

# Video shot boundary detection using singular value decomposition and statistical tests

Zuzana Černeková

Constantine Kotropoulos

Ioannis Pitas

Aristotle University of Thessaloniki

Department of Informatics

Artificial Intelligence and Information Analysis Laboratory,

Box 451

54124 Thessaloniki,

Greece

## **Address for correspondence :**

Professor Ioannis Pitas

Aristotle University of Thessaloniki

Department of Informatics

Artificial Intelligence and Information Analysis Laboratory,

54124 Thessaloniki

GREECE

Tel. ++ 30 2310 99 63 04

Fax ++ 30 2310 99 63 04

email: [pitas@aiia.csd.auth.gr](mailto:pitas@aiia.csd.auth.gr)

## Abstract

In this paper, we deal with video shot-cut detection in digital videos using singular value decomposition (SVD). SVD is performed on a matrix, whose columns are the 3D frame color histograms. We have used SVD for its capabilities to derive a refined low dimensional feature space from the high dimensional raw feature space, where similar video patterns are placed together and can be easily clustered. After performing SVD, a two-phase process is employed to detect the shots. In the first phase, a dynamic clustering method is used to create the frame clusters. In the second phase, every two consecutive clusters, obtained by the clustering procedure, are tested for a possible merging in order to reduce false shot cut detections. In the merging phase, statistical hypothesis testing is used. The detection technique was applied to several TRECVID video test sets that exhibit different types of shots and contain significant object and camera motion inside the shots. It is demonstrated that the method detects cuts and gradual transitions, such as dissolves and fades, with a high accuracy.

**Keywords:** shot boundary detection, singular value decomposition, Mises-Fisher distribution

# 1 Introduction

The indexing and retrieval of digital video is an active research area. Video segmentation is a fundamental step in analyzing the content of video sequences and accessing, retrieving, and browsing large video databases efficiently [1].

The video shot is a basic structural building block of video sequences. Its boundaries need to be determined possibly automatically to allow for content-based video manipulation. A video shot can be defined as a sequence of frames captured by *one camera in a single continuous action in time and space* [2]. It should be a group of frames that have consistent visual (including color, texture, and motion) characteristics.

A video *shot cut* (abrupt cut) is an instantaneous content transition from one shot to the next one. It is obtained by simply joining two different shots without the insertion of any other photographic effect. The shot cut boundaries show an abrupt change in image intensity or color. Cuts between shots with little content difference or small camera motion and constant illumination conditions can be easily detected by looking for sharp changes in brightness. However, in the presence of either continuous fast object motion or camera movements or illumination changes, it is difficult to distinguish if the changes in brightness are due to the aforementioned reasons or the transition from one shot to the next one [1].

Fading is either the progressive darkening of a shot until the last frame becomes black (fade-out) or the gradual transition from a black frame to a fully illuminated one (fade-in). An example of fade-out is given in Figure 1a, while a fade-in can be seen in Figure 1b. Fades spread the boundary between two shots across a number of consecutive video frames. They possess start and end frames that identify the transition sequence. A *dissolve* is a gradual transition from the content of one shot to the content of the next shot. An example of

dissolve is shown in Figure 1c. Transitions between shots are widely used in TV and their appearance generally signals a shot change. Video transitions, such as fades and dissolves can be mathematically modeled as luminance scaling operations. Let  $\mathbf{x}$  and  $t$  denote pixel coordinates and time, respectively. If  $G(\mathbf{x}, t)$  is a grey scale video sequence and  $T$  is the length of the transition sequence, a grey level scaling of  $G(\mathbf{x}, t)$  is modeled as [1]:

$$E_1(\mathbf{x}, t) = G(\mathbf{x}, t) \cdot \left(1 - \frac{t}{T}\right) \quad (1)$$

for  $t \in [0, T]$ . It is not difficult to realize that (1) models a fade out. Similarly, a fade-in is modeled by:

$$E_2(\mathbf{x}, t) = G(\mathbf{x}, t) \cdot \left(\frac{t}{T}\right). \quad (2)$$

A dissolve sequence  $E_3(\mathbf{x}, t)$  is defined as a mixture of fade-out  $E_1(\mathbf{x}, t)$  and fade-in  $E_2(\mathbf{x}, t)$  with weights  $w_1$  and  $w_2$  [3]:

$$E_3(\mathbf{x}, t) = w_1 \cdot E_1(\mathbf{x}, t) + w_2 \cdot E_2(\mathbf{x}, t) \quad t \in [0, T]. \quad (3)$$

For  $t \in [0, T]$ , the most common dissolve types are *cross-dissolves* with:

$$w_1 = \frac{T-t}{T}, \quad w_2 = \frac{t}{T}, \quad (4)$$

and *additive dissolves* with

$$w_1 = \begin{cases} 1 & \text{if } t < c_1 \\ \frac{T-t}{T-c_1} & \text{otherwise} \end{cases} \quad w_2 = \begin{cases} \frac{t}{c_2} & \text{if } t < c_2 \\ 1 & \text{otherwise,} \end{cases} \quad (5)$$

where  $c_1$  and  $c_2$  are constants in the range  $[0, T]$ . Although Eqs (1)-(5) do not take into account the fact that optical cross dissolves are usually not linear, they can be treated as simplified linear approximations of the actual models. They can be extended to color images by applying them separately to each RGB color component.

The detection of shot boundaries provides a basis for nearly all video abstraction and high-level video summarization approaches. Therefore, solving the problem of shot-boundary detection is one of the major prerequisites for revealing the high-level video content structure. Moreover, other research areas can profit considerably from a successful automation of the shot-boundary detection process. Most of the first works on shot boundary detection were mainly focused on abrupt cut detection. In these approaches, a cut is detected when a certain difference measure between consecutive frames exceeds a threshold. The difference measure is computed either at a pixel level or at a block level. Noticing the weakness of pixel differencing methods, many researchers suggested the use of measures based on global information, such as intensity histograms or color histograms. The standard color histogram-based algorithm and its variations are widely used for detecting cuts [4–7]. Abrupt cut detection algorithms determine the changes between shots by comparing the differences between intensity histograms in consecutive video frames. A comparison of such abrupt cut detection algorithms is presented in [4, 8, 9]. However, the histograms are incapable to differentiate between smooth camera operations and gradual shot transitions, because they do not explicitly model the image difference caused by camera movements. In [10–12], entropy measures were used for detecting abrupt cuts and fades in gray-scale or color images. More complex features, such as image edges or motion vectors [13], improve the situation and alleviate, but do not solve completely, this problem.

Gradual transitions, such as dissolves, fade-ins, fade-outs, and wipes are examined in [3, 14–17]. These transitions are generally more difficult to be detected, due to camera and object motions within a shot. Therefore, their detection is a very powerful tool for shot segmentation and story summarization. Existing techniques in the literature for fade detection rely on twin thresholding [1] or the standard deviation of pixel intensities [8]. Lienhart [8]

proposed first to locate all monochromatic frames in the video as potential start/end points of fades. Monochromatic frames are defined as frames with standard deviation of pixel color values close to zero. Fades were then detected by starting to search in both directions for a linear increase in the standard deviation of pixel intensity/color. An average hit rate of 0.87 was reported at a false alarm rate of 0.30. An alternative approach, also based on the variance of pixel intensities, was proposed by Alattar [18]. Fades were detected first by recording all negative spikes in the time series of the second order derivative of the pixel intensity variance, and then by ensuring that the first order derivative of the mean of the video sequence was relatively constant next to the negative spike. A combination of both approaches is described in Truong et al. [19]. A conceptual solution to the shot-boundary detection problem presents Hanjalic in [20]. This solution is provided in the form of a statistical detector that is based on the minimization of the average detection-error probability.

Porter et al. [21] developed an algorithm for dissolve detection, which combines the advantages of object tracking and feature-based methods. It avoids the sensitivity of object detection but provides a measure of the temporal evolution of the video. Experimental results on commercial motion picture trailers gave 0.87 recall rate at 0.77 precision rate. Lienhart [3] proposed to detect dissolves by a learning classifier (e.g. a neural network). The classifier detects possible dissolves at multiple temporal scales and merges the results using a winner-take all strategy. The interesting part is that the classifier is trained using a dissolve synthesizer which creates artificial dissolves from any available set of video sequences.

The methods for dimensionality reduction such as principal components analysis (PCA) and latent semantic indexing (LSI), which are both using the singular value decomposition (SVD) have been used in several works. This paper extends previously reported results in [22]. Besides using SVD for color histogram dimensionality reduction it builds on the

distributional properties of angular statistics to refine ad-hoc clustering methods applied to cosine similarity measures by employing hypothesis testing. The closely related to SVD, PCA, was first proposed for scene change detection in order to find the transition between video scenes by Sahouria and Zakhor in [23]. They applied PCA to the covariance matrix of feature vectors that represent the motion in a frame and demonstrated its success in two applications. The first application accomplishes a high-level scene description without shot detection and key frame selection, while the second one uses the time sequences of motion data from every frame to classify sports sequences. Similarly, PCA was applied to the covariance matrix of color triplets in RGB color space for shot change detection by Yilmaz and Shah in [24]. To detect the shots from a video stream, they use a “cluster seeking” approach. Liu and Chen in [25] propose to model temporal statistics of the video stream using PCA or the eigenspace method. A shot boundary is detected if the new feature, the histogram of a frame, does not fit well to an existing model. Gong and Liu [26] proposed a technique for video shot segmentation and visual content-based shot classification based on the SVD followed by a clustering method. SVD was applied to the quantized RGB histograms of  $3 \times 3$  blocks to which a frame was divided. For clustering, they used a twin thresholding technique and the weighted Euclidean distance calculated between two consecutive frames. Although, [26] is somehow close to our work (i.e., both use SVD), it is different to our method. We do not divide the frames into blocks. We work on angular vectors and use suitable distribution, e.g. the von Mises-Fisher one [27, 28], furthermore, the clustering algorithm employed here is completely different. LSI was also tested for video shot detection in [29] using frame quantized histograms of hue and saturation. In [29], it was showed how LSI, together with color anglogram, can expose the semantic correlation between video frames. LSI was also applied for video content modeling and analysis in [30].

As can be seen from the cited references the predecessor of this work, [22], was appeared parallel to [29] and [30].

As successive frames in the same shot change only slightly, it can be reasonably expected that they will cluster well. Clustering can be performed using a range of video features including region shape, color, and texture [31]. Ren et al. [32] showed how gray-scale global texture based frame clustering can be used to obtain a good estimate of the number of shots present and thus detecting appropriate shot changes. They used autocorrelation texture features for describing each frame. The features are next clustered using fuzzy c-means. A temporal validity index is introduced to calculate whether frames that cluster together have some temporal relationship. Color texture features were used in [33].

In order to obtain good results for shot cut detection, it is important to choose and combine properly the method for reducing dimensionality with a suitable clustering method. In the present paper, we develop a novel method that builds on SVD and clustering of features vectors extracted from consecutive (in time) frames. More specifically, the method relies on performing SVD on a matrix created by the 3D color histograms of single frames. In contrary to the previous methods which are using as input to SVD the covariance matrix of feature vectors, we calculate the SVD of an affinity matrix. We have used SVD for its capabilities to derive a refined low dimensional feature space from a high dimensional raw feature space, where pattern similarity can easily be detected. We have been motivated by the success of SVD in document clustering and retrieval, where very good results have been reported [34, 35]. In order to detect video shots, the feature vectors after applying SVD are processed using a dynamic clustering method. As a measure of similarity we use the cosine measure. To avoid false detections, every two consecutive clusters, obtained by the just mentioned clustering procedure, are tested for a possible merging in a second phase.



Merging is performed in two steps applied consecutively. The first step uses the average cosine similarity measure of the clusters. In order to derive theoretically grounded thresholds, we perform the second merging step, which is based on statistical hypothesis testing and contributes to the novelty of our approach. It uses the von Mises-Fisher distribution [27, 28], which can be considered as the equivalent of the Gaussian distribution for directional data. By using SVD and statistical tests, we are able to detect transitions between video shots with a high accuracy.

The outline of the paper is as follows: In Section 2, a brief description of SVD is presented. The application of SVD to shot boundary detection is addressed in Section 3. After performing SVD on the histograms we can apply the method described in Section 4 for fade detection. Experimental results are presented and discussed in Section 5 and conclusions are drawn in Section 6.

## 2 Singular value decomposition

SVD is a powerful technique in linear algebra. The SVD exposes the geometric structure of a matrix, which is an important aspect in many matrix calculations. A matrix can be seen as defining a transformation from one vector space to another one. The components of SVD quantify the resulting change between the underlying geometry of these vector spaces. 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 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 SVD an important and widely-used technique.

The SVD of an  $M \times N$  matrix  $\mathbf{A}$  is any factorization of the form  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , where  $\mathbf{U}$  is an  $M \times R$  column-orthogonal matrix,  $\mathbf{V}$  is an  $N \times R$  column orthogonal matrix, and  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_R)$  is a diagonal matrix with non-negative elements,  $\sigma_1 \geq \dots \geq \sigma_R \geq 0$  for  $R = \min(M, N)$ . The diagonal elements  $\sigma_i$  are called *singular values*. They are the square roots of the largest  $R$  eigenvalues of  $\mathbf{A}\mathbf{A}^T$  or  $\mathbf{A}^T\mathbf{A}$ . The  $R$  columns of  $\mathbf{V}$  and  $\mathbf{U}$  are called the *right singular vectors* and the *left singular vectors*, respectively. The right singular vectors are the eigenvectors of  $\mathbf{A}^T\mathbf{A}$ , whereas the left singular vectors are the eigenvectors of  $\mathbf{A}\mathbf{A}^T$ . Only  $r = \text{rank}(\mathbf{A})$  singular values are non-zero. Accordingly

$$\mathbf{A}_{M \times N} = \left[ \underbrace{\mathbf{U}_1}_{r} \underbrace{\mathbf{U}_2}_{R-r} \right] \begin{bmatrix} \mathbf{\Sigma}_1 & 0 \\ 0 & 0 \end{bmatrix} \left[ \begin{array}{c} \overbrace{\left[ \mathbf{V}_1^T \right]}^N \\ \underbrace{\left[ \mathbf{V}_2^T \right]}_{R-r} \end{array} \right] \} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T \quad (6)$$

where  $\mathbf{\Sigma}_1 = \text{diag}[\sigma_1, \dots, \sigma_r]$  is nonsingular.

### 3 Shot cut detection using SVD and clustering

Let  $\mathbf{a}_i$  denote an  $M$ -dimensional feature vector that is used to represent the  $i$ -th video frame,  $i = 1, 2, \dots, N$ . The feature vector is obtained by calculating the color histogram of each frame as follows. The three-dimensional normalized histogram in the RGB color space with 16 bins for each of the  $R, G, B$  color components is derived. Accordingly, the dimensionality of feature vectors is  $M = 16^3 = 4096$ . Next we create the  $M \times N$  affinity matrix

$$\mathbf{A} = [\mathbf{a}_1 | \mathbf{a}_2 | \dots | \mathbf{a}_N]$$

whose columns are the  $M$ -dimensional feature vectors of all the frames. The feature vectors are normalized, so that  $\|\mathbf{a}_i\| = 1$ ,  $i = 1, 2, \dots, N$ . Each feature (i.e. color) is associated with a row vector of  $\mathbf{A}$  having dimensions  $1 \times N$ . Each frame is described by a column vector of

$\mathbf{A}$  having dimensions  $M \times 1$ . Typically,  $\mathbf{A}$  is a sparse matrix, a fact that is exploited by the numerical analysis algorithms for the computation of SVD.

The column vectors of  $\mathbf{A}$ , that is, the frame color histograms, are projected onto the orthonormal basis formed by the column vectors of the left singular matrix  $\mathbf{U}$ . The frame coordinates 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 formed by the column vectors of the right singular matrix  $\mathbf{V}$  or, equivalently, the row vectors of  $\mathbf{V}^T$ . The color representation, in terms of their coordinates in this projection, is given by the rows of  $\mathbf{U\Sigma}$ .

### 3.1 Reduced space representation

In our application  $M < N$ . Typically,  $M = 4096$  and  $N = 10000$ . If we preserve the  $K$  largest singular values of  $\Sigma$ , where  $K < r \ll M$ , the resulting matrix is denoted by  $\Sigma_K$ . If we project the original feature vector  $\mathbf{a}_i$  from the  $M$ -dimensional feature space to the  $K$ -dimensional feature space, then the projected frame histogram is the row vector of dimensions  $(1 \times K)$ :

$$\tilde{\mathbf{v}}_i^T = \mathbf{v}_i^T \mathbf{\Sigma}_K, \quad i = 1, 2, \dots, N \quad (7)$$

where  $\mathbf{v}_i^T$  is the  $i$ -th row vector of  $\mathbf{V}_1$  in (6). Therefore, each column vector  $\mathbf{a}_i$  in  $\mathbf{A}$  is mapped to a row vector  $\tilde{\mathbf{v}}_i^T$ .

The truncated feature space removes the noise or the trivial variations in the video sequence. Thus, the frames with similar color distribution patterns are expected to be mapped close to each other. In analogy with the SVD-based document clustering and retrieval [34,35], the clustering of visually similar frames in the reduced feature space will certainly yield better results than in the raw feature space.

A commonly used frame similarity measure is the cosine measure  $\Phi(f_i, f_j)$  between two frames  $f_i$  and  $f_j$  [30, 36], that is the cosine of the angle between the vectors  $\tilde{\mathbf{v}}_i$  and  $\tilde{\mathbf{v}}_j$

$$\Phi(f_i, f_j) = \cos(\tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j) = \frac{(\tilde{\mathbf{v}}_i^T \cdot \tilde{\mathbf{v}}_j)}{\|\tilde{\mathbf{v}}_i\| \|\tilde{\mathbf{v}}_j\|}. \quad (8)$$

The cosine measure (8) is simply the inner product between the normalized projected frame histograms

$$\tilde{\tilde{\mathbf{v}}}_i = \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|}, \quad \tilde{\tilde{\mathbf{v}}}_j = \frac{\tilde{\mathbf{v}}_j}{\|\tilde{\mathbf{v}}_j\|}. \quad (9)$$

Using the similarity measure (8) we obtain values in the range  $[0, 1]$ , where 1 stands for the identical frames. The more different the vectors are, the smaller cosine measure value is obtained. The cosine measure and the Euclidean distance between  $\tilde{\mathbf{v}}_i$  and  $\tilde{\mathbf{v}}_j$  give the same results, if and only if  $\tilde{\mathbf{v}}_i$  and  $\tilde{\mathbf{v}}_j$  have unit norm, which is not the case here. The Euclidean distance and the angular measures represent two distinct approaches to judge similarity. Euclidean distance measures are intrinsic, based solely on the group of frames under study [37]. In this case, all directions are considered equal from a given point in the frame feature space and frame similarity depends only on the Euclidean distance. On the contrary, angular measures are extrinsic, representing a view of the frame feature space from its origin. An angular measure does not consider the distance of each frame feature vector from the origin, but only its direction. The cosine measure projects the entire frame space onto a  $K$ -dimensional sphere of fixed radius around the origin.

To detect video shot cuts we are using a two-phase process. In the first phase, a dynamic clustering method is used to create frame clusters (Section 3.2). In the second phase, every two consecutive frame clusters are tested for a possible merging by applying sequentially the techniques described in Sections 3.3 and 3.4.

### 3.2 Frame Clustering

In the first phase, feature vectors are assigned into clusters. We deal with clusters because the feature vectors that correspond to frames within the same shot are grouped together. Within each cluster, we have a typical density of feature vectors which is considerably higher than that within transition areas. Clustering method based on the notion of cluster density, which is designed to discover clusters of arbitrary shape, was proposed in [38]. This algorithm is not applicable here due to another characteristic of our data, the time-ordering of frames and consequently the extracted feature vectors. Since we wish to cluster the sequence into shots with a natural time ordering, for every frame (feature vector) we should decide if it belongs to the last cluster created or it is a seed for a new cluster. By doing so, we avoid having to specify a priori the number of clusters, as for example, we have to, in the widely used  $k$ -means clustering.

We propose a novel clustering method, which takes into account both aspects namely, the density and the time ordering of the feature vectors. The feature vectors are clustered into  $L$  frame clusters,  $\{\mathcal{C}_i\}_{i=1}^L$ , by comparing the similarity measure (8) to an appropriately chosen threshold  $\delta$ . The clustering algorithm works as follows.

Initialization:

- It refers to the first two frames  $f_1$  and  $f_2$  represented by their feature vectors  $\tilde{\mathbf{v}}_1$  and  $\tilde{\mathbf{v}}_2$ . They form the cluster  $\mathcal{C}_1$  by definition. The cluster mean is simply

$$\overline{\mathbf{m}}_1 = \frac{1}{2}\{\tilde{\mathbf{v}}_1 + \tilde{\mathbf{v}}_2\}. \quad (10)$$

Recursion:

- The next frame  $f_3$  is tested whether it can be appended to  $\mathcal{C}_1$  or it becomes a seed for

a new cluster by comparing the cosine measure between  $\overline{\mathbf{m}}_1$  and  $\tilde{\mathbf{v}}_3$  to  $\delta$

$$\cos(\overline{\mathbf{m}}_1, \tilde{\mathbf{v}}_3) < \delta. \quad (11)$$

If the inequality (11) is fulfilled, then we create a new cluster with mean

$$\overline{\mathbf{m}}_2 = \tilde{\mathbf{v}}_3. \quad (12)$$

Otherwise, we include  $f_3$  into  $\mathcal{C}_1$  and update  $\overline{\mathbf{m}}_1$  as follows:

$$\overline{\mathbf{m}}'_1 = \overline{\mathbf{m}}_1 + \frac{1}{n_1 + 1} \mathbf{d} \quad (13)$$

where  $\mathbf{d} = \tilde{\mathbf{v}}_3 - \overline{\mathbf{m}}_1$  and  $n_1$  is the number of elements in the cluster  $\mathcal{C}_1$ .

- When the 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 time instant  $l$ . Let us denote this cluster by  $\mathcal{C}_j$ . Then  $\cos(\overline{\mathbf{m}}_j, \tilde{\mathbf{v}}_l)$  is compared to  $\delta$  similarly to (11). If it is less than  $\delta$ , we create a new cluster,  $\mathcal{C}_{j+1}$ , that is represented by  $\overline{\mathbf{m}}_{j+1} = \tilde{\mathbf{v}}_l$ . Otherwise,  $\overline{\mathbf{m}}_j$  is updated after the inclusion of  $\tilde{\mathbf{v}}_l$ , as in (13).

The low cardinality clusters (i.e., those having few frames) usually correspond to shot transitions like dissolve, fade or wipe. Accordingly, only the clusters having a large cardinality (i.e., those having been assigned many frames) are retained and associated to shots.

We summarize the algorithm as follows [22]:

1. Calculate the color histograms for each frame with 16 bins for each  $R, G$  and  $B$  component.
2. Create matrix using the histograms as columns.
3. Perform SVD on  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ .

4. Apply the previously described dynamic frame feature clustering method.
5. Choose clusters that contain at least 5 frames as candidates of video shots. A discussion related to the empirically derived sufficient number of 5 frames is made in Section 5.

### 3.3 Frame cluster merging

Due to the fixed threshold  $\delta$  used in frame clustering, it may happen that some shots are split into different clusters. To avoid false shot transition detection, the clusters obtained by the procedure described in Section 3.2 are tested for a possible shot merging. Merging is performed in two steps applied sequentially. Since we take into account the time order of frames in every step, we are testing only two consecutive clusters for merging.

The first step is based on the fact that, if a frame cluster was erroneously split in two clusters (e.g.  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$ ), the cosine similarity measure between the last frame in cluster  $\mathcal{C}_k$  and the first frame in cluster  $\mathcal{C}_{k+1}$  is comparable to the average cosine similarity measures calculated within the clusters  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$ . Let us denote by  $f_j^i$  the  $j$ -th frame of the  $i$ -th cluster and by  $\tilde{\mathbf{v}}_j^i$  the projected frame feature vector  $f_j^i$ . We calculate the average cosine measure  $\bar{\phi}_k$  over the series of cosine measures between any two consecutive frames  $f_i^k, f_{i+1}^k$  assigned to cluster  $\mathcal{C}_k$

$$\bar{\phi}_k = \frac{1}{n_k - 1} \sum_{i=1}^{n_k-1} \cos(\tilde{\mathbf{v}}_i^k, \tilde{\mathbf{v}}_{i+1}^k) \quad (14)$$

where  $n_k$  is the number of frames assigned to  $\mathcal{C}_k$ . Then we compare the mean cosine measures  $\bar{\phi}_k$  and  $\bar{\phi}_{k+1}$  of two consecutive clusters  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$  to the cosine measure between the last frame  $f_{n_k}^k$  in cluster  $\mathcal{C}_k$  and the first frame  $f_1^{k+1}$  in cluster  $\mathcal{C}_{k+1}$  using the inequalities

$$\cos(\tilde{\mathbf{v}}_{n_k}^k, \tilde{\mathbf{v}}_1^{k+1}) < \beta \cdot \bar{\phi}_k \quad \text{AND} \quad \cos(\tilde{\mathbf{v}}_{n_k}^k, \tilde{\mathbf{v}}_1^{k+1}) < \beta \cdot \bar{\phi}_{k+1} \quad (15)$$

where  $\beta$  is a constant, which is chosen experimentally. Obviously, the constant  $\beta$  admits

values less than 1. If (15) is fulfilled, the clusters  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$  are preserved as separated clusters and the frame cluster  $\mathcal{C}_{k+1}$  is tested next for a possible merging with  $\mathcal{C}_{k+2}$ . Otherwise,  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$  are merged together.

### 3.4 Frame cluster merging based on statistical hypothesis

In this section, we describe the second step of frame cluster merging. In order to determine the necessary thresholds theoretically, we use statistical hypothesis testing. An all-purpose probability model for unit-norm random angular vectors that are distributed unimodally with rotational symmetry is the Von Mises-Fisher distribution [27, 28] (to be defined subsequently). It is the equivalent to the classical normal distribution used in “linear” data. A clustering on the unit hypersphere using Von Mises-Fisher distribution is also described in [39].

Let us denote a random direction in  $K$  dimensions by the unit norm vector  $\mathbf{l}$ .  $\mathbf{l}$  can be expressed in terms of spherical polar coordinates by applying a proper transformation [27, 28]. Let us consider random samples on a  $K$ -dimensional sphere of unit radius around the origin that are simply the normalized projected frame histograms that belong to cluster  $\mathcal{C}_k$

$$\mathbf{l}_1^k = \tilde{\tilde{\mathbf{v}}}_1^k, \mathbf{l}_2^k = \tilde{\tilde{\mathbf{v}}}_2^k, \dots, \mathbf{l}_{n_k}^k = \tilde{\tilde{\mathbf{v}}}_{n_k}^k. \quad (16)$$

The *sample mean vector* is defined by [27, 28]

$$\bar{\mathbf{l}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{l}_i^k \quad (17)$$

and its *direction* is given by

$$\bar{\bar{\mathbf{l}}}_k = \frac{\bar{\mathbf{l}}_k}{\|\bar{\mathbf{l}}_k\|}. \quad (18)$$

$\bar{\bar{\mathbf{l}}}_k$  can be regarded as the mean direction of the samples. Let  $\bar{R}_k = \|\bar{\mathbf{l}}_k\| = \sqrt{\bar{\mathbf{l}}_k^T \bar{\mathbf{l}}_k}$ . The parameter  $\bar{R}_k$  is closely related to the notion of the *spherical variance*. A value of  $\bar{R}_k$  close



to 0 implies that the points  $\mathbf{l}_1^k, \mathbf{l}_2^k, \dots, \mathbf{l}_{n_k}^k$  are uniformly distributed, whereas a value of  $\bar{R}_k$  close to 1 implies that the points are heavily concentrated near  $\bar{\bar{\mathbf{l}}}_k$ . Another two terms of interest in our analysis are: (i) the quantity

$$R_k = n_k \bar{R}_k \quad (19)$$

called the *resultant length* and (ii) the vector

$$\mathbf{r}_k = n_k \bar{\mathbf{l}}_k = \sum_{i=1}^{n_k} \mathbf{l}_i^k \quad (20)$$

known as the *resultant vector* [27, 28].

Let  $\mathbf{l}$  be a random unit norm vector that obeys a  $K$ -variate Von Mises-Fisher distribution having rotational symmetry about the unit norm direction  $\boldsymbol{\mu}$  with concentration parameter  $\kappa$ . Then, the probability that the random unit-norm vector  $\mathbf{l}$  is contained in the differential surface element on the unit radius hypersphere in  $K$  dimensions,  $dS_K$ , is given by

$$h(\mathbf{l})dS_K = c_K(\kappa) \exp\{\kappa \boldsymbol{\mu}^T \mathbf{l}\} \quad \kappa \geq 0, \quad \boldsymbol{\mu}^T \boldsymbol{\mu} = 1 \quad (21)$$

where  $c_K(\kappa)$  is a normalizing constant given by

$$c_K(\kappa) = \frac{\kappa^{(K-1)/2}}{(2\pi)^{K/2} I_{\frac{K-1}{2}}(\kappa)} \quad (22)$$

with  $I_r(\kappa)$  denoting the modified Bessel function of the first kind and order  $r$ . Let us assume that the projected frame histograms (16) assigned to cluster  $\mathcal{C}_k$  follow a  $K$ -variate von Mises-Fisher distribution with mean direction  $\boldsymbol{\mu}$  and concentration parameter  $\kappa$ , (21).

We propose merging two consecutive clusters of shot feature vectors  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$  by comparing their sample mean directions  $\bar{\bar{\mathbf{l}}}_k$  and  $\bar{\bar{\mathbf{l}}}_{k+1}$ , with the mean direction of the tentative cluster formed after a hypothesized merging cluster  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$ ,  $\boldsymbol{\mu}_0$ , and by deciding that the merging is valid if neither of  $\bar{\bar{\mathbf{l}}}_k$  and  $\bar{\bar{\mathbf{l}}}_{k+1}$  is significantly different from  $\boldsymbol{\mu}_0$ .

Let  $\mathbf{r}_k$  be the resultant vector for the normalized projected frame histograms in  $\mathcal{C}_k$  and  $\mathbf{r}_{k+1}$  be the resultant vector for the normalized projected frame histograms in  $\mathcal{C}_{k+1}$ . Let  $\boldsymbol{\mu}_0$  be the mean direction after a tentative merging of the aforementioned clusters

$$\boldsymbol{\mu}_0 = \frac{\bar{\boldsymbol{\mu}}_0}{\|\bar{\boldsymbol{\mu}}_0\|} \quad (23)$$

where

$$\bar{\boldsymbol{\mu}}_0 = \frac{\mathbf{r}_k + \mathbf{r}_{k+1}}{n_k + n_{k+1}} \quad (24)$$

We consider the following hypothesis testing problem for cluster  $\mathcal{C}_k$ :

$$\begin{aligned} H_0 : \boldsymbol{\mu} &= \boldsymbol{\mu}_0 \\ H_1 : \boldsymbol{\mu} &\neq \boldsymbol{\mu}_0. \end{aligned} \quad (25)$$

Let  $\theta_k$  be the angle between  $\mathbf{r}_k$  and  $\boldsymbol{\mu}_0$ , then

$$\mathbf{r}_k^T \boldsymbol{\mu}_0 = R_k \cos \theta_k \quad (26)$$

where  $R_k$  is the resultant length of the  $k$ -th resultant vector  $\mathbf{r}_k$  that corresponds to  $\mathcal{C}_k$ . The null hypothesis is accepted if [27, 28]

$$\cos \theta_k \geq 1 - \frac{(n_k - R_k)F_{K-1, (n_k-1)(K-1); \alpha}}{(n_k - 1)R_k}. \quad (27)$$

where  $F_{K-1, (n_k-1)(K-1); \alpha}$  is the upper  $\alpha$  percentage point of the  $F$ -distribution with degrees of freedom  $K - 1$  and  $(n_k - 1)(K - 1)$ . A similar hypothesis testing for  $\mathcal{C}_{k+1}$  yields that the null hypothesis is accepted if

$$\cos \theta_{k+1} = \frac{1}{R_{k+1}} \mathbf{r}_{k+1}^T \boldsymbol{\mu}_0 \geq 1 - \frac{(n_{k+1} - R_{k+1})F_{K-1, (n_{k+1}-1)(K-1); \alpha}}{(n_{k+1} - 1)R_{k+1}}. \quad (28)$$

Merging is performed only if the null hypothesis is accepted for both  $\mathcal{C}_k$  and  $\mathcal{C}_{k+1}$  i.e. if (27) and (28) hold simultaneously.

We have obtained the frame clusters which correspond to the shots. The frames between two frame clusters correspond to the gradual shot transitions like dissolve, fade or wipe. An example of dissolve is shown on Figure 2.

## 4 Fade detection

In order to efficiently distinguish between fades and other types of transitions, after performing SVD on the histograms in Section 3.1 we can apply the following method to detect fades. We use a reference black frame  $f_{BK}$ . After performing SVD,  $f_{BK}$  is represented by  $\tilde{\mathbf{v}}_{BK}$ . We can easily detect the frames which are black or very dark as the ones having a large cosine similarity to  $\tilde{\mathbf{v}}_{BK}$ . We put them in the set

$$T = \{\tilde{\mathbf{v}}_i, i = 1, \dots, n \mid \cos(\tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_{BK}) > \delta_f\} \quad (29)$$

where  $\delta_f$  is a predefined threshold. The time instant, where the distance between  $\tilde{\mathbf{v}}_i \in T$  and the reference frame  $\tilde{\mathbf{v}}_{BK}$  is at a local minimum is detected and is characterized as the end-time instant  $t_e$  of the fade-out. Typically the time is measured by the number of video frames. The next step consists in searching for the fade-out starting point  $t_s$  in the preceding frames using the criterion

$$\cos(\tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_{BK}) \geq \cos(\tilde{\mathbf{v}}_{i-1}, \tilde{\mathbf{v}}_{BK}) + \epsilon_f \quad (30)$$

where  $\epsilon_f$  is another predefined threshold. When the first violation of (30) occurs, the frame  $\tilde{\mathbf{v}}_i$  is identified as the starting point of the fade-out  $t_s$ . A similar procedure is also applied for fade-in detection with  $t_s$  being detected first. Finally, since a fade is a gradual transition and has a boundary spread across a number of frames, the video segment is considered as a fade if  $t_e - t_s \geq 2$  (i.e., the spread contains at least 2 video frames). Otherwise, it is labeled

as a shot cut. An example of fade-out and fade-in is shown in Figure 3. It can be clearly seen that a fade-out is a transition from a shot to the reference black frame and the following fade-in is a transition from the reference black frame to another shot.

## 5 Experimental results and discussion

To enable future comparison with other boundary detection techniques, the proposed method was tested on newscasts from the reference video test set TRECVID 2003 [40], containing video sequences of duration longer than 6 hours that have been digitized with a frame rate of 29.97 frames per second (fps) at a resolution of  $352 \times 264$  pixels. 6 video sequence of duration longer than 2 hours from TRECVID 2006 [41] were also added to the testing set to show performance of the method on the most recent content. The TRECVID 2006 videos sequences have been digitized with a frame rate of 29.97 fps at a resolution of  $352 \times 264$  pixels as well. We used downsampled frames with resolution  $176 \times 132$  pixels to speed up the calculations in our experiments. The TRECVID 2003 and TRECVID 2006 ground truth was used as well (see Table 1).

In order to evaluate the performance of the shot cut detection method presented in Section 3, the recall and precision measures, were used [37, 42]. Let  $GT$  denote the ground truth set for the detection task under study,  $Det$  the detected (correctly or falsely) set using our methods. The *Recall* measure, also known as the true positive function or sensitivity, corresponds to the ratio of correct experimental detections over the number of all true detections:

$$Recall = \frac{|Det \cap GT|}{|GT|}, \quad (31)$$

where  $|GT|$  denotes the cardinality of set  $GT$ . The *Precision* measure corresponds to the

accuracy of the method considering false detections and it is defined as the number of correct experimental detections over the number of all experimental detections:

$$Precision = \frac{|Det \cap GT|}{|Det|}. \quad (32)$$

For performance assessment we have also used a combination of recall and precision, the so-called  $F_1$  measure, defined as

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}. \quad (33)$$

At first, we have tested whether the obtained data exhibit a clustering tendency or they are uniformly distributed on the hypersphere after projecting the histograms to the refined feature space using SVD with  $K = 10$ . The motivation for such test is the fact that uniformly distributed data cannot be clustered. This is not the case for our data. We have found that all the frame histograms after projection are located in the  $AOB$  cone, as can be seen in Figure 4. For visualization purposes we have used  $K = 3$  in Figure 4. We have tested the method with several choices of the number of singular values retained  $K \in \{3, 5, 10, 20, 50, 100\}$ . For small values of  $K$ , it is found that information useful for clustering is lost. Based on our experiments, it is found that the best choice is  $K = 10$ . For this value, the shot clusters are separated well. A further increase of  $K$  does not improve clustering.

In the clustering step, we set threshold  $\delta$  so that we get an over-segmentation with the fewest possible miss detections. The false detections are further reduced by applying successively the merging steps of Sections 3.3 and 3.4. By varying threshold  $\delta$  in rule (11) within the range  $[0.80, 0.99]$  we have obtained the recall-precision curves shown in Figure 5. One can see, that the best results are obtained by applying the cluster merging procedure of Section 3.3 followed by the statistical tests for validating cluster merging described in Section 3.4. The procedure described in Section 3.3 alone does not produce always the best

results. In our experiments, we found that a good compromise between recall and precision corresponds to the choice  $\delta = 0.98$ . The corresponding  $F_1$  curves for both cluster merging approaches are drawn in Figure 6. One can see that for  $\delta \in [0.9, 0.99]$  we obtain consistently a higher  $F_1$  measure by applying sequentially the cluster merging procedures of Sections 3.3 and 3.4 on the TRECVID 2003 test set.

For the cluster merging step of Section 3.3 we used a threshold  $\beta$  in (15) less than 1 to guarantee that two consecutive clusters are well separated. Four recall-precision curves obtained for  $\delta \in \{0.993, 0.99, 0.98, 0.97\}$  by varying the threshold  $\beta$  within the range  $[0.962, 0.998]$  are shown in Figure 7. One can see that the recall-precision curve obtained for  $\delta = 0.98$  outperforms the others. In Figure 8, the corresponding  $F_1$  curves are plotted. By fixing  $\delta$  to the most favorable value ( $\delta = 0.98$ ) we can select  $\beta$  so that we optimize  $F_1$ . The best operating point was found to be for  $\beta = 0.98$ . However, one can see that there is not only a single value of  $\beta$  that yields  $F_1$  greater than 0.8 but a whole range of values. Therefore, choosing  $\beta$  is not a difficult task.

In any case, we can set the thresholds experimentally by the leave-one-out method. We performed the leave-one-out method on 6 video sequences so that in every run we left out one video sequence for testing. We obtained for every run one  $F_1$  curve by varying parameter  $\beta$  in range  $[0.971, 0.999]$  while threshold  $\delta$  is fixed to 0.98. In Figure 9 the mean  $F_1$  curve is plotted. The interval of  $\pm$  one standard deviation of the  $F_1$  measure is indicated with bars overlaid. Then, we performed the tests on one unseen video sequence of the same category, namely TV news. The corresponding  $F_1$  curve is plotted in Figure 9. One can see that the  $F_1$  curve on the unseen video sequence having fixed  $\delta$  to 0.98 follows the general trends of the mean  $F_1$  curve determined by leave on out method. All the thresholds used in our method are independent to frame rates and image resolution.

Table 2 summarizes the recall and precision measures for cuts, fades, and others gradual transitions (i.e., dissolves and wipes) using  $\delta = 0.98$ ,  $\beta = 0.98$ , and  $K = 10$ . By applying a similar approach to that used for determining the most suitable values of  $\delta$  and  $\beta$ , we found that  $\delta_f$  in (29) and  $\epsilon_f$  in (30) used for fade detection, should be chosen 0.999 and 0.97, respectively. The large majority of transitions were correctly detected.

We compared our results for abrupt cuts to the ones obtained by a technique proposed in [7]. This approach combines two standard shot boundary detection schemes based on color frame differences and color vector histogram differences between successive frames. The method operates in the HLS color space and ignores the luminance information in order to overcome the possible drawback of histogram sensitivity to shot illumination changes. In order to detect cuts an adaptive thresholding method was employed. The recall-precision curve obtained by varying the threshold of the method in [7] is shown in Figure 10. The results of this algorithm applied to the same video sequences are summarized in Table 3. The abrupt cut detection rate of this approach is inferior to that of the proposed method (Table 2). However, the method proposed in [7] fails to detect gradual transitions.

Using only histogram differences of successive frames we are not able to detect the gradual transitions between the shots, because in this case, the changes are small. Camera and object motion can introduce a larger variation than a gradual transition. Using SVD and projecting the histograms to the refined feature space gives us the opportunity to distinguish between motion and transition and to avoid the false alarms, even without using any motion information or motion compensation. In our approach, after projecting the histograms to the reduced feature space, the shots with small camera motion and object movements inside the shot form a cluster having small dispersion, while the shots with some action inside form clusters with a large dispersion as can be seen in Figure 11. The transition between two

shots is shown as a path between the two dense clusters. An example of a dissolve pattern in the projected space is shown in Figure 2. Therefore, we can easily detect and distinguish between transitions and shots. Using the fade detection method described in Section 4 we can successfully detect and distinguish between fades and other types of gradual transitions. This increases the precision of shot transition detection.

To support that SVD is essential, we have performed the clustering and merging steps directly on the full length vectors (histograms) of dimensions  $4096 \times 1$ . In Figure 12, one can see that at precision rate 0.80 the recall rate improved by more than 5% when SVD is used due to its noise reduction capabilities. To calculate SVD for matrix  $4096 \times 10000$  takes 79.53 seconds. In any case, one could limit SVD to a subset of singular values (i.e. those of interest) and the corresponding singular vectors. The elapsed time for one video sequence having 50253 frames was 326 seconds for performing SVD and another 15 seconds to perform the clustering and merging steps. Whereas, clustering and merging directly on the full length vectors of the same video sequence took 693 seconds. Therefore, SVD speeds up shot boundary detection two times. Thus, by using SVD, we gain time and obtain a higher performance than when we process raw color histograms.

The results obtained for abrupt cut detection are within the best three results reported for TRECVID 2003. The best results obtained by Amir at IBM group were 0.94 and 0.95 for recall and precision, respectively. The corresponding  $F_1 = 0.945$ . The features used in the work of Amir are 3D RGB color histograms, 3D localized edges direction histogram, gray-level thumbnails, average frame luminance, black and monochrome detector. Differences with a number of previous frames and an adaptive threshold based on the average value in a 61 frame window are computed. All these features are then fused by a completely heuristic finite-state machine. Second and third best results were reported by the Uni-



versity of Iowa and CLIPS-IMAG. The group from University of Iowa used basic methods as comparison of adjacent frames based on 512-bin global color histogram, frame color distance similarity, Sobel filtering, and detected edge differences. Then boolean predicate and arithmetic product of the basic methods were employed. They obtained 0.89 and 0.98 for recall and precision, respectively. The corresponding  $F_1 = 0.933$ . The method of CLIPS group is based on image differences with motion compensation which uses optical flow as a pre-process and direct detection of dissolves. It also includes direct detection of camera flashes. They obtained 0.90 and 0.92 for recall and precision, respectively. The corresponding  $F_1 = 0.909$ . The  $F_1$  obtained by our method for abrupt cut detection is 0.914. The results of the groups obtained for the abrupt cuts are summarized in Table 4. An improvement of the results reported for our method can be obtained by fusing different features, as we did in [43]. The results for gradual transitions obtained by our method are comparable to the best results published in the TRECVID 2003 competition [40], which were obtained by Amir at IBM Research [44]. In the case of gradual transitions, for a recall fixed to 0.84, the corresponding precision is found to be 0.76 ( $F_1 = 0.798$ ), whereas our method produces a precision of 0.80 at the same recall rate ( $F_1 = 0.819$ ). The results of the groups obtained for the gradual transition (including dissolves, fades and wipes) are summarized in Table 5. The results from TRECVID2003 were not overcome in the competition TRECVID2004 where similar video sequence (CNN and ABC news from year 1998) were used. It should be noted that the TRECVID competition places greater importance on the quality and stability of the results than the originality and the theoretical foundation of the methods.

We have performed tests on 6 video sequences used in the latest TRECVID 2006 to assess the performance of the method on the most recent content without further tuning the several thresholds employed in our method. This test set contains news video sequences from CNN

and NBC, and contrary to the TRECVID 2003, chinese and arabic news videos are added. The obtained results for abrupt cuts and the gradual transitions (including fades, dissolves and wipes) are shown in Table 6. In this Table the 2 best results reported on TRECVID 2006 are also included. One can see that our results are not far behind the best ones.

However, the presented method is sensitive to camera flashes because a matrix constructed from the color histograms of frames is used as an input matrix for SVD. Color frame histograms are affected by the flashes and SVD cannot smooth them efficiently. Figure 13 shows a camera flash inside the shot. We can see that the frames after the flash belong to the same cluster as the ones before. We decided to retain only the frame clusters having more than 5 frames in order to avoid false positives which appear due to camera flashes. In the merging phase, these two clusters, which were initially separated by the flash frames, are merged together since they have similar mean directions.

The proposed method inherits the weaknesses of the histogram-based algorithms. This fact can induce false transition rejections. In some cases, such missed transitions appear in commercials containing artistic camera edits. Similarly, consecutive dissolves between shots of short duration that have a similar color distribution were not detected correctly in our experiments.

## 6 Conclusions

A new technique for automated shot transition detection using Singular Value Decomposition and clustering was presented. The detection technique was tested on TRECVID video sequences having various types of shots and significant object and camera motion inside the shots. The experiments demonstrated that, by using the projected feature space we can

efficiently differentiate between gradual transitions and cuts, pans, object or camera motion, while most of the methods based on histograms fail to characterize these types of video transitions. Several variations of the proposed method can be developed. For example the feature vector can be constructed from different features describing the video context or their combination. Different clustering method could also be incorporated.

## **7 Acknowledgment**

This work has been supported by two FP6 European Networks of Excellence, funded under the European Commission IST FP6 program; VISNET “Networked Audiovisual Media Technologies” and MUSCLE “Multimedia Understanding through Semantics, Computation, and Learning (FP6-507752).

The C-SPAN video used in this work is provided for research purposes by C-SPAN through the TREC Information-Retrieval Research Collection. C-SPAN video is copyrighted.

# References

- [1] A. D. Bimbo, *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc, San Francisco, California, 1999.
- [2] X. U. Cabedo and S. K. Bhattacharjee, “Shot detection tools in digital video,” in *Proc. Non-linear Model Based Image Analysis 1998*, Springer Verlag, Glasgow, July 1998, pp. 121–126.
- [3] R. Lienhart, “Reliable dissolve detection,” in *Proc. SPIE Storage and Retrieval for Media Databases 2001*, vol. 4315, January 2001, pp. 219–230.
- [4] A. Dailianas, R. B. Allen, and P. England, “Comparison of automatic video segmentation algorithms,” in *Proc. SPIE Photonics East’95: Integration Issues in Large Commercial Media Delivery Systems*, vol. 2615, Philadelphia 1995, Oct. 1995, pp. 2–16.
- [5] G. Ahanger and T. Little, “A survey of technologies for parsing and indexing digital video,” *Journal Visual Communication and Image Representation*, vol. 7, no. 1, pp. 28–43, 1996.
- [6] N. V. Patel and I. K. Sethi, “Video shot detection and characterization for video databases,” *Pattern Recognition*, vol. 30, no. 4, pp. 583–592, April 1997.
- [7] S. Krinidis, S. Tsekeridou, and I. Pitas, “Multimodal interaction for scene boundary detection,” in *Proc. IEEE Int. Conf. Nonlinear Signal and Image Processing (NSIP01)*, Baltimore, Maryland, USA, 3-6 June 2001.

- [8] R. Lienhart, “Comparison of automatic shot boundary detection algorithms,” in *Proc. SPIE Storage and Retrieval for Image and Video Databases VII*, vol. 3656, San Jose, CA, U.S.A. January 1999, pp. 290–301.
- [9] J. Nesvadba, F. Ernst, J. Perhac, J. Benois-Pineau, and L. Primaux, “Comparison of shot boundary detectors,” in *Proc. Int. Conf. on Multimedia and Expo (ICME05)*, Amsterdam, The Netherlands, 2005, pp. 788–791.
- [10] T. Butz and J. Thiran, “Shot boundary detection with mutual information,” in *Proc. 2001 IEEE Int. Conf. Image Processing*, vol. 3, Greece, October 2001, pp. 422–425.
- [11] 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.
- [12] Z. Cernekova, I. Pitas, and C. Nikou, “Information theory-based shot cut/fade detection and video summarization,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82–91, January 2006.
- [13] C. L. Huang and B. Y. Liao, “A robust scene-change detection method for video segmentation,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11 no.12, pp. 1281–1288, 2001.
- [14] 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. Image Processing*, vol. 3, 2000, pp. 929–932.

- [15] 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.
- [16] R. Lienhart and A. Zaccarin, "A system for reliable dissolve detection in video," in *Proc. 2001 IEEE Int. Conf. Image Processing, Thessaloniki, Greece*, vol. 3, Oct. 2001, pp. 406–409.
- [17] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *ACM Journal Multimedia Systems*, vol. 7, pp. 119–128, 1999.
- [18] A. M. Alattar, "Detecting fade regions in uncompressed video sequences," in *Proc. 1997 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1997, pp. 3025–3028.
- [19] B. T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *ACM Multimedia 2000*, November 2000, pp. 219–227.
- [20] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?" *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12 no.2, pp. 90–105, 2002.
- [21] S. Porter, M. Mirmehdi, and B. Thomas, "Detection and classification of shot transitions," in *Proc. The 12th British Machine Vision Conference, BMVA Press*, 2001, pp. 73–82.
- [22] Z. Cernekova, C. Kotropoulos, and I. Pitas, "Video shot segmentation using singular value decomposition," in *Proc. 2003 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. III, Hong-Kong, April 2003, pp. 181–184, (appears also in *Proc. 2003 IEEE Multimedia and Expo*, pp. 301–304, Baltimore, July 2003.).

- [23] E. Sahouria and A. Zakhori, “Content analysis of video using principal components,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1290–1298, December 1999.
- [24] A. Yilmaz and M. A. Shah, “Shot detection using principal coordinate system,” in *Proc. 2000 Int. Conf. Internet and Multimedia Systems and Applications, IMSA*, Las Vegas, Nov 20-23, 2000, p. 168.
- [25] X. Liu and T. Chen, “Shot boundary detection using temporal statistics modeling,” in *Proc. 2002 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, Orlando, Florida, USA, May, 2002, pp. 3389–3392.
- [26] Y. Gong and X. Liu, “Video shot segmentation and classification,” in *Proc. 2000 Int. Conf. Pattern Recognition*, vol. I, 2000, pp. 860–863.
- [27] N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical Analysis of Spherical Data*. Cambridge, U.K.: Cambridge Univ. Press, 1993.
- [28] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*, 2/e. London, UK: Academic Press, 1980.
- [29] R. Zhao and W. I. Grosky, “A novel video shot detection technique using color anglogram and latent semantic indexing,” in *Proc. 23rd Int. Conf. Distributed Computing Systems Workshops (ICDCSW ’03)*, Providence, Rhode Island, 2003, pp. 550–555.
- [30] F. Souvannavong, B. Merialdo, and B. Huet, “Video content modeling with latent semantic analysis,” in *Proc. 3rd Int. Workshop Content-Based Multimedia Indexing*, September 22 - 24, 2003.

- [31] Y. Rui, T. Huang, and S. Chang, “Image retrieval: past, present and future,” *Journal of Visual Communication and Image Representation*, vol. 10, pp. 1–23, 1999.
- [32] W. Ren, M. Singh, and S. Singh, “Automated video segmentations,” in *Proc. 3rd IEEE Int. Conf. Information, Communications & Signal Processing, ICICS 2001, Singapore*, October 2001.
- [33] I. Ide, R. Hamada, S. Sakai, and H. Tanaka, “Relating graphical features with concept classes for automatic news video indexing,” in *Proc. 1999 Workshop Intelligent Information (IJCAI-99), Stockholm, Sweden*, 1999.
- [34] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal American Society for Information Science*, vol. 41 no. 6, pp. 391–407, 1990.
- [35] J. R. Bellegarda, “Exploiting latent semantic information in statistical language modeling,” *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, August 2000.
- [36] C. O’Toole, A. Smeaton, N. Murphy, and S. Marlow, “Evaluation of automatic shot boundary detection on a large video test suite,” in *Proc. The Challenge of Image Retrieval (CIR99) – 2nd UK Conf. Image Retrieval*, Newcastle, UK, February 25–26, 1999.
- [37] R. R. Korfhage, *Information Storage and Retrieval*. New York: J Wiley, 1997.
- [38] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining, AAAI Press*, 1996, pp. 226–231.



- [39] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using von Mises-Fisher distributions,” *Journal of Machine Learning Research*, vol. 6, pp. 1345–1382, 2005.
- [40] “TREC video retrieval evaluation,” 2003. [Online]. Available: <http://www-nlpir.nist.gov/projects/tv2003/tv2003.html>
- [41] “TREC video retrieval evaluation,” 2006. [Online]. Available: <http://www-nlpir.nist.gov/projects/tv2006/tv2006.html>
- [42] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. N.Y.: Academic Press, 1990.
- [43] Z. Cernekova, C. Kotropoulos, N. Nikolaidis, and I. Pitas, “Video shot segmentation using fusion of svd and mutual information features,” in *Proc. 2005 IEEE Int. Symp. Circuits and Systems (ISCAS)*, Kobe, Japan, 23-26 May, 2005, pp. 3849–3852.
- [44] A. Amir, “The IBM shot boundary detection system at TRECVID 2003,” in *TREC Video Retrieval Evaluation*, Nov. 2003.



[Zuzana Černeková received the Diploma of Master of Science in 1999 from Comenius University, Bratislava, Slovakia.

She has studied informatics with specialization on Mathematics methods of informatics and Computer Graphics. She took Doctor of Natural sciences (RNDr.) in 2000. She was a researcher and lecture assistant at the Department of Computer Graphics and Image Processing, Faculty of Mathematics and Physics, Comenius University, Bratislava, Slovakia. She has also conducted research in Trinity College Dublin, Ireland during the summer of 2002 and in ZGDV Germany in May 2007. She has published 17 papers and contributed in a book chapter in her areas of expertise. Her research interests lie in the areas of computer graphics, visualization, multimedia, video processing and pattern recognition. She is currently a pre-doc researcher and Ph.D. student at the Department of Informatics, Aristotle University of Thessaloniki, Greece.

Ms. Černeková is/was a member of the SCCG organizing committee.




[Constantine Kotropoulos received the received the Diploma degree with honors in Electrical Engineering in 1988 and the PhD degree in Electrical & Computer Engineering in 1993, both from the Aristotle University of Thessaloniki.

Since 2002 he has been an Assistant Professor in the Department of Informatics at the Aristotle University of Thessaloniki. From 1989 to 1993 he was a research and teaching assistant in the Department of Electrical & Computer Engineering at the same university.

In 1995, he joined the Department of Informatics at the Aristotle University of Thessaloniki as a senior researcher and served then as a Lecturer from 1997 to 2001. He has also conducted research in the Signal Processing Laboratory at Tampere University of Technology, Finland during the summer of 1993. He has authored 28 journal papers, 135 conference papers, and contributed 5 chapters to edited books in his areas of expertise. He is co-editor of the book “Nonlinear Model-Based Image/Video Processing and Analysis” (J. Wiley and Sons, 2001). His current research interests include speech, audio, and language processing; signal processing; pattern recognition; multimedia information retrieval; biometric authentication techniques, and human-centered multimodal computer interaction.

Prof. Kotropoulos was a scholar of the State Scholarship Foundation of Greece and the Bodossaki Foundation. He is a senior member of the IEEE and a member of EURASIP, IAPR, ISCA, and the Technical Chamber of Greece.



[] Ioannis Pitas (SM'94) received the Dipl. Elect. Eng. in 1980 and the Ph.D. degree in electrical engineering in 1985, both from the University of Thessaloniki, Thessaloniki, Greece.

Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate or Visiting Assistant Professor at several Universities. He has published 153 journal papers, 400 conference papers and contributed in 22 books in his areas of interest and edited or coauthored another 5. He

is/was principal investigator/researcher in more than 40 competitive R&D projects and in 11 educational projects, all mostly funded by the European Union.

Dr. Pitas has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as Associate Editor or co-Editor of four international journals and General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.

Table 1: *Video sequence used and the respective numbers of frames, abrupt cuts, fade-ins, fade-outs, and other transitions.*

TRECVID2003 video sequences					
video	frames	cuts	fade-ins	fade-outs	other transitions
<b>6 debate videos</b>	125977	230	0	0	0
<b>4 CNN news</b>	209978	1287	57	57	464
<b>4 ABC news</b>	206144	1269	64	69	553
TRECVID2006 video sequences					
video	frames	cuts	gradual transitions		
<b>6 news videos</b>	225564	1292	664		

## 8 List of figure captions

1. *Consecutive frames from “news” video sequence showing: (a) a fade-out, (b) a fade-in and (c) a dissolve.*
2. *Projected frame histograms on the subspace defined by the fifth and sixth singular vectors reveal a dissolve pattern between two shots.*
3. *Fade detection visualized on the subspace defined by the first and second left singular vectors.*
4. *Three-dimensional visualization of video frames obtained after SVD in the subspace formed by the first three left singular vectors. The inner cone shows the placement of fades.*
5. *Recall-precision curves by varying the threshold  $\delta$  in rule (11) on TRECVID 2003 video*

Table 2: Shot detection results using the method described in Sections 3-4.

TRECVID2003 video sequences						
video	cuts		fades		other transitions	
	Recall	Precision	Recall	Precision	Recall	Precision
<b>6 debate videos</b>	1.00	1.00	-	-	-	-
<b>4 CNN news</b>	0.89	0.93	0.86	0.89	0.86	0.76
<b>4 ABC news</b>	0.86	0.93	0.94	0.97	0.79	0.77
<b>TRECVID2003 total</b>	0.89	0.94	0.90	0.93	0.82	0.77

Table 3: Shot detection results obtained for abrupt cuts using the method presented in [7].

Histogram based method		
video	Recall	Precision
<b>6 debate videos</b>	1.00	1.00
<b>4 CNN news</b>	0.87	0.83
<b>4 ABC news</b>	0.85	0.81
<b>TRECVID2003 total</b>	0.87	0.83

Table 4: Shot detection results obtained for abrupt cuts reported in TRECVID 2003.

Abrupt cut detection in TRECVID2003			
Group	Recall	Precision	$F_1$
<b>IBM</b>	0.94	0.95	0.945
<b>Univ. of Iowa</b>	0.89	0.98	0.933
<b>AUTH</b>	<b>0.89</b>	<b>0.94</b>	<b>0.914</b>
<b>CLIPS</b>	0.90	0.92	0.909

Table 5: *Shot detection results obtained for gradual transitions reported in TRECVID 2003 for recall fixed to 0.84.*

Gradual transition detection in TRECVID2003			
Group	Recall	Precision	$F_1$
IBM	0.84	0.76	0.798
AUTH	<b>0.84</b>	<b>0.80</b>	<b>0.819</b>

Table 6: *Shot detection results using the method described in Sections 3-4 compared to the results reported in TRECVID 2006.*

TRECVID2006 video sequences						
video	cuts			gradual transitions		
	Recall	Precision	$F_1$	Recall	Precision	$F_1$
AUTH	<b>0.91</b>	0.83	0.868	<b>0.80</b>	0.77	0.785
AT&T Labs	0.86	0.92	0.889	0.77	<b>0.85</b>	<b>0.808</b>
Tsinghua University	0.87	<b>0.93</b>	<b>0.899</b>	0.75	0.81	0.778

test set, when the projected frame histograms are clustered by applying the procedure of Section 3.2 and the resulting clusters are merged by either the ad hoc procedure of Section 3.3 alone or the sequential application of the ad hoc procedure of Section 3.3 and the statistical tests of Section 3.4. Running  $\delta$  values are shown overlaid.

6.  $F_1$  measure obtained for varying  $\delta$  in rule (11) on TRECVID 2003 video test set.
7. Recall-precision curves by varying the threshold  $\beta$  in range  $[0.962, 0.998]$  for several  $\delta \in \{0.993, 0.99, 0.98, 0.97\}$  on TRECVID 2003 video test set.
8.  $F_1$  measure related to the recall-precision curves in Fig.7 by varying the threshold  $\beta$  in (15).
9. Mean  $F_1$  curve for the leave-one-out method with  $\pm$  one standard deviation intervals obtained from 6 video sequences by varying threshold  $\beta$  in (15) in range  $[0.971, 0.999]$  and fixed  $\delta = 0.98$ ; The  $F_1$  curve for one unseen video sequence of the same category - TV news is also plotted overlaid.
10. Recall-precision graph by varying the threshold used in [7] in the range  $[2.2, 2.7]$  for detecting cuts on TRECVID 2003 video test set.
11. Projected frame histograms onto the subspace defined by the fifth and sixth left singular vectors reveal the intra-shot dispersion due to motion.
12. Recall-precision curve obtained for  $\delta = 0.98$  and varying  $\beta$  in range  $[0.998, 0.96]$  by applying the SVD and the clustering and merging steps and 2 recall-precision curves obtained by applying the clustering and merging steps directly on the histograms of dimensionality 4096 for  $\delta \in \{0.96, 0.98\}$  and  $\beta$  varying in range  $[0.7, 0.98]$  on TRECVID 2003 video test set.



13. *Projected frame histograms onto the subspace defined by the fifth and sixth left singular vectors reveal a flash pattern within a shot.*