a row vector of $\mathbf{A}$ of dimensions $1 \times N$ and each frame is described by a column vector of $\mathbf{A}$ of dimensions $M \times 1$.

The column vectors of $\mathbf{A}$, that is, the frame color histograms, are projected onto the orthonormal basis formed by vectors of the left singular matrix $\mathbf{U}$. The coordinates of the frames in this space are given by the columns of $\mathbf{\Sigma V}^T$.

The row vectors of $\mathbf{A}$ (i.e., the colors) are projected onto the orthonormal basis by the column vectors of the right singular matrix $\mathbf{V}^T$ or equivalently the row vectors of $\mathbf{V}$. The representation of colors, in terms of coordinates in this projection, is given by the rows of $\mathbf{U\Sigma}$.

3.1. Clustering

Performing SVD we project vectors from the M-dimensional feature space to a K-dimensional ($K << R$) refined feature space, by preserving only the K largest singular values of $\mathbf{S}$. Let us call the resulting matrix $\mathbf{S}_K$. Let us denote by $\widetilde{\mathbf{v}}_i = \mathbf{v}_i \mathbf{\Sigma}_K$ the projected frame histograms. Then each column vector $\mathbf{a}_i$ in $\mathbf{A}$ is mapped to a row vector $\widetilde{\mathbf{v}}_i$. The truncated feature space removes the noise or the trivial variations in the video sequence. The frames with similar color distribution patterns will be mapped close to each other. From an analogy with the SVD-based document clustering and retrieval, clustering of visually similar frames in the refined feature space will certainly yield better results than in the raw feature space.

As a measure of similarity we have defined the angle between the row vectors $\widetilde{\mathbf{v}}_i$ and $\widetilde{\mathbf{v}}_j$:

$$\Phi(f_i, f_j) = \cos(\widetilde{\mathbf{v}}_i, \widetilde{\mathbf{v}}_j) = \frac{(\widetilde{\mathbf{v}}_i \cdot \widetilde{\mathbf{v}}_j^T)}{||\widetilde{\mathbf{v}}_i||||\widetilde{\mathbf{v}}_j||} \qquad (1)$$

Using the similarity measure (1) we obtain values in the range $[0, 1]$, where 1 stays for identical frames. The more different the vectors are, a closer value to 0 is obtained.

To detect shots we are using a dynamic clustering method. The frames are clustered into L clusters, $\{c_i\}_{i=1}^L$, by comparing their similarity measure (1) to a threshold δ. The clustering algorithm works as follows.

Initialization:

- It refers to the first two frames f_1 and f_2 represented by $\widetilde{\mathbf{v}}_1$ and $\widetilde{\mathbf{v}}_2$. They form the cluster c_1 by definition. The cluster mean is simply

$$\overline{\mathbf{m}}_1 = \frac{1}{2}\{\widetilde{\mathbf{v}}_1 + \widetilde{\mathbf{v}}_2\}. \qquad (2)$$

Recursion:

- Frame f_3 is tested if it should be added to c_1 or to become a seed for a new cluster. We test if

$$\Phi(\overline{\mathbf{m}}_1, \widetilde{\mathbf{v}}_3) < \delta \qquad (3)$$

If the inequality (3) is fulfilled then we create a new cluster with mean

$$\overline{\mathbf{m}}_1 = \widetilde{\mathbf{v}}_3. \qquad (4)$$

Otherwise, we include f_3 into c_1 and update $\overline{\mathbf{m}}_1$.

- When frame f_l is to be processed, we are interested in testing if f_l is to be included into the last cluster formed chronologically up to the l-th time instant. Let us denote by c_j this cluster. Then, we test if

$$\Phi(\overline{\mathbf{m}}_j, \widetilde{\mathbf{v}}_l) < \delta. \qquad (5)$$

In case (5) is satisfied we create a new cluster c_{j+1} represented by $\overline{\mathbf{m}}_{j+1} = \widetilde{\mathbf{v}}_l$. Otherwise $\overline{\mathbf{m}}_j$ is updated after inclusion of $\widetilde{\mathbf{v}}_l$.

The sparse clusters usually show the transition between the shots. Accordingly, from the obtained clusters, the dense ones are identified and associated to shots.

We summarize the algorithm like this:

1. Calculate the color histograms for each frame with 16 bins for each R, G, B component.

2. Create the matrix $\mathbf{A}$ using the histograms as columns.

3. Perform SVD on matrix $\mathbf{A}$: $\mathbf{A} = \mathbf{U\Sigma V}^T$.

4. Apply the just described dynamic clustering method.

5. Choose clusters, that are not sparse.

video	frames	cuts	fade-ins	fade-outs	dissolves
basketball	3882	44	7	4	3
news	5471	29	6	6	5
teste	3229	17	0	0	4

Table 1. *The video set used in the experiments and the respective number of frames, abrupt cuts, fade-ins, fade-outs, and dissolves.*

4. EXPERIMENTAL RESULTS

The proposed method was tested on several real TV sequences having many commercials in-between, characterized by significant camera effects like zoom-ins/outs and pans, abrupt camera movement and significant object and camera motion inside single shots. For each video sequence, a human observer has determined the precise location and duration of the edits to be used as ground truth. The corresponding data are depicted in Table 1.

Let GT denote the ground truth, Seg be the segmented (correct and false) shots using our method and $|E|$ be the number of elements (frames) of a set E. In order to evaluate the performance of the segmentation method presented in Section 3, the following measures, inspired by receiver operating characteristics in statistical detection theory [2, 11] were used:

- The *recall* measure, also called true positives function or sensitivity, corresponding to the probability of detection:

$$Recall = \frac{|Seg \bigcap GT|}{|GT|}. \tag{6}$$

- The *precision* corresponding to the accuracy of the method considering the false detection:

$$Precision = \frac{|Seg \bigcap GT|}{|Seg|}. \tag{7}$$

We have tested the method with several choices of K. Based on experiments the best choice was shown to be $K = 10$. For this value the clusters are separated well. By increasing K the data become more messy. The static shots with small camera and object movements inside the shot are projected to data with small dispersion, while shots with some action inside the shot are projected to data with a large dispersion. Table 2 summarizes the recall and precision rates measured for cuts, gradual transitions, as well as for both of them using $\delta = 0.9$ and $K = 10$.

By varying $\delta \in [0.75, 0.98]$ we have obtained the recall-precision curve shown in Figure 1.

The transition between two shots is shown as a path between two dense clusters of points in the projected space in Figure 2. Therefore, we can easily detect the transition.

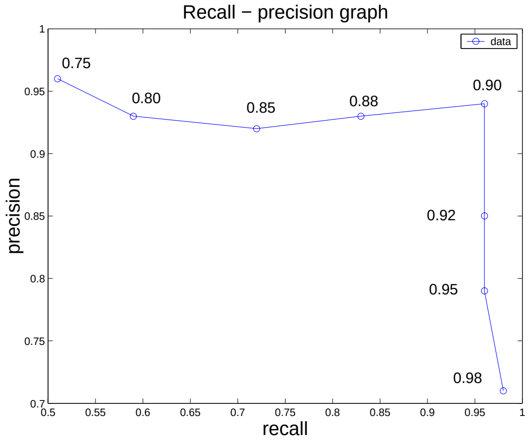

Fig. 1. Recall-precision curve by varying the threshold δ.

We can also distinguish between transitions and shots with high camera and object movements inside them. Typical shots containing small and big motion are demonstrated in Figure 3. If we used any method based on histogram comparisons, which are the most common, we would hardly identify movements inside a shot and transitions between two shots. Most of the time, the motion inside a shot gives rise to false alarms. Using SVD, we can distinguish between the motion and transition and avoid the false alarms. The method is able to detect dissolves, which we did not address in the previous work with the entropy measures [9].

5. CONCLUSIONS AND DISCUSSION

A new technique for automated shot transitions detection using singular value decomposition was presented. The method is able to detect well the dissolves which we did not address in the previous work with the entropy measures. The reported results are promising. However, using a fixed threshold in the dynamic clustering method yields some deficiencies. False detections occur for shots with big motion, because the clusters are more spread and a fixed threshold cannot preserve all frames in the same cluster. We intend to improve the clustering method by employing an adaptive threshold which would vary based on the density of the frames in the space and by introducing criteria for a possible merging of clusters.

video	cuts		gradual transition		global	
	Recall	Precision	Recall	Precision	Recall	Precision
basketball	0.95	0.98	0.85	1.00	0.93	0.98
news	0.93	0.90	1.00	1.00	0.96	0.94
teste	1.00	1.00	1.00	1.00	1.00	1.00

Table 2. *Shot detection results using a fixed threshold.*

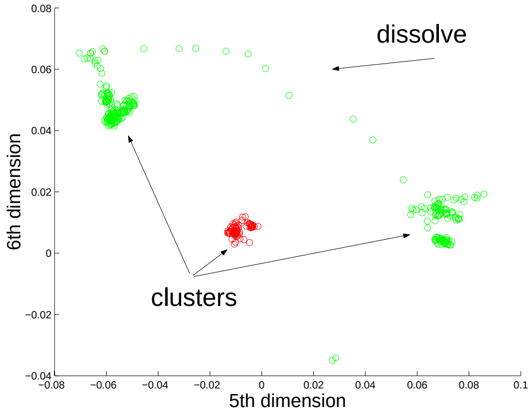

Fig. 2. By using the fifth and sixth dimension of the projected frame histograms we can successfully detect the dissolve in sequence "teste" .

6. REFERENCES

[1] A. Del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, Inc, San Francisco, California, 1999.

[2] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Proc. of SPIE Storage and Retrieval for Image and Video Databases VII, San Jose, CA, U.S.A.*, January 1999, vol. 3656, pp. 290–301.

[3] A. Dailianas, R. B. Allen, and P. England, "Comparison of automatic video segmentation algorithms," in *Proceedings, SPIE Photonics East'95: Integration Issues in Large Commercial Media Delivery Systems, Oct. 1995, Philadelphia*, 1995, vol. 2615, pp. 2–16.

[4] G. Ahanger and T.D.C. Little, "A survey of technologies for parsing and indexing digital video," *Journal of Visual Communication and Image Representation*, vol. 7, no. 1, pp. 28–43, 1996.

[5] S. Tsekeridou and I. Pitas, "Content-based video parsing and indexing based on audio-visual interaction," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 522–535, 2001.

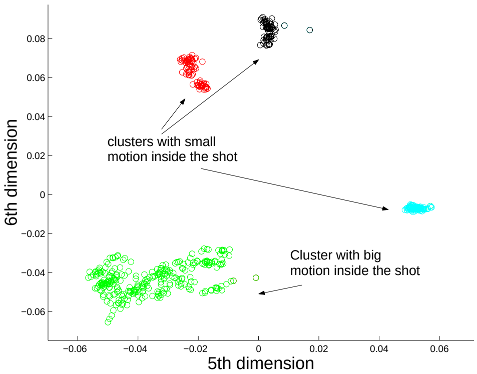

Fig. 3. Projected frame histograms into the fifth and sixth dimension show difference between shots with small and big motion inside the shot.

[6] R. Lienhart, "Reliable dissolve detection," in *Proc. of SPIE Storage and Retrieval for Media Databases 2001*, January 2001, vol. 4315, pp. 219–230.

[7] M. S. Drew, Z.-N. Li, and X. Zhong, "Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences," in *Proc. 2000 IEEE Int. Conf. on Image Processing*, 2000, vol. 3, pp. 929–932.

[8] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, November 2000.

[9] Z. Cernekova, C. Nikou, and I. Pitas, "Shot detection in video sequences using entropy-based metrics," in *Proc. 2002 IEEE Int. Conf. Image Processing, Rochester,N.Y.,USA,22-25 September,*, 2002.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41(6), pp. 391–407, 1990.

[11] C. E. Metz, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, vol. 8, pp. 283–298, 1978.