Fast Video Shot Boundary Detection Based on SVD and Pattern Matching

Zhe-Ming Lu and Yong Shi

***Abstract*— Video shot boundary detection (SBD) is the first and essential step for content-based video management and structural analysis. Great efforts have been paid to develop SBD algorithms for years. However, the high computational cost in the SBD becomes a block for further applications such as video indexing, browsing, retrieval, and representation. Motivated by the requirement of the real-time interactive applications, a unified fast SBD scheme is proposed in this paper. We adopted a candidate segment selection and singular value decomposition (SVD) to speed up the SBD. Initially, the positions of the shot boundaries and lengths of gradual transitions are predicted using adaptive thresholds and most non-boundary frames are discarded at the same time. Only the candidate segments that may contain the shot boundaries are preserved for further detection. Then, for all frames in each candidate segment, their color histograms in the hue-saturation-value) space are extracted, forming a frame-feature matrix. The SVD is then performed on the frame-feature matrices of all candidate segments to reduce the feature dimension. The refined feature vector of each frame in the candidate segments is obtained as a new metric for boundary detection. Finally, cut and gradual transitions are identified using our pattern matching method based on a new similarity measurement. Experiments on TRECVID 2001 test data and other video materials show that the proposed scheme can achieve a high detection speed and excellent accuracy compared with recent SBD schemes.**

***Index Terms*— Fast shot boundary detection, adaptive thresholds, dimensionality reduction, cut transition detection, gradual transition detection, pattern matching.**

I. Introduction

With the development of computer science and information technology, digital information has an explosive growth. As a result, digital video information keeps increasing and so do the new applications, such as video-on-demand, motion analysis, intelligent surveillance, video conference, and Internet TV. Some new multimedia technologies have been employed to deal with the problems arising in the increasing applications of video information. Video indexing, browsing, retrieval, representation and other video technologies for content management have been researched for years [1], [2]. As a video sequence always contains diverse and complex information, effective management of digital video information has been a challenge of the multimedia industry. Video shot boundary detection (SBD), also called video temporal segmentation, as a part of video structural analysis, is the first and essential step for further technologies because it is intrinsically and inextricably linked to the way of organizing the video. A video shot is defined as a sequence of frames taken by a single camera in an uninterrupted run [3]. Video shots are the basic units of a video. Usually, a video shot has a group of continuous frames which have similar information and visual features such as colors, motions and textures. Transition manners between shots have two basic types: cut transition (CT, or abrupt change) and gradual transition (GT). For cut transition, the next shot appears immediately right after the last frame of the previous shot. On the contrary, in gradual transition, the neighboring frames in two consecutive shots change in a mild way such as dissolve, fade in/out, wipe and other gradual effects. Therefore, gradual transition contains several frames that have interrelated visual information. Video shot boundary detection is a process of identifying the transition between two adjacent shots [4].

Most early works on shot boundary detection focused on cut detection, considering the phenomenon that two consecutive frames around the abrupt cut have great discontinuities. As a result, these approaches usually detected a cut transition when a feature distance measure between consecutive frames exceeds a threshold. Pixel difference [5], [7] is a kind of common used feature in the uncompressed domain. Considering the sensitivity to object and camera motions, many researchers adopted intensity or color histograms as features based on global information [8], [9]. Cuts are detected by comparing the variations between two consecutive histograms. Comparisons among such kind of detection algorithms are given in [10]. In the compressed domain, elements such as DCT (discrete cosine transform) coefficients [11], [12], motion vectors [13], can be also used as features.

Different from cut detection, gradual transition detection is much more complex. First, a gradual transition between two consecutive shots usually lasts for a while, and the difference between consecutive frames is not as obvious as in cut transitions. Second, GT has different types, e.g. dissolve, fade in/out, and wipe, thus the GT detector should be able to deal with the diversity of GT effects. Third, the GT detection is more likely to be confused by object motions and camera operations than cut detection. Various algorithms [7], [14], [15] detect GT effects by evaluating the degree to which the transition matches the corresponding model. In [16], cut and fade transitions are

Manuscript received April 29, 2012; revised November 27, 2012 and July 17, 2013; accepted September 10, 2013. Date of publication September 16, 2013; date of current version October 4, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ton Kalker.

The authors are with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China (e-mail: zheminglu@zju.edu.cn; gydyt@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2013.2282081

detected based on mutual information and the joint entropy between consecutive frames. Edge information [17], [18] has also been used in shot detection. In [26], Petersohn adopted the edge energy of MPEG DC-coefficients to determine dissolve candidates by searching for U-shapes diagrams. One limitation of most GT detection algorithms is that they focus on only one specified GT effect, which makes the whole GT detection more complex. Thus, unified gradual transition detection algorithms are desired in practical applications. Reference [19] presents the common characteristics of a gradual transition, and Reference [20] has derived an iteration algorithm based on these characteristics.

Besides the detection accuracy, the high computation complexity of detection algorithms, especially for GT, is the most protuberant bottleneck in real-time applications. In [21], Li *et al.* employ preprocessing techniques to segment video and select candidate segments which are considered as suspect shot change fragments, then the locations of cut and gradual transitions are obtained approximately. Then, Li *et al.* improved the LTD algorithm [20] to detect the gradual transitions after preprocessing. This method has great breakthrough on the computational speed. However, in some complex cases, the adaptive threshold used in the preprocessing stage does not perform so well. Singular value decomposition (SVD) is an algebraic tool to reduce the feature dimension and extract principal components. In [22] and [23], SVD is applied to shot boundary detection, where frame clustering with refined feature vectors is employed. However, their main drawback is the time overhead of SVD on large matrices, typically 4096×10000-sized [23]. In our proposed scheme, the candidate segment selection and SVD are also employed, but meanwhile we overcome their drawbacks and propose a new algorithm to detect both cut and gradual transitions with low computational complexity. Experiments on video data from different types show that the proposed scheme can provide high speed and accuracy for detecting both abrupt and gradual transitions.

The outline of this paper is as follows: Section II introduces the preliminaries of our scheme, including candidate segment selection, SVD-based dimension reduction and feature extraction. In Section III, our detection algorithms are illustrated in detail. Experimental results and comparisons are presented in Section IV. Finally, we draw conclusions in Section V.

II. Preliminaries

In this section, the two preliminary processes in our scheme, i.e., SVD-based feature extraction and candidate segment selection, are presented. These two steps contribute to high processing speed.

A. Singular Value Decomposition and Feature Extraction

The SVD is a useful technique in linear algebra. A matrix can be viewed as a transformation from one feature space to another, and singular values reveal principle information of the matrix. Primary structure of a matrix or a transformation can be extracted by using singular value decomposition.

The SVD of an $M \times N$ matrix $\boldsymbol{A}$ is defined as follows:

$$\boldsymbol{A} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{U}$ is an $M \times M$ matrix whose columns are the eigenvectors of the matrix $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ that are termed the left eigenvectors, $\boldsymbol{V}$ is an $N \times N$ matrix whose columns are the eigenvectors of the $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ matrix that are termed the right eigenvectors, and $\boldsymbol{\Sigma}$=diag($\boldsymbol{\Sigma}_1$,$\boldsymbol{\Sigma}_2$, . . ., $\boldsymbol{\Sigma}_R$) is a matrix whose diagonal elements $\boldsymbol{\Sigma}_1 \geq \sigma_2 \geq \cdots \geq \sigma_R \geq 0$ with R =min(M, N) are called the singular values of $\boldsymbol{A}$ that are the largest R square roots of eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ or $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$. The property of SVD is related to the rank of the matrix $\boldsymbol{A}$, i.e., there are only r=rank ($\boldsymbol{A}$) singular values that are nonzero. Thus, we have

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{U}_r & \boldsymbol{U}_o \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_r^T \\ \boldsymbol{V}_o^T \end{bmatrix} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^T \tag{2}$$

Here, $\boldsymbol{U_r}$ is an $M \times r$ matrix, $\boldsymbol{V}_r^{\mathrm{T}}$ is an $r \times N$ matrix and $\boldsymbol{\Sigma}_r$ = diag($\boldsymbol{\Sigma}_1$, . . .,$\boldsymbol{\Sigma}_r$) is a diagonal matrix respectively. The magnitude of a singular value is closely related to the importance of the corresponding eigenvectors in the matrices $\boldsymbol{U}_r$ and $\boldsymbol{V}_r^{\mathrm{T}}$, i.e., the larger a singular value is, the more important the corresponding eigenvectors are in the matrix $\boldsymbol{A}$. Since the singular values are sorted in the descending order, and generally, the first several singular values are much greater than the following ones. The first several singular values are much greater than the following ones. For a large-sized $M \times N$ matrix $\boldsymbol{A}$, keeping the first k singular values (usually $k \ll M$) is equivalent to keeping the primary components of the original matrix $\boldsymbol{A}$. For a given parameter k, if we preserve the first k elements of $\boldsymbol{\Sigma}_r$ to form a resulting matrix $\boldsymbol{\Sigma}_k$, keeping the corresponding eigenvectors in $\boldsymbol{U}_r$ and $\boldsymbol{V}_r^{\mathrm{T}}$ respectively, thus we can obtain $\boldsymbol{U}_r\ \boldsymbol{\Sigma}_k \boldsymbol{V}_r^{\mathrm{T}}$as a refined form of $\boldsymbol{A}$. According to these characteristics for dimensionality reduction, SVD has been successfully employed in principal components analysis (PCA), latent semantic indexing (LSI), data compression, noise reduction, and so on.

In our work, the normalized hue-saturation-value (HSV) color histograms are adopted as frame features. Histograms are good at detecting global differences between frames [22], since they are insensitive to the positions of objects [10], and thus they perform well at a moderate computational cost [24]. Existing works have shown that color histograms computed in the HSV color space can generally lead to good performance in SBD [24], [25]. For a 24-bit image, the transformation from the RGB space to the HSV space is as follows:

$$V = \max(R, G, B) \tag{3}$$

$$S = \begin{cases} \frac{V - \min(R,G,B)}{V} \times 255, & \text{if } V \neq 0 \\ 0, \text{else} \end{cases} \tag{4}$$

$$H = \begin{cases} \frac{30(G-B)}{S} & \text{if } V = R \\ 60 + \frac{30(B-R)}{S} & \text{if } V = G \\ 120 + \frac{30(R-G)}{S} & \text{if } V = B \end{cases} \tag{5}$$

Here, if H<0, then we should set $H = H$+180. Thus, the H, S and V values fall in the intervals [0, 180], [0, 255] and [0, 255] respectively. We obtain the color histogram by quantizing H, S and V components into 18, 12, and

8 bins respectively, resulting in 1728 bins totally. Reference [24] shows that 3D histograms perform well and the luminance component is an important feature in shot separation. The purpose of using 8 bins for the V component is to tolerate the change in light intensity.

For N (N is generally less than 100 in our work) consecutive frames in a video sequence, we extract an M (namely, M=1728) bins histogram from the ith frame as an M-dimensional vector $\boldsymbol{a}_i$, forming an $M\times N$ "frame-feature" matrix $\boldsymbol{a} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{A}_N]$ for N frames together, where each column vector represents the information from each frame.

After performing the SVD on the matrix $\boldsymbol{A}$, color histograms are mapped onto orthonormal column vectors of left singular matrix $\boldsymbol{U}_r$. And N frames are mapped onto column vectors of $\boldsymbol{\Sigma}_k \boldsymbol{V}_r^{\mathrm{T}}$ as coordinates of this refined feature space, i.e.,

$$[\boldsymbol{a}_1, ..., \boldsymbol{a}_N] = [\boldsymbol{u}_1, ..., \boldsymbol{u}_r][\boldsymbol{v}_1^T \boldsymbol{\Sigma}_r, ..., \boldsymbol{v}_N^T \boldsymbol{\Sigma}_r] \tag{6}$$

Where $\boldsymbol{U}_j$ (j=1, ...,r) is the column vector of $\boldsymbol{U}_r$, and $\boldsymbol{a}_i$, $\boldsymbol{V}_i$ (i=1,..., N) are the column vector of $\boldsymbol{A}$ and the row vector of $\boldsymbol{V}_r$ respectively. As discussed previously, if we preserve the k largest singular values of $\boldsymbol{\Sigma}_r$, where $k \leq r \cdot M$, we can map the original vector $\boldsymbol{a}_i$ from the M-dimensional space onto a k-dimensional vector $\boldsymbol{\Sigma}_i$:

$$\boldsymbol{\beta}_i = \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{\Sigma}_k, i = 1, 2, \ldots, N \tag{7}$$

The refined feature space keeps the primary information and removes noise or unimportant variations between consecutive frames. Therefore, the frames belong to different shots are easier to be detected. Cosine distance $\Phi(f_i, f_j)$ is defined as the similarity between two frames f_i and f_j by calculating the cosine of the angle of two mapped vectors $\boldsymbol{\beta}_i$ and $\boldsymbol{\Sigma}_j$:

$$\Phi(f_i, f_j) = \cos(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = \frac{(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)}{||\boldsymbol{\beta}_i|| \cdot ||\boldsymbol{\beta}_j||} \tag{8}$$

Obviously, the more similar two vectors are, the closer their cosine distance is. On the contrary, for two frames with many differences, their cosine distance is quite small. The Euclidean distance can also lead to the same purpose, but it needs the normalization operation to fit the further steps well. And the normalization step needs more computational cost. Here, the reason why we choose the cosine distance is that the range of it is 0 to 1, which is more suitable for our algorithm.

B. Candidate Segment Selection

In general, a video sequence with some shots inside has a lot of non-boundary frames and several boundary frames. Candidate segment selection is the first step in our scheme. Its main purpose is to eliminate the non-boundary frames and to reduce the computational complexity in the subsequent steps. Candidate segment selection is based on the fact that consecutive frames in one short temporal segment within a shot always have high correlations [21]. Thus if the first and last frames of a segment are found to be highly similar, this segment can be considered as a non-boundary segment.

In our scheme, we adopt most of the steps of the pre-processing technique in [21], but we adopt a new adaptive threshold to make it suitable for complex video content. The candidate segment selection can be illustrated as follows:

Step 1: Segment the video sequence into segments of length 21. Calculate the distance between the first and last frames of each segment respectively as follows:

$$\begin{aligned} &d(20n, 20(n+1)) \\ &\quad = \sum_x \sum_y |F(x, y; 20n) - F(x, y; 20(n+1))| \end{aligned} \tag{9}$$

where $F(x, y; k)$ represents the intensity of the pixel at the position (x, y) of Frame k. In the following description, we call $d(20n,20(n+1))$ the segment distance for the n-th segment and denote it as $d^{20}(n)$ for short. Here we choose the differences in pixel intensity as the frame distance for the reason that it is a common feature and it is simple to calculate.

Step 2: Gather every consecutive ten segments into one group and calculate the adaptive threshold (T_{L}) for each group:

$$T_L = \mu_L + a\left(1 + \ln\left(\frac{\mu_G}{\mu_L}\right)\right)\sigma_L \tag{10}$$

where μ_{G} denotes the global mean value of all $d^{20}(n)$ in the whole video, μ_{L} denotes the local mean value of all $d^{20}(n)$ in one group, $\boldsymbol{\Sigma}_{\mathrm{L}}$is the local standard deviation in one group.

In some cases, when several cut transitions exist in a segment with relatively few frames, e.g. 3 or 4 cut transitions in a 200 frames segment, CTs will be lost through the original T_{L} in [21], i.e., T_{L}=1.1μ_{L}+0.6(μ_{G}/μ_{L})σ_{L}. Compared with the original (μ_{G}/μ_{L}), 1+ln(μ_{G}/μ_{L}) in our equation is smaller. To guarantee that all the shot boundaries are selected into the candidate segment list, it is reasonable to select a relatively small T_{L}. The choice of the parameter a will be given in Section IV.

Step 3: Compare every $d^{20}(n)$ with T_{L}, if it is greater thanT_{L}, the segment corresponding to $d^{20}(n)$ is considered to contain highly different frames, and thus it is classified as a candidate segment. Meanwhile, the relationship between neighboring distance values are also taken into account. Any segment whose distance value $d^{20}(n)$ satisfies the following condition is classified as a candidate segment as well:

$$\begin{aligned} &(d^{20}(n) > 3d^{20}(n-1) \cup d^{20}(n) > 3d^{20}(n+1)) \\ &\cap\, d^{20}(n) > 0.8\mu_G \end{aligned} \tag{11}$$

Step 4: Perform the first round bisection-based comparison to refine the candidate segments as follows: first, the forward distance ($d_F^{20}(n)$) and backward distance ($d_B^{20}(n)$) values are defined respectively as:

$$d_F^{20}(n) = \sum_x \sum_y |F(x, y; 20n + 10) - F(x, y; 20n)| \tag{12}$$

$$d_B^{20}(n) = \sum_x \sum_y |F(x, y; 20n + 10) - F(x, y; 20n + 20)| \tag{13}$$

In order to determine the location of shot boundary, each candidate segment of length 21 should be further handled according to the relationships among $d_F^{20}(n)$, $d_B^{20}(n)$ and $d^{20}(n)$. Concretely, each candidate segment can be categorized into one of four types as below:

If $(d_F^{20}(n)/d_B^{20}(n) > 1.5 \cap d_F^{20}(n)/d^{20}(n) > 0.7)$, which indicates that the difference between the forward and backward distances is distinct and the forward distance is large enough by comparing with the segment distance. In this case, the shot boundary is considered to be located in the first 11 frames.

Else if $(d_B^{20}(n)/d_F^{20}(n) > 1.5 \cap d_B^{20}(n)/d^{20}(n) > 0.7)$, the shot boundary is considered to be located in the last 11 frames.

Else if $(d_F^{20}(n)/d^{20}(n) < 0.3 \cap d_B^{20}(n)/d^{20}(n) < 0.3)$, no shot boundaries exist in the segment, and this segment is removed from the candidate segment list.

Otherwise, the segment is preserved as a gradual transition may exist in this segment.

Step 5: Perform the second round bisection comparison on all segments of length 11 obtained in Step 4 with the similar operations given in Step 4. The main purpose of this step is to look for cut transitions. Here, candidate segments of length 6 (the segment with 6 frames is the minimal processing unit in our work) can be obtained as suspect CT segments. The rest long successive candidate segments whose length is large than 6 will be merged into longer segments as the candidate GT segments. As a result, detection algorithms for CT and GT can be employed distinguishingly by candidate segments of different lengths.

After the above candidate segment selection operation, a large number of non-boundary frames, about a half of the frames in a video, are eliminated. The rest segments are considered as the potential segments with boundaries, and will be processed in the subsequent steps.

The complexity of SVD is a computationally intractable problem, since the complexity grows rapidly with the increase of the matrix size. However, this problem is greatly solved in our work. As we know, when applying the SVD to a large matrix, the computational complexity is much higher than that for small matrices. After candidate segment selection, a video sequence is segmented into short segments. Thus, the SVD is only operated on these candidate segments, i.e., small matrices, thus the processing time is reduced a lot. In a word, we overcome the drawbacks of SVD while keeping its advantages, the refined vector of a frame is simply used to detect shot boundaries. The experimental results in Section IV will show its superiority in terms of processing speed.

III. Fast Shot Boundary Detection

In this section, the main stage of our scheme is introduced. According to the different lengths of candidate segments, namely a candidate CT segment with 6 frames and a candidate GT segment with more frames, different detecting methods are employed. It should be pointed that, some other changes in a video sequence may be selected into candidate segments as well, but most of them can be distinguished from shot transitions by following detection methods. In practical applications, a candidate CT segment may be a part of GT, meanwhile a candidate GT segment may contain a CT. In our work, we take these two cases into account, which is superior to several existing works. In the proposed scheme, we concentrate on the reduction of detection time. The candidate segment selection and the SVD operation are employed to improve the accuracy and speed of shot boundary detection.

A. Cut Transition Detection

When applying the SVD to each candidate CT segment, namely each 6-frame candidate segment, let N =6 and k =6 in (7), then a set of vectors $\mathbf{\Sigma}_i$ (i =0, ...,N-1) can be obtained. Thus the 1728-dimensional color feature is reduced into a 6-dimensional vector. Here we utilize the similarity value between two consecutive frames as defined in (8) to detect CTs. Let $\Phi(t) = \Phi(f_t, f_{t-1}) = \cos(f_t, f_{t-1})$ denote the cosine distance between Frame f_t and Frame f_{t-1} in a certain candidate CT segment, and let $G = \cos(\mathbf{\Sigma}_0, \mathbf{\Sigma}_5)$ denote an adaptive parameter for each segment. Obviously, G means the distance between the first and last frames in a segment of length 6. A cut transition in the t-th frame is declared if the following two criteria are satisfied:

$$G < 0.95 \tag{14}$$

$$\Phi(t) < p + (1-p)G \tag{15}$$

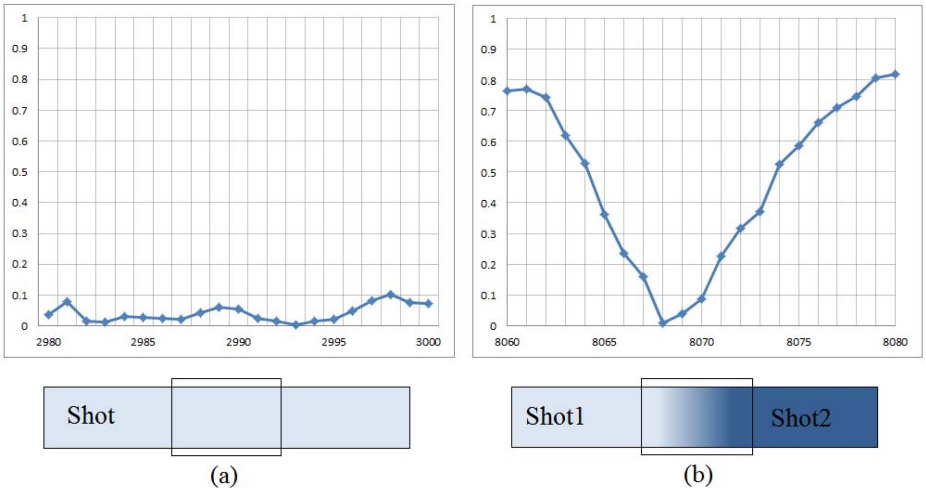
where t =1, ...,N-1 and p is a parameter ranging from 0 to 1.

If the first criterion (14) cannot be satisfied, the detection of this segment is immediately aborted and the segment is discarded. The second criterion (15) is used to identify the gap between consecutive similarity values as an indicator of cut transitions. For the case that no frames in the segment to be detected satisfy the second criterion, GT detection is required, because the similarity between two consecutive frames during a GT is always much higher. Therefore, in this case, if the segment to be detected is isolated, we lengthen it by adding 5 frames before and after this segment respectively. Otherwise, it should be merged into its neighboring segment. For another abnormal case, i.e., more than one frame is identified as cut transitions, the same operation as the above case is implemented.

B. Gradual Transition Detection

In our work, the main contribution lies in that a novel GT detection scheme is proposed. In our scheme, we extract refined feature vectors by SVD and then employ an inverted triangle pattern matching method based on inner characteristics of a gradual transition. As mentioned in Section III.A, a few CTs could not be located accurately into a certain 6-frame segment. Consequently, cut detection is also employed in a GT candidate segment only if the gradual transition detection fails to detect any shot change.

The distance between two consecutive frames does not show distinct characteristics in detecting gradual transitions. However, two frames over the transition segment which belong to different shots exhibit differences definitely. Thus, we extract N+2 frames to form a frame-feature matrix for a candidate GT segment of length N by adding one frame before and after the segment respectively. For all the candidate GT segments, N is at least equal to 11, typically 21, 41 or more. When we employ the SVD on each 1728×(N+2) matrix, since k must be not large than the rank of the matrix $\boldsymbol{A}$, namely



(a) (b)

Fig. 1. The curve $\boldsymbol{d}$ $(\boldsymbol{t})$ for different segments: (a) a segment with moving objects; (b) a segment with a gradual transition.

N+2 at most, thus setting k=10 is suitable for GT detection (The same choice is made in [23]). Based on our further experiments, increasing the parameter k for long segments brings more detail information and noises to the refined feature vectors, making the detection more sensitive to tiny changes and noises.

After applying SVD, a set of vectors $\{\mathbf{\Sigma}_{\mathrm{s}}$, $\mathbf{\Sigma}_0$, ...,$\mathbf{\Sigma}_{N-1}$, $\mathbf{\Sigma}_{\mathrm{e}}\}$can be obtained, where $\mathbf{\Sigma}_i$ (i=0, ..., N-1) is for Frame f_i (i=0, 1, ... , N-1) in the candidate segment while $\mathbf{\Sigma}_{\mathrm{s}}$ and $\mathbf{\Sigma}_{\mathrm{e}}$ are for the added frames f_{s} and f_{e} before and after the segment, respectively. In general, frames within a shot are similar and static. Here f_{s} stands for the last frame of the previous shot while f_{e} stands for the first frame of the next shot. An ideal gradual transition from one shot to the next is a series of consequent mild changes frame by frame. As a result, within a gradual transition, frames close to f_{s} are visually similar to the previous shot. Similarly, frames close to f_{e} are visually similar to the next shot. Meanwhile, for the ideal middle frame f_{m} within a gradual transition, the feature distance between f_{s} and f_{m} is generally equal to that between f_{m} and f_{e}. For the t-th frame f_t (t=0, ..., N-1) in the candidate segment, let $\Phi(f_{\mathrm{s}}, f_t)$ and $\Phi(f_t, f_{\mathrm{e}})$ respectively denote the similarity between f_{s} and f_t and the similarity between f_t and f_{e}, we define the *absolute distance difference* $d(t)$ as follows:

$$d(t) = |\Phi(f_s, f_t) - \Phi(f_t, f_e)| \tag{16}$$

The purpose of above definition is to track the change in similarity difference for each frame in the candidate segment. According to (5), $d(t)$ can be rewritten as

$$d(t) = |\frac{(\boldsymbol{\beta}_s, \boldsymbol{\beta}_t)}{||\boldsymbol{\beta}_s|| \cdot ||\boldsymbol{\beta}_t||} - \frac{(\boldsymbol{\beta}_t, \boldsymbol{\beta}_e)}{||\boldsymbol{\beta}_t|| \cdot ||\boldsymbol{\beta}_e||}| \tag{17}$$

Obviously, $d(t)$ takes value in the interval [0, 1]. In the case that f_{s} and f_{e} are completely different, for t=0 or N-1, $d(t)$ is extremely close to 1, while for t = (N+1)/2 or N/2 (i.e., the middle frame of a GT), $d(t)$ is extremely close to 0. If calculating $d(t)$ from t=0 to N-1 for a gradual transition, we find that $d(t)$ approximately linearly decreases with t in the first half of the candidate segment while linearly increases with t in the second half. Consequently, the curve of $d(t)$ exhibits the shape of an obvious inverted isosceles triangle. Since color histograms are used as features, for other non-transition changes such as object movement and global camera zooming in/out, their $d(t)$ will not exhibit the inverted triangle pattern as shown in Fig. 1, where two segments from a documentary of NASA 25th anniversary are used.

In [19] and [20], the triangle pattern is used for detect gradual transitions but the iteration algorithm is needed to determine the length of GT. In [26], Petersohn adopted a U-shape pattern as the first stage for detecting dissolve shot boundaries. Petersohn utilized the edge energy of MPEG DC-coefficients as the metric in his first stage algorithm, and selected dissolve candidates by searching for U-shapes diagrams. And the diagram matching is used for determining the approximate lengths and positions of dissolves. His work contributes to the pattern matching methods, but our inverted triangle pattern is different from his. In addition, we are able to take advantage of the features of our proposed pattern to detect gradual transitions accurately and quickly.

From Fig. 1, we can see that the absolute distance differences $d(t)$ obtained from different segments have a great discrepancy, which can be used to distinguish gradual transitions from non-gradual transitions. Based on above phenomena, an inverted-triangle-pattern matching method is proposed in our work. If the curve of $d(t)$ for a candidate segment fits the inverted triangle pattern, then this segment can be declared as a gradual transition. Thus, the computational complexity of GT can be greatly reduced. There are two reasons: first, the position and length of each potential GT are predicted by candidate segment selection. Second, GT detection is only performed on candidate segments, which are a small fraction of video sequence. The criteria of pattern matching in an N-frame segment are given as follows:

i) As the most important characteristic of a GT, the difference between the maximum and minimum values of $d(t)$ should be distinct, i.e.,

$$\max(d(t)) - \min(\mathrm{d(t)}) > T_1 \tag{18}$$

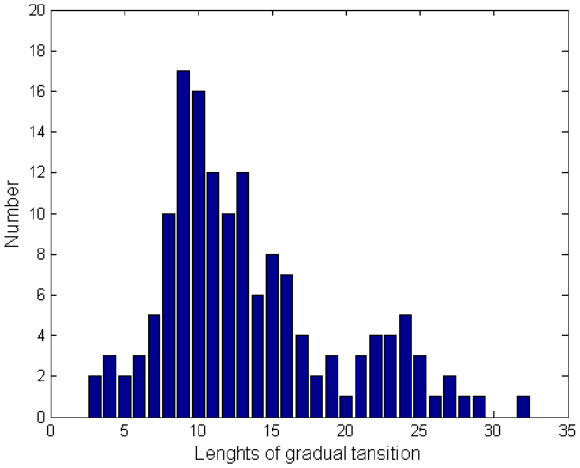


Fig. 2. Histogram of gradual transition lengths.

ii) Approximate symmetry is another character of $d(t)$, i.e., the point $t_{\rm m}$with the minimum value $d_{\rm min}$ should be the center point of the curve, and the offset of its location should be no more than a given threshold T_2:

$$|(t_m - (N+1)/2)/N| \le T_2 \tag{19}$$

iii) The number of abnormal points in the curve should be limited, namely,

$$(K_{L\_ab} + K_{R\_ab})/N \le T_3 \tag{20}$$

where $K_{\rm L\_ab}$ is the number of ascending points ($d(t)$>$d(t$-1)) in the first part of segment before the minimum point $t_{\rm m}$, and $K_{\rm R\_ab}$ is the number of descending points ($d(t)$<$d(t$-1)) after $t_{\rm m}$.

The length of candidate segment N has a great impact on the performance of pattern matching. To prove this point, we observe the statistical property of gradual transition lengths from five video sequences—one news, one sport game, three documentaries from the TRECVID 2001 test data—which have 148 shots with gradual transitions in total. The shot boundaries of these transitions are obtained manually. The histogram of gradual transition lengths is shown in Fig. 2. We can see that most gradual transition lengths are around 10 frames while seldom ones exceed 30 frames. As a result, in our work, we assume that the length of gradual transition is generally smaller than 31 frames. For a candidate segment which satisfies (18), if it is longer than 31 frames at same time, we should normalize it into 31 frames by taking 15 frames on each side of the minimum point of $d(t)$. Then the detection can restart.

Besides the length adjustment, for GT detection, there are two situations where the position of a candidate segment should be adjusted. In these two situations, the inverted triangle is biased, thus we should adjust the position of the candidate segment in order to avoid a false negative. Concretely, if the criterion (18) is satisfied while (19) is not, assume the bias between t_m and (N+1)/2 is L, then the corresponding GT candidate segment should be adjusted by L frames, i.e., the position of the segment should be moved L frames backward or forward while keeping its length unchanged. Two examples for these situations are shown in Fig. 3. As shown in Fig. 3 (a), the minimum point is on the right side of the middle point, indicating that the candidate segment should be moved forward. On the contrary, in Fig. 3 (b), the candidate segment should be moved backward. In view of the computational complexity and stability of performance, the adjustment procedure is applied at most two times, and the movement of a candidate segment is limited within 10 frames every time.

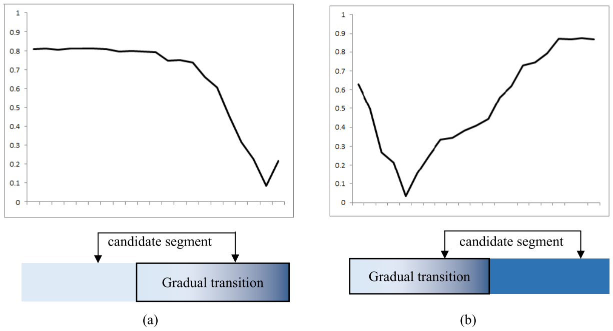


Fig. 3. Two situations where candidate segments should be adjusted. (a) the candidate segment should be moved forward; (b) the candidate segment should be moved backward.

Based on the discussion above, our GT detection method can be summarized as follows:

Step 1: For the current N-frame candidate GT segment, extract N+2 frames with 2 frames added before and after the segment, and calculate the HSV histogram with 1728 bins (18 bins for H, 12 bins for S, and 8 bins for V) for each frame. Employing the SVD on the 1728×(N+2) frame-feature matrix, a set of feature vectors $\{\mathbf{\Sigma}_{\rm s}, \mathbf{\Sigma}_0, \ldots, \mathbf{\Sigma}_{N-1}, \mathbf{\Sigma}_{\rm e}\}$ with reduced dimensions are obtained.

Step 2: Calculate the adaptive parameter $G = \cos(\mathbf{\Sigma}_0, \mathbf{\Sigma}_{N-1})$, and go to the next step if the following condition is satisfied:

$$G < 0.9 \tag{21}$$

Otherwise this segment is discarded. Here we use a strict threshold compared with (14) because a candidate GT segment is generally longer than CT, thus more differences should be considered.

Step 3: Calculate the absolute distance difference $d(t)$ (t=0,… ,N-1) for every frame in each candidate segment. If the criterion (18) is satisfied, for the candidate segment not longer than 31 frames, directly go to Step 4, while for the candidate segment longer than 31 frames, the normalized 31-frame candidate segment will be redetected from Step 1. Otherwise, go to Step 1 for the next candidate segment.

Step 4: Check whether the candidate segment satisfies the criterion (19) or not. If it is not satisfied, we perform the aforementioned segment position adjustment at most twice (Note that for each time, Steps 1-4 should be re-performed). If the criterion (19) is still not satisfied, go to Step 5. Next, the criterion (20) is checked. If it is satisfied, go to Step 6. Otherwise, go to Step 5.

Step 5: Employ our CT detection scheme on the segment where no GT is declared. If the cut transition is detected normally, go to Step 1 for next candidate segment.

TABLE I
VIDEO SEQUENCES USED AND THEIR RESPECTIVE DESCRIPTIONS

Video	Frames	Transitions			Sources
		Total	Cut	Gradual	
D1	11356	65	38	27	TRECVID 2001 test data (NASA 25th Anniversary,Airline Safety and Economy, Perseus Global Watcher)
D2	16586	73	42	31	
D3	12304	103	39	64	
D4	31389	153	98	55	
D5	12508	71	45	26	
D6	13648	85	40	45	
News1	6662	35	15	20	NBC News
News2	12925	55	23	32	
Soccer	22137	69	43	26	UEFA Champions League
Movie	10894	114	114	0	Chinese movie “Lost on Journey”
Sitcom	6300	72	72	0	“The Big Bang Theory”
Total	**156709**	**895**	**569**	**326**	

TABLE II
COMPARISONS OF ADAPTIVE THRESHOLDS IN CANDIDATE SEGMENT SELECTION

Video	Total shot boundaries	Shot boundaries still involved after Candidate Segment Selection	
		Using the threshold of the fast framework [21]	Using the threshold of our proposed scheme
D1	65	63	65
D4	153	142	152
D5	71	66	71
Soccer	69	55	66

Step 6: Declare the candidate segment as a GT in terms of the first and the last frames. Then go to Step 1 for next candidate GT segment.

IV. EXPERIMENTAL RESULTS

To evaluate the performance, we have tested the proposed shot boundary detection scheme by using a variety of test videos, as listed in TABLE I. The 6 documentaries (D1 to D6 in TABLE I) are got from the TRECVID 2001 test data released on the “Open-Video Project” [28], while the news video is from the NBC “Nightly News”. All these video sequences are reformed into the uncompressed AVI format with a resolution of 320×240 pixels, with a length of 156709 frames, and a duration of 90 minutes in total. There are 895 shot transitions in total. Among them, 326 gradual transitions include all the common types—dissolve, fade in/out, and wipe.

The efficiency of the proposed algorithm is compared with the fast framework proposed in [21] and the method in [27] to demonstrate the superiority of our scheme in speed and accuracy. Similar to other shot boundary schemes, the performance is evaluated by recall, precision and F_1 metrics that are defined as follows:

$$\text{recall} = \frac{N_C}{N_C + N_M} \tag{22}$$

$$\text{precision} = \frac{N_C}{N_C + N_F} \tag{23}$$

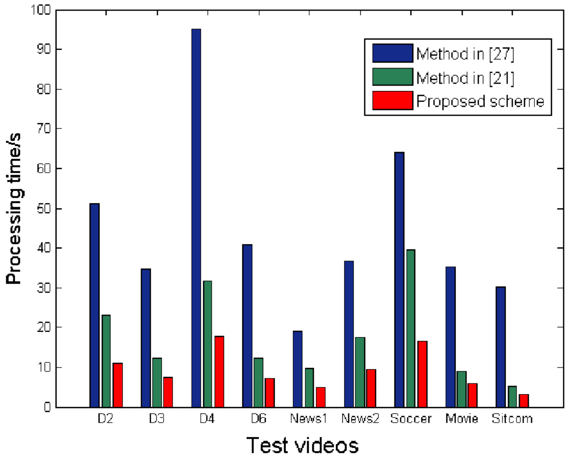


Fig. 4. Comparison of processing time on various test video sequences between the proposed scheme and other two shot boundary detection methods.

TABLE III
COMPARISON OF TOTAL TIME AND RESPECTIVE TIME FOR CT/GT DETECTION

Video	Time saved (%)	The fast framework [21]			Proposed scheme		
		Total	CT det.	GT det.	Total	CT det.	GT det.
D2	52.2	23.2s	0.2s	19.6s	11.1s	0.7s	8.4s
D3	39.0	12.3s	0.2s	10.3s	7.5s	0.7s	5.5s
D4	43.7	31.6s	0.4s	27.4s	17.8s	1.8s	12.2s
D6	42.3	12.3s	0.3s	10.3s	7.1s	0.8s	4.8s
News1	49.0	9.6s	0.1s	8.5s	4.9s	0.3s	3.6s
News2	46.6	17.6s	0.2s	15.5s	9.4s	0.7s	6.5s
Soccer	58.1	39.6s	0.2s	32.3s	16.6s	0.5s	13.0s
Movie	34.4	9.0s	0.5s	6.6s	5.9s	1.8s	2.7s
Sitcom	41.5	5.3s	0.4s	3.7s	3.1s	1.2s	0.9s

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{24}$$

where N_C is the number of shot boundaries that are detected correctly, N_M is the number of shot boundaries that are missed, N_F is the number of shot boundaries that are detected falsely, and F_1 is a measure considering both recall and precision.

A. Parameter Selection

To train and select parameters, video sequences D1 and D5 from the database TRECVID 2001 are adopted as the training set. In our scheme, the parameter a in (10) has great effect on the candidate segment selection, resulting in a global influence on detection performance including speed and accuracy. Since CT detection and GT detection are relatively independent, there is only slight relationship between the parameter in the criterion (15) and the parameters in the criteria (18)–(20), thus they can be determined separately.

First, we focus on CT detection to estimate the approximate ranges for parameters a and p. It can be seen that the recall and precision for CT detection are influenced by p directly. On the other hand, a has an indirect effect on the recall, since a stricter threshold T_L will lead to more miss selections of candidate segments in both CT and GT, while a looser one will produce more non-boundary segments which

TABLE IV

COMPARISONS OF TIME, RECALL, PRECISION AND F1 BETWEEN THE PROPOSED SCHEME AND THE METHOD IN [27]

Video	The method in [27]				Proposed scheme			
	Time	Recall	Pre	F1	Time	Recall	Pre	F1
D2	51.1s	0.881	0.771	0.822	11.1s	0.905	0.905	0.905
D3	34.6s	0.897	0.673	0.769	7.5s	0.667	0.867	0.754
D4	95.1s	0.867	0.876	0.871	17.8s	0.888	0.897	0.892
D6	40.7s	0.950	0.883	0.916	7.1s	0.950	0.974	0.962
News1	19.1s	0.933	0.583	0.718	4.9s	0.867	0.867	0.867
News2	36.8s	1.000	0.697	0.821	9.4s	1.000	1.000	1.000
Soccer	64.1s	0.953	0.683	0.796	16.6s	0.767	1.000	0.868
Movie	35.3s	0.956	0.879	0.916	5.9s	0.886	0.953	0.918
Sitcom	30.3s	0.972	0.921	0.946	3.1s	0.986	1.000	0.993

may increase the false positive in detection. The preliminary conclusion is that $a$takes value in the interval [0.5, 0.7] and p in [0.4, 0.6], based on which the CT detection results are acceptable.

Then we determine the parameters for GT detection. Since T_2 is a parameter for adjusting the position of suspect gradual transitions, we set it to be 0.25 by considering the computational complexity. It is found that a conservative threshold for candidate segment selection leads to low detection speed especially in a video sequence with many GTs, thus a is set to 0.7 in our work. The rest parameters are set by maximizing the performance while considering the trade-off between speed and accuracy. Finally, in our experiments, the parameters are set as follows: $a=$ 0.7, $p=$ 0.48, T_1=0.33, T_3=0.3.

To show the superiority of our adaptive threshold in candidate segment selection, comparisons between two adaptive methods are given in TABLE II. We choose training videos D1, D5, and the longest two test videos D4, Soccer, and count their shot boundaries before and after candidate segment selection. It can be seen that our adaptive threshold is better for containing more shot boundaries after candidate segment selection. The result here has a significant influence on the recall, because once a shot boundary is eliminated in the candidate segment selection step, it can hardly be found in the following steps. Thus a moderate threshold is necessary.

B. Comparison of Detection Speed

In the fast framework [21], the detection speed has a great improvement compared with the LTD (Linear Transition Detection) algorithm [20], where the speedup ratio varies from 21 to 70. Thus, here, we only compare the processing time of the proposed scheme with that of the fast framework proposed in [21] by applying them to each test video. As shown in Fig. 4, the total processing time of our scheme is far less than that of the fast framework. Therefore, we can draw a conclusion that our scheme is about 40 to 140 times faster than the LTD algorithm. Their respective time for CT and GT detection are compared in TABLE III.

In [27], Lu *et al.* employ near-duplicate video shots detection by selecting adaptive reference frames for shot representation. Since this approach has a structure different from ours, the overall comparison is shown in TABLE IV. In this approach, Pearson’s Correlation Coefficient (PCC) is calculated between adjacent frames, it is a time-consuming operation which generally exists in many other shot detection methods. It can be seen that our approach has great superior performance in processing time.

In our scheme, by employing candidate segment selection, most non-boundary frames are discarded. Detection speed is thus accelerated a lot because the detection is performed on only a few frames. It can be seen that the CT detection is very fast for both our scheme and the scheme in [21], however our scheme saves much time on GT detection. For those long video sequences with many gradual transitions, e.g. D1, D2, D4, and Soccer, the speed of our scheme on GT detection is more than two times that of fast framework. The main reason is that, in our scheme, the video feature is extracted only once when constructing the “frame-feature” matrix in the SVD transform, while many other GT detection schemes need to extract video features repetitively during the sliding of the processing window. On the other hand, the complexity of the SVD on huge matrices is avoided by candidate segment selection. Thus, the overall computational complexity can be reduced by our novel scheme.

C. Comparison of Detection Accuracy

To demonstrate the accuracy of our scheme, the recall, precision and F1 measures for all the test videos are also compared between the proposed scheme and the fast framework in [21] respectively, as listed in TABLE V. And the comparison with the method in [27] is presented in Table IV. As aforementioned, the parameters of our scheme are set by considering the compromise between speed and accuracy, thus the precision of both CT detection and GT detection is quite good while keeping a satisfactory recall. One reason for the low recall of the fast framework [21] is that it misses some cut transitions in GT candidate segments and some gradual transitions related to several short candidate segments. However, our scheme is able to detect these two kinds of transitions, to reduce the rate of miss detection. Furthermore, the adjustment of lengths and positions of gradual transitions can ensure the high precision rate. A concrete example is given

TABLE V

COMPARISONS OF RECALL, PRECISION AND F1 BETWEEN THE PROPOSED SCHEME AND THE FAST FRAMEWORK [21]

Video	The fast framework [21]						Proposed scheme					
	Cut Transition			Gradual Transition			Cut Transition			Gradual Transition		
	Recall	Pre	F1	Recall	Pre	F1	Recall	Pre	F1	Recall	Pre	F1
D2	0.571	1.000	0.727	0.484	0.833	0.612	0.905	0.905	0.905	0.935	0.725	0.817
D3	0.462	1.000	0.632	0.313	0.870	0.460	0.667	0.867	0.754	0.734	0.940	0.824
D4	0.755	0.987	0.856	0.436	0.649	0.522	0.888	0.897	0.892	0.727	0.741	0.734
D6	0.899	1.000	0.947	0.511	0.885	0.648	0.950	0.974	0.962	0.844	0.927	0.884
News1	0.733	1.000	0.846	0.250	0.500	0.333	0.867	0.867	0.867	0.800	0.889	0.842
News2	0.864	0.950	0.905	0.375	0.571	0.453	1.000	1.000	1.000	0.719	0.958	0.821
Soccer	0.209	1.000	0.346	0.385	0.370	0.377	0.767	1.000	0.868	0.923	0.727	0.813
Movie	0.807	0.948	0.872	----	----	----	0.886	0.953	0.918	----	----	----
Sitcom	0.875	1.000	0.933	----	----	----	0.986	1.000	0.993	----	----	----

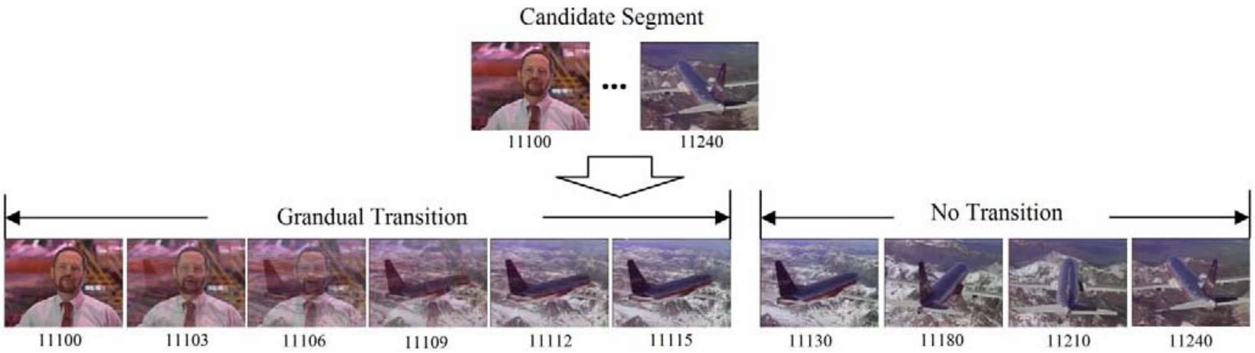


Fig. 5. An example result of gradual transition detection by our method.

in Fig. 5, where the segment from the 11100-th frame to the 11240-th frame in the test video D5 is selected as a candidate segment by our scheme. This segment contains a GT and a fragment of object movement lasting about 100 frames. It is falsely rejected by the fast framework proposed in [21], but it is detected successfully by our method since our method can remove the object movement fragment from the candidate segment. There are many examples like this in our comparison experiments. In fact, our precise result in GT detection is attributed to our position and length adjustment procedures.

V. CONCLUSION

Real-time applications on video content management are demanded urgently, motivated by this, we present a fast shot boundary detection scheme in this paper. By employing candidate segment selection, most non-boundary frames are eliminated before CT and GT detection. Singular value decomposition is used to reduce the dimension of feature. These two aspects help us obtain a high detection speed. Cut transition detection and gradual transition detection are employed on segments with different lengths distinguishingly, decreasing the rate of miss detections. For gradual transition detection, a novel unified algorithm is presented by adopting the basic property of gradual transition—approximate linear changes of frame features within two consecutive shots. We observe that the change of absolute distance difference $d(t)$ with the time t exhibits the shape of an inverted triangle, and thus a pattern matching method is proposed to detect gradual transitions. Different types of gradual transitions—dissolves, fade in/out, wipes—can be detected with good performance. Experimental results show that the detection speed of our scheme is very fast and outperforms the state-of-the-art fast detection algorithm. Furthermore, owing to SVD, the feature extraction is employed only once when the frame-feature matrix is constructed. Thus, our scheme can accelerate the detection speed more obviously on video sequences with higher resolution, and perform well in real-time systems with practical video materials.

REFERENCES

[1] N. Dimitrova, Z. Hong-Jiang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *IEEE Multimedia*, vol. 9, no. 3, pp. 42–55, Jul./Sep. 2002.

[2] L. A. Rowe and R. Jain, "ACM SIGMM retreat report on future directions in multimedia research," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 1, no. 1, pp. 3–13, Feb. 2005.

[3] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation. A review," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 28–37, Mar. 2006.

[4] J. H. Yuan, H. Y. Wang, L. Xiao, W. J. Zheng, J. M. Li, F. Z. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 168–186, Feb. 2007.

[5] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 90–105, Feb. 2002.

[6] C. W. Su, H. Y. M. Liao, H. R. Tyan, K. C. Fan, and L. H. Chen, "A motion-tolerant dissolve detection algorithm," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1106–1113, Dec. 2005.

[7] C. Cai, K. M. Lam, and Z. Tan, "A unified shot boundary detection method based on linear prediction with Bayesian cost functions," in *Proc. IEEE Int. Workshop VLSI Design Video Technol.*, May 2005, pp. 101–104.

[8] N. V. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognit.*, vol. 30, no. 4, pp. 583–592, Apr. 1997.

[9] Y. P. Tan, J. Nagamani, and H. Lu, "Modified Kolmogorov-Smirnov metric for shot boundary detection," *Electron. Lett.*, vol. 39, no. 18, pp. 1313–1315, Sep. 2003.

[10] A. Dailianas, R. B. Allen, and P. England, "Comparison of automatic video segmentation algorithms," *Proc. SPIE*, vol. 2615, pp. 2–16, Jan. 1996.

[11] B. L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 6, pp. 533–544, Dec. 1995.

[12] D. Lelescu and D. Schonfeld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 106–117, Mar. 2003.

[13] W. Zheng, J. Yuan, H. Wang, F. Lin, and B. Zhang, "A novel shot boundary detection framework," *Proc. SPIE*, vol. 5960, pp. 410–420, Jul. 2005.

[14] L. Yang, H. Lu, B. Wang, X. Xue, and Y. P. Tan, "Shot boundary classification and refinement using inter-frame similarity patterns," in *Proc. 5th Int. Conf. Inf., Commun. Signal Process.*, May 2005, pp. 673–677.

[15] A. Mittal, L. F. Cheong, and L. T. Sing, "Robust identification of gradual shot-transition types," in *Proc. Int. Conf. Image Process.*, vol. 2. 2002, pp. II-413–II-416.

[16] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 82–91, Jan. 2006.

[17] J. U. Won, Y. S. Chung, I. S. Kim, J. G. Choi, and K. H. Park, "Correlation based video-dissolve detection," in *Proc. Int. Conf. Inf. Technol., Res. Educ.*, Aug. 2003, pp. 104–107.

[18] D. Adjeroh, M. C. Lee, N. Banda, and U. Kandaswamy, "Adaptive edge-oriented shot boundary detection," *EURASIP J. Image Video Process.*, vol. 2009, pp. 1–13, Jun. 2009.

[19] J. Bescos, G. Cisneros, J. M. Martinez, J. M. Menendez, and J. Cabrera, "A unified model for techniques on video-shot transition detection," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 293–307, Apr. 2005.

[20] C. Grana and R. Cucchiara, "Linear transition detection as a unified shot detection approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 483–489, Apr. 2007.

[21] Y. N. Li, Z. M. Lu, and X. M. Niu, "Fast video shot boundary detection framework employing pre-processing techniques," *IET Image Process.*, vol. 3, no. 3, pp. 121–134, Jun. 2009.

[22] Y. H. Gong and X. Liu, "Video shot segmentation and classification," in *Proc. 15th Int. Conf. Pattern Recognit.*, vol. 1. Sep. 2000, pp. 860–863.

[23] Z. Cernekova, C. Kotropoulos, and I. Pitas, "Video shot-boundary detection using singular-value decomposition and statistical tests," *J. Electron. Imaging*, vol. 16, no. 4, pp. 043012-1–043012-13, 2007.

[24] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 1–13, Feb. 2000.

[25] H. Lu and Y. P. Tan, "An effective post-refinement method for shot boundary detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1407–1421, Nov. 2005.

[26] C. Petersohn, "Dissolve shot boundary determination," in *Proc. IEE Eur. Workshop Integr. Knowl., Semantics Digit. Media Technol.*, 2004, pp. 87–94.

[27] S. Y. Lu, Z. Y. Wang, M. Wang, M. Ott, and D. Feng, "Adaptive reference frame selection for near-duplicate video shot detection," in *Proc. IEEE 17th Int. Conf. Image Process.*, Sep. 2010, pp. 2341–2344.

[28] (2001). *TREC Video Retrieval Test Collection* [Online]. Available: http://www.open-video.org/collection_detail.php?cid=7

Zhe-Ming Lu (M'03–SM'06) was born in Zhejiang, China, in 1974. He received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in measurement technology and instrumentation from the Harbin Institute of Technology (HIT), Harbin, China, in 1995, 1997, and 2001, respectively.

He became a Lecturer with HIT in 1999. Since 2003, he has been a Professor with the Department of Automatic Test and Control, HIT. He is currently a Full Professor with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. He has published more than 250 papers, seven monographs in Chinese, one monograph in English, and three book chapters in English. His current research interests include multimedia signal processing, information security, and complex networks.

Dr. Lu received the second prize of National Defense Science and Technology of China in 2001, the first prize of the Ministry of Education of China and the Award of Youth Science and Technology of Harbin City in 2002. He was the Chairman of the Youth Science and Technology Association of HIT from 2003 to 2006. He organized and chaired the invited session titled "Image Compression and Digital Watermarking Techniques" in the 7th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA, in 2003. He received the 100 Most Excellent Doctors in China Award for authoring more than 40 papers in the field of vector quantization in 2003. He won the second prize of Heilongjiang Province Science and Technology and the first prize of Heilongjiang Colleges and Universities Science and Technology in 2004. He was elected as the New Century Excellent Talents in Universities of China in 2004. One of his papers was awarded the Best Paper Award in KES in 2005 and one was awarded the Best Paper Award in ICICIC in 2006. He won the second prize of Colleges and Universities Science and Technology of Ministry of Education of China in 2006. He was recognized by the Natural Science Foundation for Outstanding Youths of Zhejiang Province in 2011. He has been a member of IEICE since 2012. He was awarded the third prize of Zhejiang Province Science and Technology in 2012 and the first prize of Zhejiang Colleges and Universities Science and Technology in 2013.

Yong Shi was born in Shaanxi, China, in 1988. He received the B.S. degree in information and communication engineering and the M.S. degree in aerospace engineering from Zhejiang University, Hangzhou, China, in 2010 and 2013, respectively. His current research interests include image processing and video analysis.