Multimed Tools Appl (2014) 71:1749–1770
DOI 10.1007/s11042-012-1301-z

To accelerate shot boundary detection by reducing detection region and scope

Guangyu Gao · Huadong Ma

Published online: 21 December 2012

Abstract Video Shot Boundary Detection (SBD) is the fundamental process towards video summarization and retrieval. A fast and efficient SBD algorithm is necessary for real-time video processing applications. Extensive work has focused on accurate shot boundary detection at the expense of demanding computational costs. In this paper, we propose a fast SBD approach that reduces the computation pixel-wise and frame-wise while still giving satisfactory accuracy. The proposed approach substantially speeds up the computation through reducing both detection region and scope. Color histogram and mutual information are used together to measure the difference between frames. Corner distribution of frames is utilized to exclude most of false boundaries. We conduct extensive experiments to evaluate the proposed approach, and the results show that our approach can not only speed up SBD, but also detect shot boundaries with high accuracy in both Cut (CUT) and Gradual Transition (GT) boundaries.

Keywords Shot boundary detection · Skipping interval · Mutual information · Camera motion · Corner distribution

1 Introduction

Video processing and analysis has become more and more important since the video resources emerge quickly with the increased availability of video cameras. Video segmentation is one of the fundamental step in video processing and analysis. Generally, videos are segmented using Shot Boundary Detection (SBD) as the basic approach. Here, *shot* is defined as a continuous sequence of frames captured using

G. Gao · H. Ma (✉)
Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing University of Posts and Telecommunications, Beijing 100876, China
e-mail: mhd@bupt.edu.cn

one camera [2]. Usually, all the frames in a shot have consistent visual characteristics, such as color, texture and motion. The transitions between two shots can be abrupt or gradual, and these transitions can be classified into CUT transitions or gradual transitions (GT) respectively [4]. CUT transition is an instantaneous transition from one shot to the next. There are no transitional frame between two shots. The gradual transition is formed by the editor to insert an effect of a fade, a wipe, and a dissolve so that the frames change slowly with a span of several frames.

Over the last decade, there have been many approaches proposed for shot boundary detection, such as [1, 11, 24, 26, 30, 32]. However, most of these methods mainly aimed at implementing SBD accurately. In this paper, we propose an approach to implement SBD efficiently while maintaining the accuracy. The computational complexity mainly depends on the number of frames being processed and the frame resolution. The proposed approach is computationally efficient as we intelligently process only few frames and few pixels per frame. In comparison with CUTs, GTs are more difficult to be detected, due to the complex editing effects as well as camera and/or object motions within a shot. Thus, by skipping frames, the hard GT detection can be transformed into the easy cut detection, and as a result, almost all GTs can be found. The main contributions of this paper are as follows.

1. We accelerate the SBD process in spatial domain in two aspects: one, by processing only the pixels in *Focus Regions (FR)*. Two, mutual information and color histogram are computed together since the mutual information is more efficient for SBD detection and the color histogram is easier to be calculated.
2. We accelerate the SBD process in temporal domain by skipping frames adaptively. Instead of degrading the accuracy, almost all boundaries could be detected including GTs which are hard to be detected.
3. The corner distribution of frames near candidate shot boundaries is adopted to remove most of the false boundaries and to find the precise interval of the true boundary.

The remainder of this paper is organized as follows. We review some related work in Section 2. In Section 3.1, we present details of the processed pixels reduction in spatial domain, and processes on how to reduce computation temporally by skipping processing of redundant frames will be discussed in Section 3.2. In Section 4, we will describe how to remove the false boundaries. Experimental results will be shown in Section 5. Finally, We present our conclusions in Section 6.

2 Related work

Recently, many SBD methods have been proposed, such as the methods based on pixels [24, 30], image edges [11], motion vectors [13, 15] and color histograms [9, 20]. Authors of [4] presented the comparison among existing methods. Pixel-based method is sensitive to luminance, camera motion or object motion, and methods based on grayscale or color histogram will be ineffective when frames in two different shots have resembling histogram. In addition, these methods also can not handle the frame difference caused by large camera motion or abrupt change in illuminance [4]. Therefore, many people made use of more complex features, such as image blocks and motion vectors [11, 13], to improve detection accuracy. But these

more sophisticated features still cannot completely solve these problems [10], and just help to alleviate the situation. Besides, Yuan et al. [31] proposed an unified SBD system based on graph partition model to do formal study of SBD. Huang et al. [14] implemented SBD via local key-point matching, and the method based on information theory was also discussed in [28, 33]. However, as described in [10], these methods alleviated the situation to some extent, rather than solved the problems caused by camera motion and change of illumination completely.

Although some of the above mentioned methods can deal with CUT detection well, they were less than satisfactory and may be prone to error when handling GT. Therefore, many researchers focused on GT detection, especially the specific type of GT, when the video is always pre-segmented by CUT detection and other different types of GT detection. For example, Wu et al. [27] proposed to find horizontal and vertical wipes by computing pixel-wise DC coefficient differences between consecutive *I* and *P* frames. The dissolves and wipes were also detected by Pei and Chou [21, 22] who employed the macroblocks of *P* and *B* frames in MPEG videos. Lienhart [18] treated video as series of frames and analysed them with a multiresolution method. Then, pattern classification techniques were used to train a model for dissolve detection. Recently, Su et al. [25] utilized the monotonicity of intensity changes during transitions to detect dissolves, and their algorithm can tolerate fast motions.

In addition, several robust SBD methods were proposed to deal with both CUT and GT well in recent years. In [26], an enhanced graph partition method was used to detect shot boundaries based on non-linear scale space filtering for reducing computation. Zhu et al. [32] also used a coarse-to-fine algorithm to detect shot boundary in view of reducing the effect of noise and motion. Furthermore, such a two-step method can efficiently avoid a number of false alarms and speed up computation process drastically. Meanwhile, Adjeroh et al. [1] proposed an adaptive edge-oriented framework which is distinct in its use of multiple multilevel features in the required processing, and a five fold efficiency improvement in shot characterization and classification.

While some of the afore-mentioned methods are considered to save processing time, it implies that the computational complexity is another important factor to measure the SBD performance, especially in real-time applications. Actually, some work has addressed this problem of SBD in the compressed domain, while the absence of the decoding process allows for a much faster algorithm. The features extracted from the compressed videos are those directly available from the MPEG streams, including discrete cosine transform (DCT) coefficients, motion vectors and predicted directions for each block. An instance is the work of Zhao et al. [7]. They proposed a detection system that realized a fast, effective and tractable SBD. The MPEG decoder and feature vector generation module were first applied to extract the features which, then, were put into diverse detectors of CUT detector, FOI detector, GT detector, and motion detector separately.

However, compressed domain-based approaches are highly dependent on the compression standards, and have the drawback of low reliability and accuracy, especially in the presence of high motion [12]. In summary, there are also several very efficient approaches using uncompressed videos. For example, Lefévre et al. [16] gave the computational complexity in different SBD methods with the real-time situation. Although some methods were proposed as fast SBD methods, such as [19],

they did not really accelerate the processing. Generally, the adjacent frames in GT are so similar, but frames with a temporal distance will be very different. Accordingly, the authors of [6, 17] presented fast SBD methods by skipping several frames. For instance, Danisman et al. in [6] set a skipping interval (named TSFI) as 1 and 5 frames. This method can not efficiently improve the efficiency, since if the GTs last more than 5 frames, they can not be found accurately with such small skipping interval. Besides, Li [17] used the bisection-based comparisons to eliminate nonboundary frames. However they employed the pixel-wise distance of the luminance component to compare two frames, which is sensitive to color, flush and so on. Furthermore, they used 20 frames as the skipping interval for distance calculation which reduces the processed frames in a limited scope, and also the fixed interval of 20 frames is not so reasonable.

In our approach, we pursue not only high efficiency, but also high accuracy for finding all the boundaries. In order to achieve this goal, we speed up the SBD process using the defined focus region to reduce the spatially redundant pixels, as well as boldly skip self-adaptive interval frames to reduce the temporal redundant frames. Besides, the color histogram is also taken into account in combination with mutual information for measurement of difference between frames. Now, by skipping frames, we can actually detect almost all shot boundaries, including CUTs and GTs of fade, dissolves and wipes, as well as several camera and object motion caused false boundaries. With the frames' corner distribution analysis, the introduced false boundaries can be removed.

3 Fast SBD by reducing redundant pixels and frames

3.1 Spatially redundant pixels reduction

A video has thousands of frames, and each frame has thousands of pixels. These vast frames and pixels make the computation complexity very high, which is the reason that many SBD methods or systems are of low efficiency. Although spatial sub-sampling of frames has been suggested to improve video processing efficiency in [3, 29], it still depends on the choice of the spatial window. Smaller window size is sensitive to object and camera motions, while arbitrary window size could not make the remaining pixels represent the frame well. For example, authors of [3] only used the histogram difference of upper halves in two successive frames, which generated many false boundaries because of the loss of important information in down halves.

Generally, the most essential information in a frame is always concentrated around the center of a frame, and the more the pixels are close to the frame center, the more important the pixels are. In order to reduce the processing time, redundant pixels should be removed and only informative pixels are kept for processing. To accomplish this, a Focus Region (FR) is defined for each frame. The FR of a $P \times Q$ sized frame is extracted in the following steps.

(1) Each image is divided into non-overlapping sub-regions of size $(P/p) \times (Q/q)$ to get $p \times q$ number of sub-regions.
(2) The most external surrounding sub-regions (Colored with red in Fig. 1) are defined as the non-focus region.

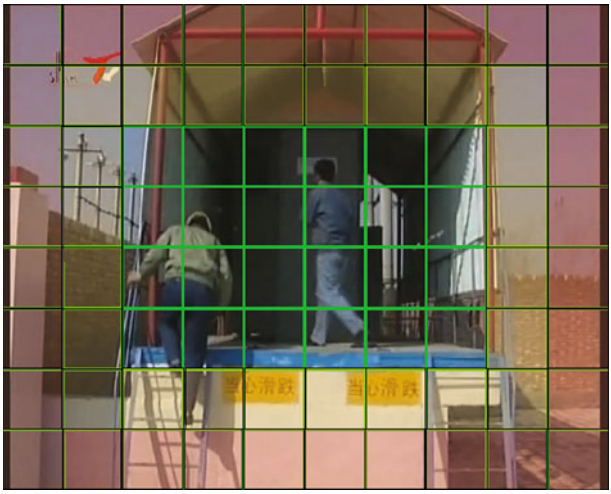

Fig. 1 Illustration of FR. The frame is partitioned into 8×10 sub-regions, and the outermost round sub-regions are non-focus region, second outermost round sub-regions are the second focus region, remaining sub-regions are the most focus region

(3) The outer-most external surrounding sub-regions (Yellow sub-regions) are defined as second focus region.
(4) Remaining sub-regions around the center are defined as focus region.

To get an informative while compact representation of a frame, the non-focus region is discarded, the second focus region is down-sampled by keeping only pixels with odd x-coordinates. The focus region is fully kept.

3.2 Temporally redundant frames reduction

Mutual Information (MI) [5] is an important conception in information theory to measure the amount of information that transported from one random variable to another. It measures the reduction of uncertainty of one random variable given the knowledge of another. Actually, authors of [28, 33] utilized the MI between two video frames to measure the difference for SBD, and these methods achieved satisfactory performance. Thus, the MI is used in our approach for frames similarity measure. However, the computation cost is still high with MI computation. Therefore, in this subsection, we first proposed an efficient frame similarity measure.

In addition, although some of the recent SBD methods can do well in CUT detection, most of them are sensitive to GT detection because the changes between two consecutive frames are not so evident in GT, as shown in Fig. 5. However, frames on the two sides of GT are very different. Thus, if we skip the intermediate frames, the GT can be regarded as a CUT (Fig. 2). Thus we propose a skipping interval to do shot boundaries detection more accurately and fast. Namely, in order to maximally reduce the processed frames and also do not drop boundaries between two shots, we skip several frames at each time to find boundaries.

3.2.1 Efficient frame similarity measure

Entropy measures the information content or "uncertainty" of X and is given by [5]:

$$H(X) = -\sum p_X(x) \log p_X(x) \tag{1}$$

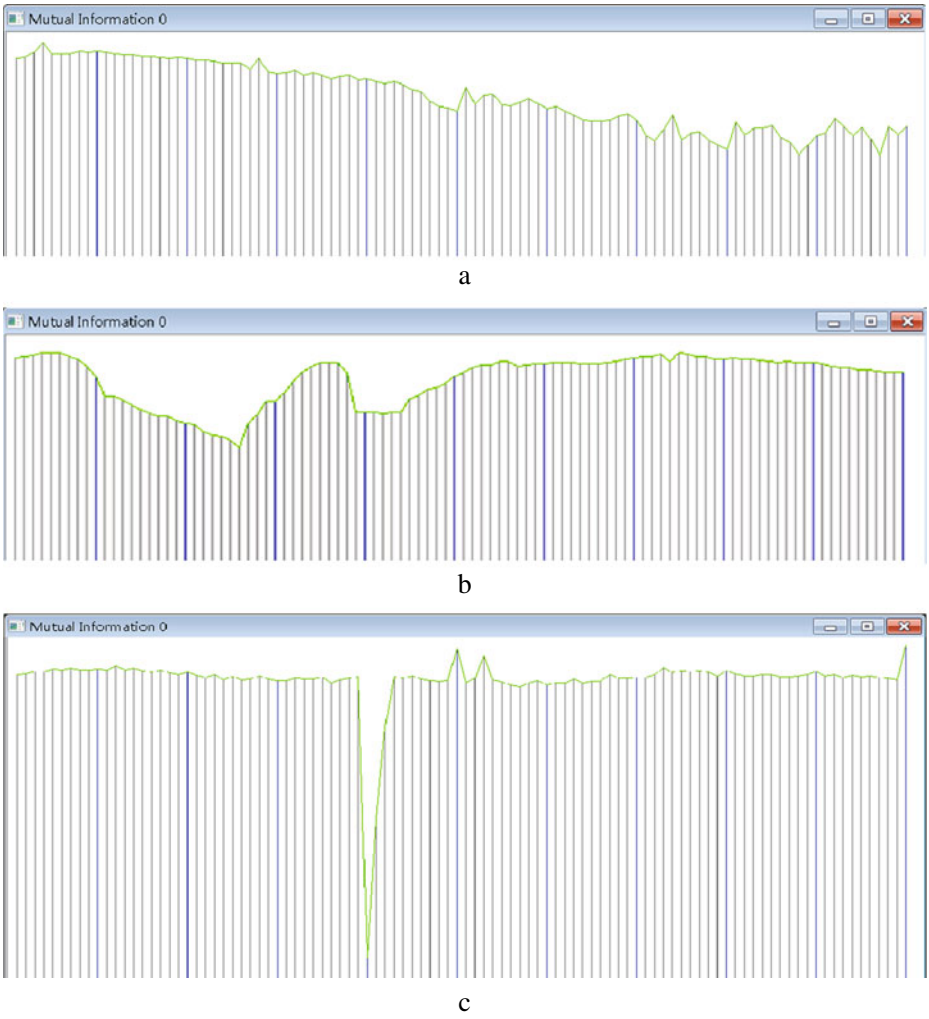

Fig. 2 Time series of the MI from video sequence on GT. The *top* two histogram **a** and **b** correspond to MI of consecutive frames [33], where the histogram in **c** is calculated by skipping frames and corresponds to the same video of **b**

The joint entropy of X,Y is defined as:

$$H(X, Y) = -\sum p_{XY}(x, y) \log p_{XY}(x, y) \tag{2}$$

The MI between the random variables X and Y is defined as:

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \tag{3}$$

Let $V = \{F_1, F_2, \ldots, F_N\}$ denotes the frames of a video clip V. For two frames (i.e. F_x and F_y), we first compute their own entropies (i.e. H_x, H_y) and their joint

entropy (i.e. $H_{x,y}$). The MI between them is given by (3). If $I^R_{x,y}$, $I^G_{x,y}$, $I^B_{x,y}$ respectively represent the MI of each RGB component, we set $I_{x,y} = I^R_{x,y} + I^G_{x,y} + I^B_{x,y}$ as the MI between frame F_x and F_y. We calculate the similarity ratio α between F_x and F_y as:

$$\alpha_{x,y} = \frac{\min(I_{x,x+1}, I_{x,y})}{\max(I_{x,x+1}, I_{x,y})} \tag{4}$$

In general, adjacent frames in the frame sequence are nearly the same, we can measure the similarity of two frames by $\alpha_{x,y}$. If $\alpha_{x,y} > T_\varepsilon$ (the threshold assigned in experiment), we can assert that $I_{x,x+1}$ is close to $I_{x,y}$ and F_x is similar to F_y.

Although MI between frames is robust to measure frame similarity, the computation cost is still quite high. However, when the entropy of frame is computing, we find that the distribution of gray values in each color channels are also obtained, which is similar to the color histogram. Therefore, it means that the color histogram can be easily and directly obtained from the entropy calculation. Thus, when both the color histogram H_x for F_x and H_y for F_y is obtained, $I_{x,y}$ in (4) can be replaced with the histogram intersection distance of H_x and H_y. While the color histogram can be more quickly and easily obtained, the computation cost is also saved. More specifically, when the MI is computing, we conduct the similarity ratio in (4) based on histogram intersection distance at first. If this ratio is very small (less than 0.5), we directly assert that these two frames are dissimilar and skip the remaining computation of MI calculation.

3.2.2 Efficient search strategy

There were some methods proposed to skip frames for more accurate and quicker shot boundaries detection [6, 17]. They chose fixed and arbitrary skipping interval d_j, for example, 5 frames or 20 frames. Smaller interval is sensitive for GT detection and with limited accelerating efficiency, while large interval might skip many shot boundaries and result in lower detection accuracy. Therefore, the skipping interval must be self-adaptive to achieve a satisfactory trade-off between detection accuracy and computation demand, and an efficient search strategy is proposed.

In order to efficiently reduce the number of processed frames and also not to drop any boundaries between two shots, we set the initial skipping interval as d_1. Then, the following skipping intervals are updated adaptively based on the similarity of frames. As shown in Fig. 3, $\{d_1, d_2, \ldots\}$ denotes the sequential skipping intervals, and

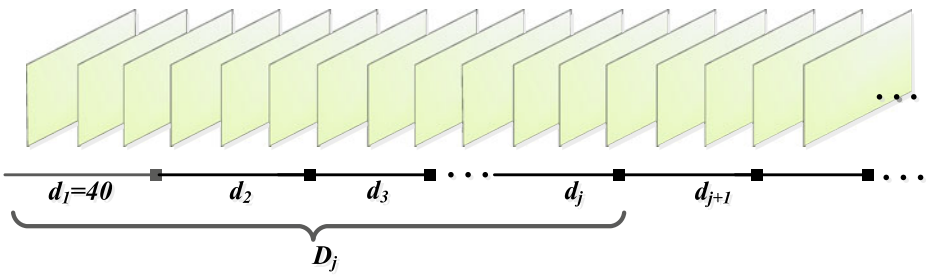

Fig. 3 Illustration of adaptive skipping interval

$\{D_1, D_2, \ldots\}$ is the serial frame number in the original video corresponding to all skipping intervals. D_k is defined as

$$D_k = \sum_{m=1}^{k} d_m \tag{5}$$

When $\alpha_{x,y}$ is the similarity ratio between the x_{th} and y_{th} frame (4), we update the skipping interval d_j as follows.

$$d_j = \sum_{k=1}^{j-1} \frac{1}{j-1} \alpha_{D_{d-1}, D_k} d_k \quad (6) \tag{6}$$

That is to say the greater α is, the larger of the skipping interval is. These updated intervals are reasonable. Because a great α in current skipping interval means the skipped frames are very similar, we can boldly skip more frames. But if α is small, it implies there are many changes in the skipped frames and we need to cautiously skip frames, so as to avoid classifying a motion as a shot boundary and also avoid missing shots with less frames. Generally, human visual reaction time is about 1–2 s. Suppose the video frame rate is about 20–25 fps, then a shot that can cause visual reaction need last for 20–50 frames at least. Therefore, the initial skipping interval d_1 is set to 40.

After that, given the current processed frame F_i, if $\alpha_{i,i+d_i} > T_\varepsilon$ (the threshold assigned in experiment), we can assert that F_i is similar to F_{i+d_j}, and skip to process the next d_{j+1} frames. Otherwise it means that there is a shot boundary existing between F_i and F_{i+d_j}. Thereby, we use a bisection search to find a refined boundary in this range. First, we compute $\alpha_{i,i+d_j/2}$ and $\alpha_{i+d_j/2,i+d_j}$. If $\alpha_{i+d_j/2,i+d_j} > T_\varepsilon$, boundary lies in the first half of F_i to F_{i+d_j}, otherwise in the second half. Then, the same process is carried out in the first half or second half of F_i to F_{i+d_j} to refine the boundary position until half of the range is only one frame. We explain the detailed searching process in Algorithm 1

Algorithm 1 Efficient search for candidate shot boundaries

Input: Video frame sequence $V = F_1, \ldots, F_N$.
Output: Candidate boundary set $CB = \{CB_1, CB_2, \ldots, CB_n\}$.

1: Set the starting frame as F_s and total skipped frame number $t = s$.
2: Initialize skipping interval $d_1 = 40$ and iterate variable $i = 1$.
3: **while** Processed frame do not exceed the frame sequence. **do**
4: Calculate the mutual information $I_{t,t+1}$ and $I_{t,t+d_i}$.
5: Compute the similarity rate $\alpha_{t,t+d_i}$.
6: **if** $\alpha_{t,t+d_i} > T_\varepsilon$ **then**
7: Set $t = t + d_i$ and $i = i + 1$.
8: Update the skipping interval d_i as (4).
9: **else**
10: Using binary search to get a candidate boundary between F_t and F_{t+d_i}.
11: Set $i = 1$.
12: Output the refined boundaries as candidate boundary set $CB = \{CB_1, CB_2, \ldots, CB_n\}$.

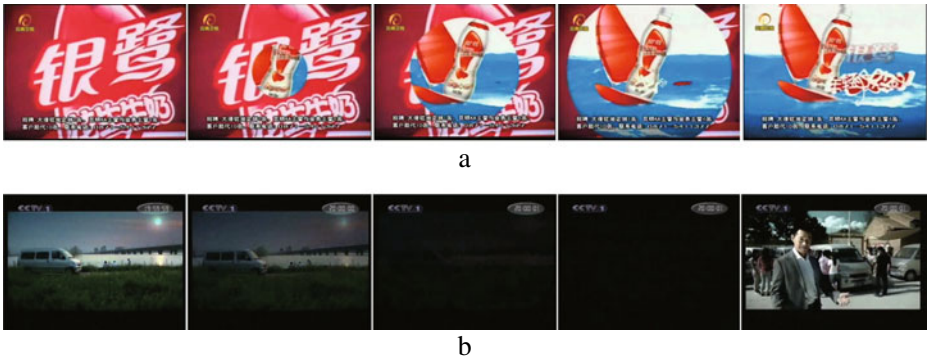

Fig. 4 Some GT examples: wipe and fade. **a** Circle wipe and **b** fade

We evidently accelerated the SBD process and detect GTs more robustly, no matter if the GT is fade in/out, dissolve or wipe, as shown in Fig. 4. Moreover, it requires to compute MI for $n-1$ times on a video sequence of n frames by using traditional frame by frame searching process, but in our approach, we just need to compute it for $\log n$ times.

So far, we have accelerated the process both in spatial and temporal domain, and nearly detected all the boundaries. However, camera or object motion could also lead to significant change of frame content when we skip frames aggressively. The above processes lead to several false shot boundaries which are caused by camera motion, camera zoom in/out, and object motion etc., as shown in Fig. 10. These false shot boundaries are named as Motion Caused False Boundaries (MCFB). Actually, most false shot boundaries in our approach belong to MCFB, thus in the next section, we will introduce how to detect and remove these false boundaries.

4 False shot boundaries removing

After the processing above, we can get two types of candidate shot boundary: the true shot boundary of CUT/GT, and the false one of MCFB. In this section we will elaborate how to use the corner distribution in frames to remove MCFBs. Corner distribution means the scattered and changed status of detected corner in frames. Because corners is the distinguishing feature of a frame, thus different frames have very different corner distribution. Namely, different candidate shot boundaries, such as CUTs, GTs and MCFBs, always have different corner distributions, and this property can be used for MCFBs removing.

4.1 Corner distribution of shot boundary and MCFB

Among all the corner detection methods, the Harris corner is the most common one [8], so we choose to use it. Actually, different types of candidate shot boundary have very different corner distribution, shown in Fig. 5. In Fig. 5, the corner distribution of frames in the GT from F_{n-4} to F_{n+4} is neither similar to frames in forward shot nor

Fig. 5 Corner distribution in gradual transitions from Frame $n-6$ to Frame $n+5$. The above 12 frames show the corners in each frame (from $n-6$ to $n+5$), and the candidate shot boundary is in frame n. In the symbol $'n:11759'$, $'n'$ denotes the frame number, and '11759' is the total number of Harris corners in the n_{th} frame, and the same for others

frames in backward shot, but it is fused by both ones. Corner distribution of frames in CUT is evidently different, such as the corner distribution of frames F_{3722} and F_{3723}, shown in Fig. 6. In MCFB, like the GT, consecutive frames are so similar, and their corner distributions are consistent, as seen in Fig. 7. However, the details of GT with editing effect can not be observed precisely by audience. But, audience can easily fell camera or object motion. Thus, frames in MCFBs last longer than that in CUTs or GTs. For example, the camera motion last about 150 frames in Fig. 7, but only 9 frames in GT of Fig. 5. It means that the frames change smoother in MCFB than that in GT, and the corner distributions in MCFB and GT are totally different, shown in Fig. 8.

Fig. 6 Example of corner distribution in a CUT between Frame 3722 and Frame 3723

Fig. 7 A sequence of frames in a camera motion

4.2 MCFB removal

As discussed in above sections, in CUT, a frame abruptly changes into an totally different one. But, changes in GT always last about 5–20 frames, which can not be awared by audience. However, changes in MCFB always last more than 100 frames. Corner distribution of frames in true boundary is very different from its forward and backward frames, regardless of GT or CUT, but it is more stable and consistent in MCFB. For example, in Fig. 5, the detected shot boundary in F_n with 11759 corners is very different from F_{n-5} with 6113 corners and F_{n+5} with 5957 corners. Therefore, for a candidate shot boundary position F_n, we compare its corner distribution with that in F_{n-c} and F_{n+c}, where c is the interval (which is not bigger than 15 in our approach). If the corner distribution of F_n is similar with that of F_{n-c} and F_{n+c}, the corner distributions is stable and consistent for frames near this candidate shot boundary, and we can assert the candidate shot boundary locate in F_n is a MCFB. We will describe in detail on how to judge if two frames is similar or not by corner distribution in the next paragraph.

In order to judge if two frames are similar, it is convenient to compare the total Harris corner number of these frames. But it is inaccurate and crude just using this simple feature, and we introduce the local corner distribution. For two frames of size $K \times L$, the local corner distribution is used to measure their similarity, general steps are as follows.

(1) Each frame is divided into $K/k \times L/l$ blocks, and each block is of size $k \times l$, as shown in Fig. 9.
(2) Corners in each block are detected by the Harris corner detector [8]. $Num_A(u, v)$ and $Num_B(u, v)$ denote the number of corners detected in block (u, v) of frame A and B respectively (suppose $Num_A(u, v) < Num_B(u, v)$, otherwise, exchange the label of A and B).
(3) Then, we compare the corner distribution of blocks in the same position of the two frames. Each block position (i.e. (u, v)) generates a compare label $CR_{u,v}$

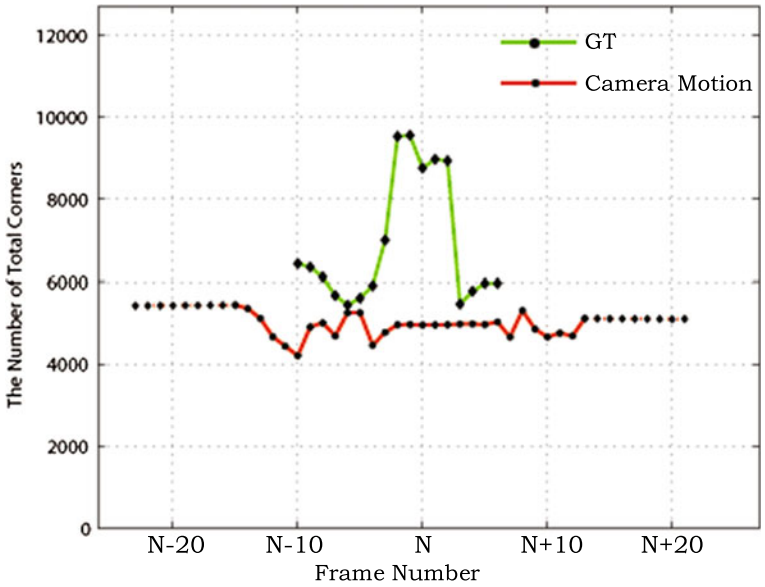

Fig. 8 Total corner numbers calculated in a GT and a camera motion. The GT and camera motion is aligned with the frame number of N

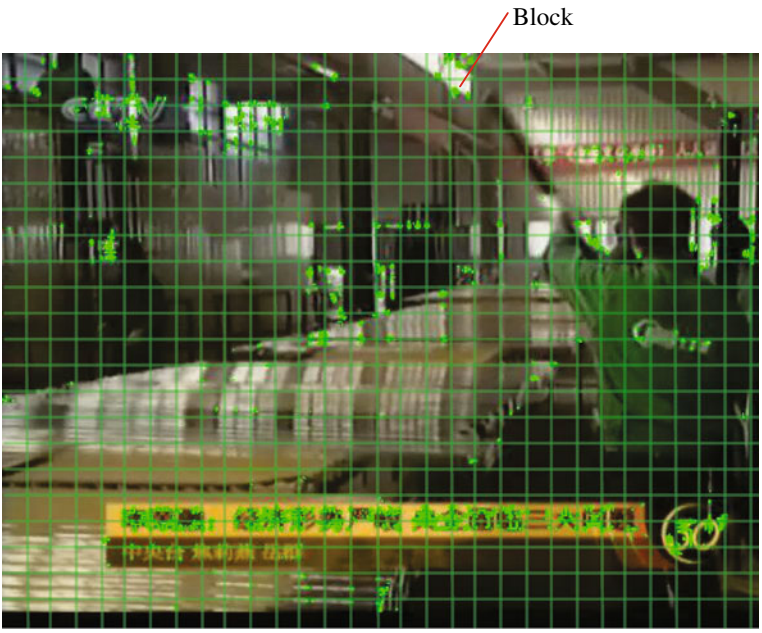

Fig. 9 Blocks in a frame of size $k \times l$

which represents the matching between the two blocks in this position and the matching rules are as follows.

$$CR_{u,v} = \begin{cases} 1 \text{ if } Num_A(u,v)/Num_B(u,v) > T_\theta \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

When $CR_{u,v}$ equals to 1, it corresponds to match. T_θ is the threshold, which is empirically fixed with 0.82 in our approach. Finally, we count the proportion of blocks with compare label $CR = 1$.

(4) The proportion of the matching blocks is calculated as,

$$MP = \frac{\sum_{u,v} CR_{u,v}}{k \times l} \tag{8}$$

If the proportion MP is bigger than 85 %, we assert that the two frames' local corner distribution is consistent and they are similar.

Generally, too small block size will be very sensitive to motions, and too large block size will overlook the local corner changes in GTs. Through observation, we empirically set size $k \times l$ as 20×20.

For example, Fig. 10a shows a shot in which the camera zooms in. The motion of the camera results in quick changes in consecutive frames, but our MCFB removing algorithm still successfully identifies the associated frames as being within a single shot. Figure 10b and c show two other cases of MCFB that are affected by sudden object moving and camera pan respectively, which are also successfully removed as false alarm.

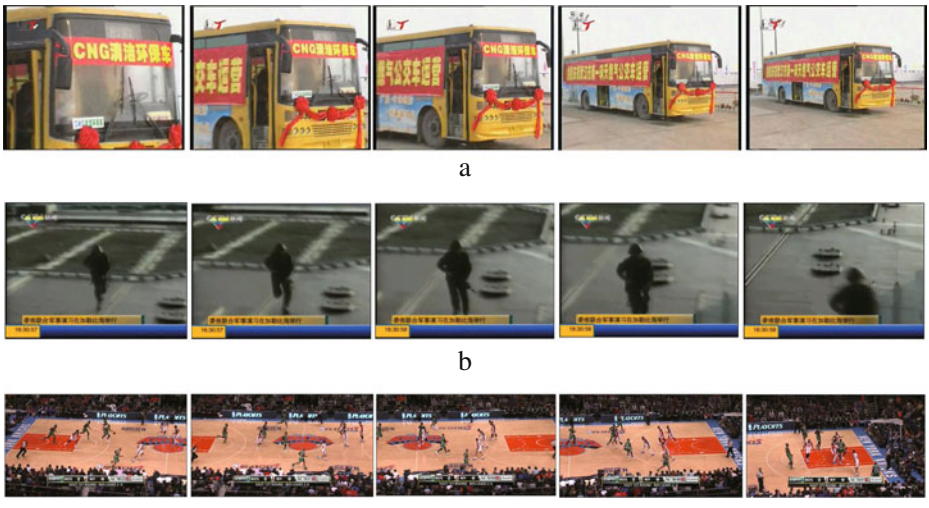

Fig. 10 Sample frames of some easily mis-detected shots. **a** Camera zoom in, **b** sudden object moving, and **c** camera pan motion

4.3 Intervals of gradual transition

The candidate boundaries found are only frame instants, not intervals. However, since many shot changes are gradual transitions, including fade in/out, wipe and dissolve, it is necessary to find the intervals of such transitions. Therefore, we find the transition interval by using the corner distribution of frames. Corner distribution of adjacent frames belong to the same shot is likely to be very similar. Thus, gradual changes occur when the number of corners varies greatly. In Section 4, the left and right frames, with corner distribution dissimilarity in the candidate transition, are possible start and end frames of that transition. The total corner number of the left frame F_{n+c} and that of its neighbors are computed, and also the first deviation of corner numbers series are calculated. When the position that the first deviation turns negative appeared, the left of the interval can found. Otherwise, the right of the interval can be obtained.

5 Experimental results

To evaluate the performance of our approach, 6 categories of videos are selected, including *news (2 clips from CCTV News Broadcast, 1 clip from Biz China), sport video (1 basketball clip, 1 football clip), movie (2 clips), teleplay (1 clip), commercials (2 clips from CCTV)* as well as *documentary (2 video fragments (Greenland Ice and Giant on the Bighornd) from TRECVID 2001, 1 clip from the 3th episode of CCTV documentary "The BRICS country")*. These videos are characterized by several CUTs, and GTs of dissolves, fade ins/outs and wipes, as well as significant camera parameter changes like zoom-ins/outs, pans and significant object and camera motions inside single shots. Except for the documentary with resolution of 352×240 and with frame rate of 29 fps, resolution of other videos vary from 640×480 to standard definition of 720×576, and the frame rate is 25 fps. Details of these test videos are shown in Table 1. In fact, in order to reasonably compare our approach with others, we have implemented all the compared methods in these test videos.

In order to evaluate different approaches with our approach, the following measures were used.

$$Recall = \frac{DT}{DT + LS} \tag{9a}$$

Table 1 The video set in our experiments

Video name	Duration (s)	Total frames	Shot number
News 1: CCTV news broadcast	3,324	83,100	692
News 2: Biz China	3,043	76,075	631
Sports 1: football	2,356	58,900	378
Sports 2: basketball	1,413	35,325	203
Movie 1: set off	1,802	41,446	511
Movie 2: love at first Hiccup. 2009	1,026	25,650	268
Teleplay: the 3th episode of marriage battle	2,072	51,800	497
Commercial	203	5,075	152
Documentary	1,615	43,867	713

$$Precision = \frac{DT}{DT + FA} \tag{9b}$$

where *DT, LS, FA* are the number of correct detections, loss detections and false positives, respectively.

We have done experiments on both the processing efficiency and accuracy with our approach, and some examples of the segmented shots are shown in Fig. 10.

5.1 Parameters selection

Using the FR regions and adaptive skipping interval, we accelerated the SBD processing by reducing detection region and scope. Actually, the partition size for FR and the similarity measure of frames in skipping interval, is very important for SBD accuracy and efficiency. Thus, in this subsection, we detailed how to set these important parameters, including the partition size for FR and threshold T_ε in Section 3.2.1.

When vast pixels in each frame cost very long processing time, the partition size of FR is designed as the first step for elimination of redundant pixels. Different partition scheme will result in different SBD detection accuracy and efficiency. When we tested several kinds of partition sizes with resolution of 640×480, we find that partition size of 20×20 achieved the satisfactory accuracy and efficiency. Meanwhile, with partition size of 4×4, the accuracy decreased obviously. Thus, we also chose another two partition size, 15×15 (square region) and 8×10 (rectangle region) between size of 20×20 and 4×4 for evaluation. So, we listed and compared these 4 different partition sizes as well as the sub-sampling of frames in [3, 29] (only using upper halves of a frame), as seen in Table 2. We tested all the 5 kinds of partition scheme using both histogram features and MI features in *clip 1*.

From Table 2, we can find that, with resolution of 640×480, arbitrary using pixels in the upper halves of a frame have accelerated the process, but both the recall and precision are not satisfactory. With our FR based method, if the partition of the frame is too fine, it can not really speed up the SBD. But if the partition size is too large, the result is like the result of using upper halves pixels [3, 29]. The results show that it saved more time by using size of 8×10 than that of 15×15, but the accuracy is not much different. In addition, we also tested several other partition schemes. And it is also nearly satisfactory in accuracy and efficiency using partition size which approximate to size of 8×8, but, both the accuracy and efficiency will be not much different. Therefore, we chose 8×10 in our approach, which can segment videos into shots accurately and fast.

Table 2 Comparisons of different partition sizes

	Partition size	Recall (%)	Precision (%)	Cost time (s)
FR based method	20×20	99.0	98.3	612
	15×10	98.8	98.1	595
	8×10	98.1	97.3	487
	4×4	87.2	88.4	197
Method of [3, 29]		78.3	82.5	356

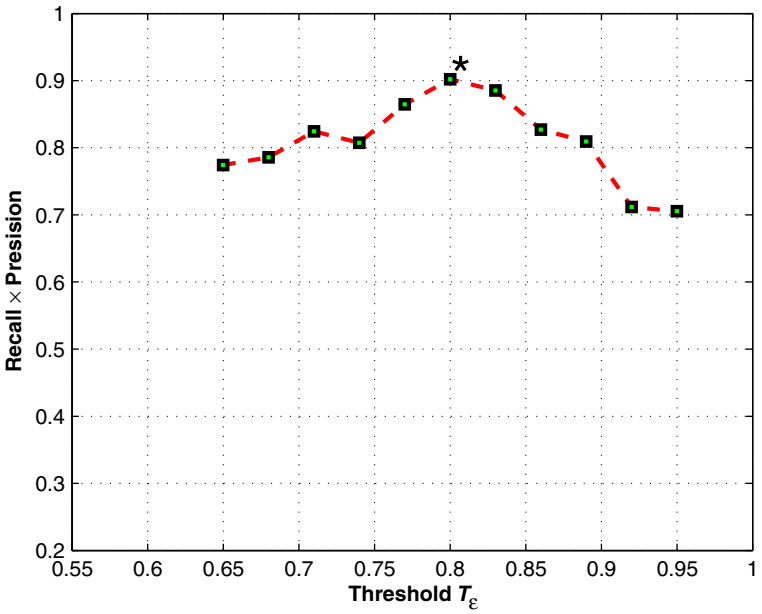

Fig. 11 The recall-precision graph obtained by varying threshold T_ε in the range [0.65, 0.95]

In addition, we also tested several choices of threshold T_ε based on the combination of recall and precision. The threshold value and recall-precision curve obtained with different threshold T_ε is shown in Fig. 11. From Fig. 11, we can find that the combination of recall and precision achieved the maximum value when threshold is equal to 0.8. Therefore, in our approach, we set $T_\varepsilon = 0.8$ to detect shot boundaries.

5.2 Comparison on accelerating results

The efficiency of the proposed approach is demonstrated by comparing with other methods published in literature in terms of detection speed and accuracy, such as [6, 17, 23, 33]. The information theory based methods such as [33], provided with

Table 3 Comparisons of FR and frame-by-frame

Videos	Recall of FR (%)	Precision of FR (%)	Cost time (s)	
			Focus region	All pixels
News 1	98.1	97.3	808	487
News 2	97.6	96.1	921	602
Sports 1	93.5	93.4	2,220	1,332
Sports 2	90.7	90.9	2,518	1,659
Movie 1	95.4	93.9	831	510
Movie 2	93.4	95.0	801	489
Teleplay	95.1	97.1	2,040	1,236
Commercial	89.2	88.1	136	91
Documentary	95.5	93.7.1	332	197

Table 4 Detection accuracy and efficiency comparisons

Accuracy and efficiency		FF	5F	*K*F	20F	40F	Ours
News 1	Recall (%)	96.7	97.0	91.1	85.1	85.8	97.1
	Precision (%)	99.7	98.1	95.6	81.0	89.2	99.3
	Cost time (s)	808	247	120	98	54	58
	Speedup ratio (FF/∗)	1	3.27	6.73	8.24	14.96	14.96
Sports 2	Recall (%)	90.7	90.9	84.2	79.3	80.9	94.9
	Precision (%)	88.2	88.5	79.0	79.5	80.2	90.3
	Cost time (s)	2,518	532	287	152	82	93
	Speedup ratio (FF/∗)	1	4.73	8.77	16.57	30.71	27.08
Movie 1	Recall (%)	94.2	95.7	80.0	80.7	85.2	95.7
	Precision (%)	98.5	96.5	87.8	88.2	86.1	96.5
	Cost time (s)	831	257	133	86	63	81
	Speedup ratio (FF/∗)	1	3.23	6.25	9.66	13.19	10.26
Teleplay	Recall (%)	93.7	96.0	87.3	90.1	80.5	96.0
	Precision (%)	94.8	97.9	86.5	87.9	87.5	97.9
	Cost time (s)	2,040	507	208	121	70	94
	Speedup ratio (FF/∗)	1	4.02	9.81	16.86	29.14	21.70

better results because it exploited the inter-frame information in a more compact way than frame subtraction based ones. Thereby, when we segment each frame into size of 8×10 for FR, we first compared the efficiency of our approach while using Focus Region with mutual information based method [33] which used all pixels. The results are shown in Table 3, and from which, we can see that the method based on FR can not only accelerate the process in spatial, but also have a satisfactory accuracy.

In addition, the authors of [6] proposed to detect shots with a fixed interval of 5 frames. And, the authors of [23] proposed a fast shot boundary detection method based on K-step slipped window, namely, they skip K ($K = 11$) frames each time to find shot boundaries. Meanwhile, Li et al. [17] have adventurously used 20 frames as the skipping interval. Thereby, we chose 4 video clips (one in each category) and provided the comparative results of different schemes, including the fixed interval of 40 frames ($40F$), 5 frames ($5F$), 11 frames (KF) in [23], 20 frames ($20F$) in [17], the frame-by-frame (FF) method based on mutual information [33] and our approach (Our's). Actually, the schemes with fixed interval used the same algorithm in [33] (i.e. mutual information based frames similarity measure) to detect shot boundaries. The skipping interval in our approach is started with 40 frames and updated by (5) and (6). Finally, the comparison results are presented in Table 4.

Table 5 Evaluation of fades detection of Cerneková's method [33] and our approach

Videos	Cerneková's method [33]		Proposed approach	
	Recall	Precision	Recall	Precision
News	0.93	0.95	1.00	1.00
Sports	0.89	0.90	0.94	0.93
Movies	0.92	0.92	1.00	0.97
Teleplay	0.93	0.92	0.95	0.95
Commercial	0.89	0.89	0.91	0.90
Documentary	0.91	0.90	0.92	0.96

Table 6 Evaluation of dissolves and wipes detection

	Recall		Precision	
	Dissolve	Wipe	Dissolve	Wipe
Proposed approach				
News	1.00	0.98	1.00	1.00
Sports	0.90	0.90	0.89	0.88
Commercial	0.88	0.86	0.81	0.85
Documentary	0.92	0.92	0.91	0.93
Cerneková's method [33]				
News	0.83	0.80	0.89	0.90
Sports	0.70	0.70	0.72	0.73
Commercial	0.58	0.69	0.70	0.68
Documentary	0.67	0.75	0.71	0.73

In Table 4, we can see that, too big value of the skipping interval will lose shot boundaries (i.e. the recall and precision of $20F$ and $40F$), but small ones don't really accelerate the SBD, i.e. the cost time of $5F$. We can find that considering the skipping interval, the cost time are evidently reduced, especially for the absolute cost time. In addition, the speedup ratio of $40F$ is even more large than our approach. It is because of the gradually reduced skipping interval in the adaptively skipping interval computation process. However, the accuracy results is obviously improved. And also, performed our approach on *Movie 1*, the speedup ratio is not improved so evidently. Because there are more complex background and motion contents as well as shots per-second in movies. Thus the adaptively updated skipping interval changed more frequently, and the MI computations are also increased in these intervals. Consequently, these process increased the cost time. It is not so reasonable to arbitrarily skip frames with a fixed value. We adaptively skipped frames to find the shot boundaries, and also the false candidate boundaries were removed by corner distribution analysis. Therefore, Table 4 also shows that almost all boundaries are detected by our approach, no matter it is CUT or GT. Especially, it increased the accuracy of GTs detection. Comparing with frame by frame processing method and method with fixed skip interval of frames, our approach really accelerated the SBD process and has more satisfactory detection accuracy.

5.3 Results of GT detection and MCFB removing

The same measure of recall and precision has been used for the evaluation of gradual transitions performance. As seen in Table 4, with skipping intervals and the bisection search algorithms, it achieved a good detection results. Besides, by

Table 7 Evaluation the effectiveness of MCFB moving

Videos	Without MCFB removing		MCFB removing	
	Recall	Precision	Recall	Precision
News	1.00	0.75	1.00	0.98
Sports	0.91	0.65	0.90	0.93
Movies	0.92	0.82	1.00	0.97
Teleplay	0.93	0.80	0.96	0.95
Commercial	0.89	0.61	0.90	0.89
Documentary	0.91	0.82	0.91	0.89

skipping interval, our approach not only accelerated the detection, but also got more accurate performance for GTs. Specifically, Cerneková Zuzana et al. proposed the joint entropy and MI based method to robustly detect fade transitions, and we

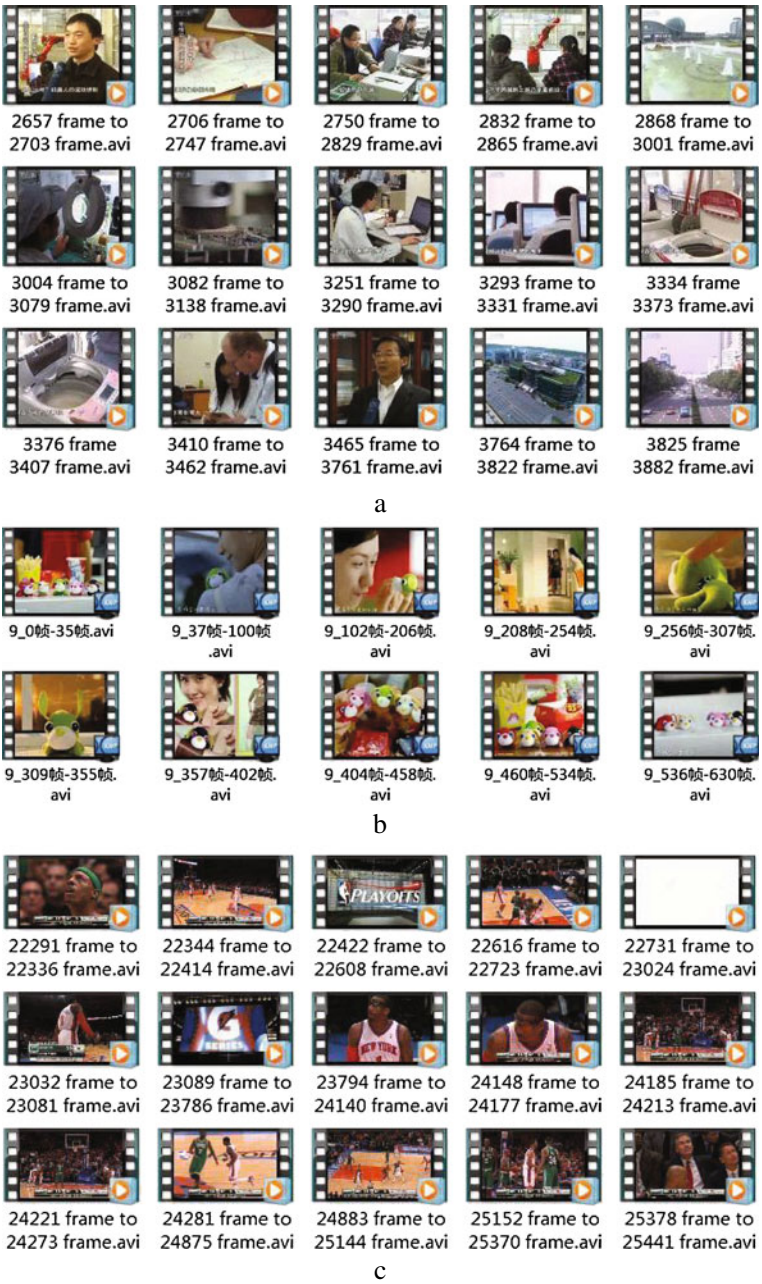

Fig. 12 Some examples of segmented shots in different videos. **a** Segmented shots in news video, **b** segmented shots in commercial, and **c** segmented shots in basketball game

compared our approach with them. As presented in Table 5, although the method of [33] detected most of the fades, there were still several dissolves and wipes missed. However, our approach can handle almost all GTs, including dissolves, fades and wipes, as shown in Table 6.

With our adaptively skipping interval which transforms GTs to the easy detected CUTs, almost all GTs have been detected and achieved with high recall performance. But the precision is very low since there are many false boundaries, such as camera motion and abrupt object moving. So we need to remove these false boundaries with the method proposed in Section 4. The improved experimental results are shown in Table 7 (Fig. 12).

6 Conclusion

The high costs of most SBD techniques have become a bottleneck in processing videos, particularly in real-time applications. In this paper, we have addressed this problem by decreasing the number of frames being processed per video, and also by choosing only those pixels located in FR per frame for frame similarity measure. By skipping several frames, we can accurately detect CUTs, and find more GTs, no matter what effects it has, dissolves, fades or wipes. Finally, for most of the false boundaries (i.e., MCFB) introduced by the above processes, the corner distribution of frame is used to remove these false boundaries and the experimental results show that our method not only speed up the SBD, but also achieve a satisfactory accuracy.

Acknowledgements The work reported in this paper is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant No.60925010, the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant No. 61121001, the Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT1049, the Co-sponsored Project of Beijing Committee of Education.

References

1. Adjeroh D, Lee MC, Banda N, Kandaswamy1 U (2009) Adaptive edge-oriented shot boundary detection. EURASIP J Image Video Process (USA) 2009:1–14
2. Cabedo XU, Bhattacharjee SK (1998) Shot detection tools in digital video. In: Proceedings of non-liner model based image analysis. Springer, Glasgow, pp 121–126
3. Chiu S-T, Lin G-S, Chang M-K (2008) An effective shot boundary detection algorithm for movies and sports. In: Proceedings of the 2008 3rd international conference on innovative computer information and control. Dalian, China, pp 173–176
4. Cotsaces C, Gavrielides MA, Pitas I (2005) A survey of recent work in video shot boundary detections. In: Proceedings of 2005 workshop on audio-visual content and information visualization in digital libraries (AVIVDiLib '05), 4–6 June 2005
5. Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York
6. Danisman T, Alpkocak A (2006) Dokuz Eyl´ul University video shot boundary detection at trecvid 2006. In: Proceedings of the TREC video retrieval evaluation (TRECVID)
7. Danisman T, Alpkocak A (2007) Bupt at trecvid 2007: shot boundary detection. In: Proceedings of the 2007 TREC video retrieval evaluation (TRECVID)

8. Derpanis KG (2004) The harris corner detector, New York
9. Han SH, Yoon KJ, Kweon IS (2000) A new technique for shot detection and key frames selection in histogram space. In: 12th workshop on image proceeding and image understanding, 2000
10. Hanjalic, A (2002) Shot-boundary detection: unraveled and resolved? IEEE Trans Circuits Syst Video Technol 12(2):90–105
11. Henga WJ, Ngan KN (2001) An object-based shot boundary detection using edge tracing and tracking. Vis Commun Image Represent 12(3):217–239
12. Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. IEEE Trans Syst Man Cybern, Part C Appl Rev 41(6):797–819
13. Huang C-L, Liao B-Y (2001) A robust scene-change detection method for video segmentation. IEEE Trans Circuits Syst Video Technol 11(12):1281–1288
14. Huang C-R, Lee H-P, Chen C-S (2008) Shot change detection via local keypoint matching. IEEE Trans Multimedia 10(6):1097–1108
15. Huang X, Ma H, Yuan H (2008) A hidden markov model approach to parsing mtv video shot. In: Proceedings of the 2008 congress on image and signal processing, vol 2, pp 276–280
16. Lefévre S, Holler J, Vincent N (2003) A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. Real-Time Imaging 9(1):73–98
17. Li Y, Lu Z, Niu X (2009) Fast video shot boundary detection framework employing pre-processing techniques. Image Process, IET 3(3):121–134
18. Lienhart RW (2001) Reliable dissolve detection. Storage Retr Media Databases. In: Proceedings of the SPIE Conference on storage and retrieval for media databases, vol 4315, pp 219–230
19. Ling X, Yuanxin O, Huan L, Zhang X (2008) A method for fast shot boundary detection based on svm. In: Proceedings of the 2008 congress on image and signal processing, vol 2. Sanya, China, pp 445–449
20. Mas J, Fernandez G (2003) Video shot boundary detection based on color histogram. In: Proceedings of the TREC video retrieval evaluation conference (TRECVID2003)
21. Pei S-C, Chou Y-Z (1999) Efficient mpeg compressed video analysis using macroblock type information. IEEE Trans Multimedia 1(4):321–333
22. Pei SC, Chou Y-Z (2002) Effective wipe detection in mpeg compressed video using macro block type information. IEEE Trans Multimedia 4(3):309–319
23. Qin T, Gu J, Chen H, Tang Z (2010) A fast shot-boundary detection based on k-step slipped window. In: Proceedings of 2010 IEEE international conference on network infrastructure and digital content, pp 190–195
24. Ren W, Sharma M, Singh S (2001) Automated video segmentation. In: International conference on information, communication, and signal processing, Singapore
25. Su C-W, Liao H-Y, Tyan H-R, Fan K-C, Chen L (2005) A motion-tolerant dissolve detection algorithm. IEEE Trans Multimedia 7(6):1106–1113
26. Tapu R, Zaharia T (2011) A complete framework for temporal video segmentation. In: Proceedings of 2011 IEEE international conference on consumer electronics. Berlin, pp 156–160
27. Wolf MWW, Liu B (1998) An algorithm for wipe detection. In: Proceedings of international conference on image processing, vol 1, pp 893–897
28. Xia D, Deng X, Zeng Q (2007) Shot boundary detection based on difference sequences of mutual information. In: Proceedings of the fourth international conference on image and graphics. Chengdu, China, pp 389–394
29. Xiong W, Lee JC-M (1998) Efficient scene change detection and camera motion annotation for video classification. Comput Vis Image Underst 71(2):166–181
30. Yeo B-L, Liu B (1995) Rapid scene analysis on compressed video. IEEE Trans Circuits Syst Video Technol 5(6):533–544
31. Yuan J, Li J, Lin F, Zhang B (2005) A unified shot boundary detection framework based on graph partition model. In: Proceedings of the 13th annual ACM international conference on multimedia. Hilton, Singapore, pp 539–542
32. Zhu S, Liu Y (2009) Automatic scene detection for advanced story retrieval. Expert Syst Appl 36(3, Part 2):5976–5986
33. Zuzana C, Ioannis P, Nikou C (2006) Information theory-based shot cut or fade detection and video summarization. IEEE Trans Circuits Syst Video Technol 16(1):82–91

Guangyu Gao received the BS degree in computer science from Zhengzhou University, China, in 2007, He is currently a PhD candidate in Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, China. His research interests include multimedia computing and computer vision.

Huadong Ma is a Professor and Director of Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China. He received his PhD degree in Computer Science from Institute of Computing Technology, Chinese Academy of Science in 1995, MS degree in Computer Science from Shenyang Institute of Computing Technology, Chinese Academy of Science in 1990 and BS degree in Mathematics from Henan Normal University in 1984. He visited UNU/IIST as research fellow in 1998 and 1999, respectively. From 1999 to 2000, he held a visiting position in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. He was a visiting Professor at The University of Texas at Arlington from July to September 2004, and a visiting Professor at Hong Kong University of Science and Technology from Dec. 2006 to Feb. 2007. His current research focuses on multimedia system and networking, sensor networks and Internet of Things, and he has published over 100 papers and 4 books on these fields. He is in Editorial Board of Multimedia Tools and Applications.