Signal, Image and Video Processing (2021) 15:627–635
https://doi.org/10.1007/s11760-020-01785-2

ORIGINAL PAPER

Video shot boundary detection based on multi-level features collaboration

Shangbo Zhou[1] · Xia Wu[1] · Ying Qi[1] · Shuyue Luo[1] · Xianzhong Xie[2]

Received: 7 January 2020 / Revised: 1 September 2020 / Accepted: 14 September 2020 / Published online: 17 October 2020

Abstract
Video shot boundary detection (SBD) is a basic work of content-based video retrieval and analysis. Various SBD methods have been proposed; however, there exist limitations in the complexity of boundary detection process. In this paper, a simple yet efficient SBD method is proposed, and the aim here is to speed up the boundary detection and simplify the detection process without loss of detection recall and accuracy. In our proposed model, we mainly use a top-down zoom rule, the image color feature, and local descriptors and combine a kind of motion area extraction algorithm to achieve shot boundary detection. Firstly, we select candidate transition segments via color histogram and the speeded-up robust features. Then, we perform cut transition detection through uneven slice matching, pixel difference, and color histogram. Finally, we perform gradual transition detection by the motion area extraction, scale-invariant feature transform, and even slice matching. The experiment is evaluated on the TRECVid2001 and the TRECVid2007 video datasets, and the experimental results show that our proposed method improves the recall, accuracy, and the detection speed, compared with some other related SBD methods.

Keywords Color feature · SURF · Slice · SIFT · Motion area · Video shot boundary detection

1 Introduction

A video usually is composed of several scenes, and a scene represents a complete plot, which is made of one or more shots. The first step of splitting a video into shots is to find the border of adjacent shots, that is video shot boundary detection. Video shot boundary is mainly of two types, i.e., cut transition (CT) and gradual transition (GT). CT is a transition where the scene has a sudden change between two adjacent frames. GT may last for several or even tens of frames, which is an artificial shot transformation effect, including fade in and fade out, dissolve, wipe, swirl, etc.

For existing SBD methods, we can class them into two major categories: multiple traditional features-based methods and learning-based methods. Methods based on multiple traditional features have been proposed to extract more information of frames and realize efficient detection, such as color texture moments [1], pixel difference [2], color histogram [3,4], and feature descriptor [5–7]. Some improved image features-based methods have also been reported, such as structure information and wavelet transform [8], the scale-invariant feature transform (SIFT)-point distribution histogram [9], and the HLFPN and descriptor [10]. Besides, there are methods combining multiple features and technologies, such as the combination of color histogram, local descriptor, and singular value decomposition(SVD) [11] and the combination of color information and texture information [12]. To better describe image feature, some fuzzy theory-based methods have been reported. For example, küçüktunç et al. proposed a kind of fuzzy color histogram method [13]. Fuzzy color distribution chart (FCDC) is also employed by Fan et al. [14]. Moreover, Chakraborty and Bhaumik, respectively, use fuzzy correlation measure [15] and multiple feature amalgamation [16] to focus on CT detection. These methods usually extract several frame information and have good detection effect.

However, to convert features to other feature spaces, some feature amalgamation or dimensionality reduction technologies are used, leading to low speed.

✉ Shangbo Zhou
shbzhou@cqu.edu.cn

[1] College of Computer Science, Chongqing University, Chongqing, China

[2] Key Laboratory of Computer Network and Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing, China

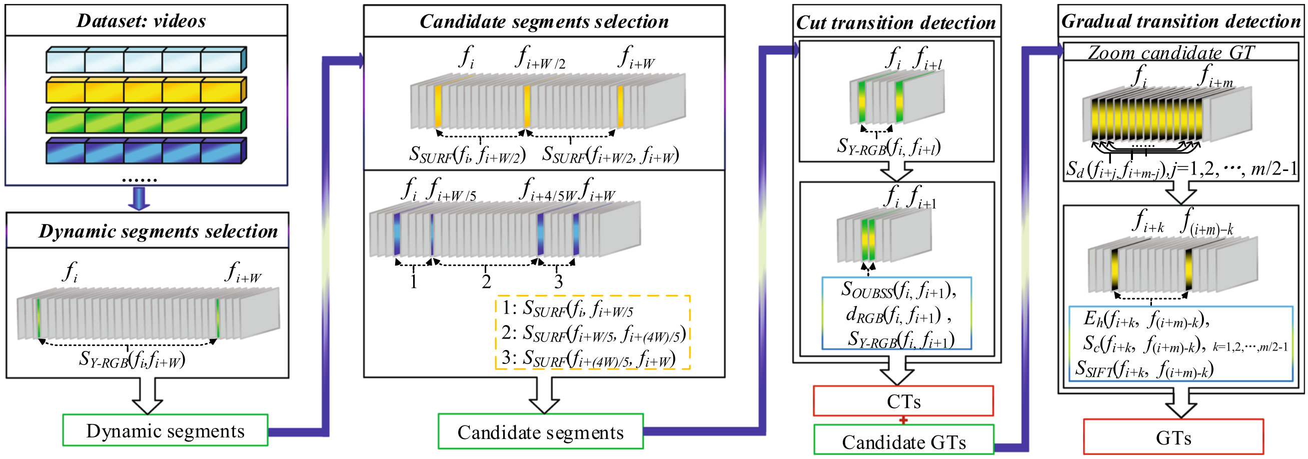
Dataset: videos
......
Dynamic segments selection
f_i
f_{i+W}
$S_{Y\text{-}RGB}(f_i, f_{i+W})$
Dynamic segments
Candidate segments selection
f_i $f_{i+W/2}$ f_{i+W}
$S_{SURF}(f_i, f_{i+W/2})$ $S_{SURF}(f_{i+W/2}, f_{i+W})$
f_i $f_{i+W/5}$ $f_{i+4/5W}$ f_{i+W}
1 2 3
1: $S_{SURF}(f_i, f_{i+W/5})$
2: $S_{SURF}(f_{i+W/5}, f_{i+(4W)/5})$
3: $S_{SURF}(f_{i+(4W)/5}, f_{i+W})$
Candidate segments
Cut transition detection
f_i f_{i+l}
$S_{Y\text{-}RGB}(f_i, f_{i+l})$
f_i f_{i+1}
$S_{OUBSS}(f_i, f_{i+1})$,
$d_{RGB}(f_i, f_{i+1})$,
$S_{Y\text{-}RGB}(f_i, f_{i+1})$
CTs
+
Candidate GTs
Gradual transition detection
Zoom candidate GT
f_i f_{i+m}
$S_d(f_{i+j}, f_{i+m-j}), j=1,2,\cdots, m/2-1$
f_{i+k} $f_{(i+m)-k}$
$E_h(f_{i+k}, f_{(i+m)-k})$,
$S_c(f_{i+k}, f_{(i+m)-k})$, $k=1,2,\cdots,m/2-1$
$S_{SIFT}(f_{i+k}, f_{(i+m)-k})$
GTs

Fig. 1 Overview of our proposed shot boundary detection model

The methods based on learning, such as genetic algorithm based [17], clustering based [18], support vector machine(SVM) based [19], and deep learning based [20], also usually have a good detection result. However, these learning-based methods usually need to train detection model with a long time, and the quality of training data has a great impact on the effect of shot boundary detection.

SBD requires not only high accuracy but also real-time performance. The methods based on multiple traditional features show that the shot boundary detection may be closely related to the collaboration of multiple image features. However, the detection process is complicated and time-consuming. In this paper, a simple and low time cost SBD method is proposed.

The main contributions are as follows. One is reducing the time cost with multi-level features collaboration. Another is using a motion area extracting algorithm to pay attention to motion area to improve GT detection performance.

The rest of the paper is organized as follows. The overview and detailed boundary detection process of our method are introduced in Sect. 2. The experimental results and analysis are listed in Sect. 3. The conclusion is stated in Sect. 4.

2 Proposed model

2.1 The whole framework of the detection model

The overview of the proposed model is shown in Fig. 1. The specific detection procedure consists of the following four parts.

(1) *Dynamic Segments Selection* dividing a video into several fixed-length sequences. The color histogram similarity of each sequence is used to judge whether a sequence is dynamic. If it is dynamic, it is considered to include shot boundaries. Otherwise, no boundary exists (see Sect. 2.3.1).

(2) *Candidate Segments Selection* obtaining candidate segment from dynamic segments. A dynamic segment may contain multiple candidate segments; thus, it is divided into multiple sub-segments and the speeded-up robust features (SURF) matching score of each sub-segment is calculated (see Sect. 2.3.2).

(3) *Cut Transition Detection* detecting CTs by the color histogram, overlapping uneven blocked SURF matching, and pixel average difference. By this step, the candidate GT segments also can be obtained (see Sect. 2.3.3).

(4) *Gradual Transition Detection* detecting GTs according to the SIFT matching score, evenly blocked histogram difference, and the motion area features (see Sect. 2.3.4).

2.2 Feature extraction ways

2.2.1 Extraction ways of color and statistical features

For dynamic segments selection and cut transition detection, we consider the color and statistical features to calculate the Y-RGB histogram similarity. A frame is sliced into N_s blocks ($N_s = 9$). And we use Eq. (1) to construct the Y-RGB histogram of the R, G, B channel image and gray image Y. Since the video frame is a 8-bit image ($2^8 = 256$), 256 bins are used to calculate the brightness changes of frame.

$$H_f = U_b^{N_s} U_i^{N_i} U_g^{N_g} \{h(b, i, g)\} \tag{1}$$

where N_i represents each block including four channels (R,G,B,Y) and N_g represents 256 bins. $h(b, i, g)$ represents the number of pixels whose bin value is g in the ith channel and the bth block. $U_b^{N_s} U_i^{N_i} U_g^{N_g} \{h(b, i, g)\}$ represents that all $h(b, i, g)$ are stacked to construct a one-dimensional vec-

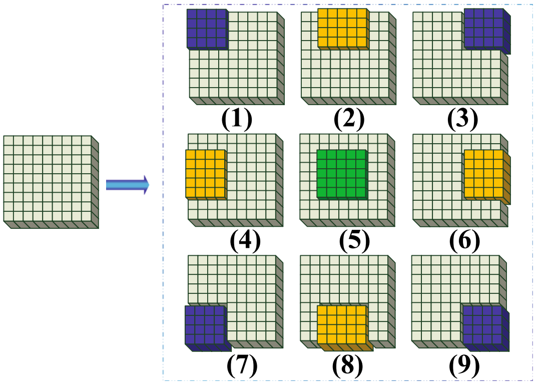

Fig. 2 Overlapping uneven blocked mechanism

tor H_f, which is equal to $[h(1,1,1), h(1,1,2), \ldots, h(9,4,256)]$.

For two frames f_1 and f_2, their Y-RGB histograms similarity $S_{\text{Y-RGB}}(f_1, f_2)$ is calculated with Eq. (2):

$$S_{\text{Y}-\text{RGB}}(f_1, f_2) = \cos(H_1, H_2) = \frac{H_1 \cdot H_2}{||H_1||\ ||H_2||} \tag{2}$$

where H_1 and H_2 represent the Y-RGB histogram.

To detect CTs more accurately, we also calculate the overlapping uneven blocked SURF matching (OUBSM) and the pixel average difference.

The calculation process of the OUBSM similarity which is denoted as $S_{\text{OUBSM}}(f_1, f_2)$ includes three steps. (1) Slicing two frames into 9 overlapping uneven blocks (Fig. 2), respectively. The ratio of width to height of the blue block is 4:4. The width to height of the orange block is 5:4. And the width to height of the green block is 5:5. (2) Calculating the SURF matching score of blocks in the corresponding area of the two frames. If the score of two blocks is less than th_{SURF}, it means that the two blocks are not matched, and the number of unmatched blocks of the two frame is increased by 1. (3) Calculating the OUBSM similarity with Eq. (3).

$$S_{\text{OUBSM}}(f_1, f_2) = \begin{cases} 0, & \text{Unmatched blocks} > 3 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

The pixel average difference d_n of two images is calculated with Eq. (4), (n = R, G, B). To make the pixel average difference has the same comparison standard, we normalize the pixel average difference to the value of [0,1]. The normalized pixel average difference d_{RGB} in RGB is calculated by Eq. (5). d_{R} , d_{G}, and d_{B} represent the pixel value difference of the R, G, and B channel image, respectively.

$$d_n(f_1, f_2) = \frac{1}{w \times h} \sum_{i=1}^{h} \sum_{j=1}^{w} |f_1(i,j) - f_2(i,j)| \tag{4}$$

$$d_{\text{RGB}}(f_1, f_2) = \frac{d_{\text{R}} + d_{\text{G}} + d_{\text{B}}}{3 \times 255}. \tag{5}$$

The aim of cut transition detection is to find two adjacent frames in a candidate segment, where the two frames have different scenes, and one is the last frame of the previous shot, and the other is the first frame of the next shot. For the characteristics of CT, the content of the adjacent shots may be a part of that of the previous shot, which may result in approximately similar color types and foreground targets of two shots. Therefore, the OUBSM can be used to reflect the structural feature, and the pixel average difference can reflect the subtle difference of two frames with similar color types and foreground objects, thus they are used.

The block similarity is used to narrow the range of candidate GT segment. Its calculation process is divided into five steps. (1) Slicing a frame into N_{s} blocks evenly with N_i channels($N_{\text{s}} = 50$, $N_i = 4$), and extracting the color histogram of each block in RGB and gray space. Each histogram has N_g bins($N_g = 32$). The histogram with 256 bins is converted into a histogram with only 32 bins by merging adjacent 8 bins. (2) Calculating the local average histogram difference d_b of each pair of blocks with Eq. (6) ($b = 1, 2, 3, \ldots, N_{\text{s}}$).

$$d_b(f_1, f_2) = \frac{\sum_{i=1}^{N_i} \sum_{g=1}^{N_g} |H_1(b,i,g) - H_2(b,i,g)|}{N_i \times N_g}. \tag{6}$$

(3) Calculating the global average histogram difference D of the two frames with Eq. (7).

$$D = \frac{\sum_{b=1}^{N_{\text{s}}} d_b}{N_{\text{s}}}. \tag{7}$$

(4) Calculating the number of unmatched blocks C_{um} of the two frames. C_{um} is initialized to 0. If d_b is larger than D, it means that the bth blocks of the two frames are not matched, and the value of C_{um} increases by 1. Otherwise, the two blocks are considered as matched, and C_{um} is unchanged. Then, calculating the block similarity S_{d} with Eq. (8).

$$S_{\text{d}} = \begin{cases} 0, & C_{\text{um}} > N_{\text{s}}/2 \\ 1, & C_{\text{um}} \le N_{\text{s}}/2 \end{cases}. \tag{8}$$

2.2.2 Extraction ways of local feature descriptors

In the process of candidate segments selection, the SURF matching score based on Fan's fuzzy color distribution char [14] is calculated. The tilt angle which ranges from − 90° to 90° is divided into 36 sets, and the tilt angle of each matching line respecting to the horizontal direction is calculated. If two frames are similar, there will be many matching lines on the same direction. The number of matching lines $C_1(i)$ in the ith set is calculated based on the tilt angle of each matching line.

The ratio of matching lines in three adjacent sets to the total matching lines $P_{\mathrm{l}}(i)$ is calculated with Eq. (9). The matching score $S_{\mathrm{SURF}}(f_1, f_2)$ is calculated with Eq. (10):

$$P_{\mathrm{l}}(i) = \frac{C_{\mathrm{l}}(i-1) + C_{\mathrm{l}}(i) + C_{\mathrm{l}}(i+1)}{\sum_{i=1}^{34} C_{\mathrm{l}}(i)}, \quad i = 1, 2, \ldots, 34 \tag{9}$$

$$S_{\mathrm{SURF}}(f_1, f_2) = \begin{cases} 1, & \max(P_{\mathrm{l}}(i)) > \mathrm{th}_{\mathrm{SURF}} \\ 0, & \max(P_{\mathrm{l}}(i)) \le \mathrm{th}_{\mathrm{SURF}} \end{cases}. \tag{10}$$

However, two frames may be almost no SURF matching line. For example, two solid color frames may have almost no keypoint, but they are similar. In this case, the SURF matching score of two frames is calculated with Eq. (11):

$$S_{\mathrm{SURF}}(f_1, f_2) = \begin{cases} 1, & N_1 < 25 \text{ and } N_2 < 25 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where N_1 and N_2 represent the number of the SURF keypoint of two frames, respectively.

2.2.3 Extraction ways of motion area features

To improve the accuracy of gradual transition detection, we extract the motion area of each candidate GT segment. Focusing on the frame motion area not only reduces some interference of background and foreground, but also allows us pay more attention to the content of the local area of frame where the visual content changes.

The process of motion area extraction includes five steps (Fig. 31–5). (1) Converting the adjacent three frames to grayscale images, and the absolute difference images of the first two images and the latter two images are, respectively, calculated. (2) Filtering the two difference images by mean filter, and performing an adaptive image binarization with the OTSU. (3) Performing a bitwise AND operation on the two binarization images to obtain a new binary image. (4) Performing a morphological operation on the binarization image to obtain a binary image which highlights the motion area. (5) Identifying the motion area of the original frame based on the motion area of the binary image. By the above five steps, the motion areas f_{m1} of the first three frames and f_{m2} of the last three frames of each candidate GT segment are obtained.

The normalized Euclidean distance E_{h} of the normalized histograms H_{m1} and H_{m2} of the motion area image f_{m1} and f_{m2} is calculated with Eq. (12). N_g represents 256 bins.

$$E_h = \frac{1}{1 + \sqrt{\sum_{i=1}^{N_g} [H_{\mathrm{m1}}(i) - H_{\mathrm{m2}}(i)]^2}}. \tag{12}$$

The calculation process of the main color similarity S_{c} of the motion area includes four steps (Fig. 36–7). (1) Two color histograms H_{m1} and H_{m2} of two motion areas with 256 bins are converted to two histograms with 32 bins by merging adjacent consecutive 8 bins. The 32 bins are considered as 32 colors. (2) Identifying the largest and the second-largest value of H_{m1}, and recording their corresponding color value s_1 and s_1, respectively. Same doing for H_{m2} and get l_2 and s_2. (3) Identifying the color value of H_{m1}, whose corresponding histogram value is greater than half of l_1, and they are denoted as $C_{\mathrm{m1}}(i), i = 1, 2, \ldots, C_1$, where C_1 represents the number of the color value satisfying the condition. With the same idea, the $C_{\mathrm{m2}}(j)$ of the histogram H_{m2} are identified, $j = 1, 2, \ldots, C_2$. (4) Calculating the main color similarity with Eq. (13). C_{s} represents the number of the same color value. If the main color similarity of the two frames is 1, they are considered as similar.

$$S_c = \begin{cases} 1, & l_1 = l_2 \text{ and } s_1 = s_2 \\ \frac{2C_{\mathrm{s}}}{C_1 + C_2}, & \text{otherwise} \end{cases}. \tag{13}$$

2.3 Description of our model

2.3.1 Dynamic segments selection

If a video segment involves multiple shots, it is considered to be dynamic. To remove some redundant video segments that belong to the same shot and reduce the detection time, we perform two steps to obtain dynamic segments: (1) dividing a video into several fixed-length sequences with W frames, where W is 50; (2) extracting the Y-RGB histograms of the first and last frame of each sequence with Eq. (1), and calculating the histogram similarity with Eq. (2). If the similarity is less than a threshold $\mathrm{th}_{\mathrm{Y\text{-}RGB}}$, the two frames are considered as dissimilar, that's to say the video sequence is dynamic; otherwise, the sequence is considered to be no boundary.

2.3.2 Candidate segments selection

As a dynamic segment with W frames may involve more than one shot boundary, each dynamic segment is divided into multiple sub-segments. The sub-segment which may contain a boundary is considered as a candidate segment. The SURF [6] can reduce the effects of rotation, illumination, and color variation of frames; thus, the SURF is used for matching of frames that may have different object or scene views.

The candidate segments are selected by dichotomy and trichotomy. First, each dynamic segment is divided into two sub-segments evenly, and the SURF matching score S_{SURF} of the first and last frame of the two sub-segments is, respectively, calculated with Eqs. (10)–(11). If both of the S_{SURF}

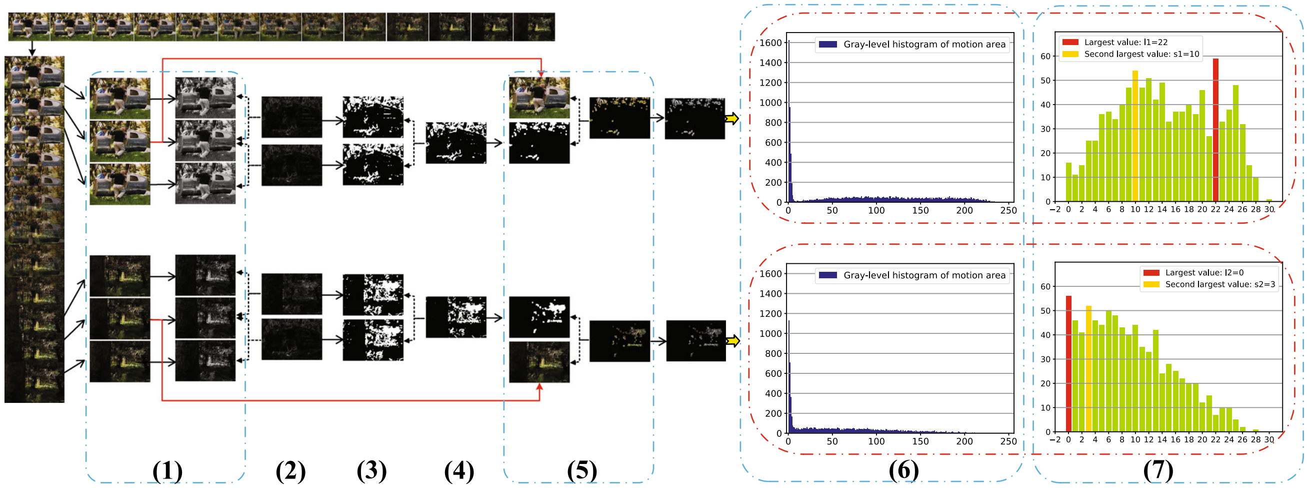
Gray-level histogram of motion area
Largest value: l1=22
Second largest value: s1=10
Largest value: l2=0
Second largest value: s2=3
(1)
(2)
(3)
(4)
(5)
(6)
(7)

Fig. 3 Process of motion area features extraction

equal 0, both the sub-segments are considered as candidate segments. If only one S_{SURF} is 0, the sub-segment with value 0 is considered as a candidate segment. If both of the S_{SURF} of the two sub-segments are equal to 1, the original dynamic segment is unevenly divided into three sub-segments with a length proportion of 1:3:1. The S_{SURF} of the three sub-segments are calculated with Eqs. (10)–(11). We determine whether there are candidate segments among the three sub-segments with the same judgment mechanism: If the S_{SURF} is 0, the corresponding sub-segment is considered as a candidate segment.

2.3.3 Cut transition detection

The process of CT detection is like a zooming process, which includes three parts. The candidate segment without a CT identified will be considered to be a candidate GT segment. The CT detection steps are as follows.

(1) Candidate cut transition segments selection. Each candidate segment is divided into several small segments with inter-frame distance th_{l}, and the Y-RGB histogram similarity of the first and last frame of each small segment is calculated with Eq. (2). If it is less than $\mathrm{th}_{Y\text{-RGB}}$, the small segment is considered as a candidate CT segment.
(2) Cut transition localization. The overlapping uneven blocked SURF matching similarity of two adjacent frames of each candidate CT segment is calculated with Eq. (3), and the normalized pixel average difference of the two frames is also calculated with Eqs. (4) and (5). If the OUBSM similarity is equal to 0 or the pixel average difference is greater than $\mathrm{th}_{\mathrm{d}\text{-RGB}}$, the adjacent frames are considered as candidate CT.
(3) Cut transition verification. To further reduce false detections, the cut transition verification is performed by calculating the similarity of candidate CT with Eqs. (10) and (11). If the similarity is less than $\mathrm{th}_{\mathrm{Y}\text{-RGB}}$, the candidate CT is considered as a CT.

2.3.4 Gradual transition detection

The process of gradual transition detection includes the following four parts:

(1) Merging segments. Two candidate gradual transition segments are merged when they are close to each other and their distance is less than or equal to threshold th_{l}.
(2) Removing wrong detected segments. Calculating the SIFT matching score $S_{\mathrm{SIFT}}(f_1, f_2)$ of the first and last frame of each candidate GT segment, if it is less than S_{SIFT}, the segment is considered as a true candidate GT segment; otherwise, no longer processing it. The reason why the SIFT is used to help GT detection is that the detection performance of SIFT on rotation is better than SURF [21]; thus, SIFT is more suitable for the detection of swirl GTs. Since the time cost of SIFT is almost three times that of SURF [21], SIFT is not used to perform candidate segments selection.
(3) Narrowing segment range. Calculating the block similarity of the first and last frame of the candidate GT segment with Eq. (8). If it equals 1, the current segment is considered as the smallest range that may contain a GT. If it equals 0, the next frame of the first frame of the current segment is taken as the first frame of candidate GT segment, and the previous frame of the last frame of the current segment is taken as the last frame.
(4) Calculating the length of each candidate GT segment that has been narrowed, and confirming gradual transition with the length, which involves motion area feature extraction and its similarity calculation.

Table 1 Test videos from TRECVid2001 and TRECVid2007

#	Video	Size	Minutes	Frames	Transition types		
					CT	GT	Total
1	anni005	320 × 240	6.19	11,362	40	27	67
2	anni009	320 × 240	6.50	12,305	39	66	105
3	BOR08	352 × 240	28.07	50,567	377	158	535
4	BOR12	352 × 240	13.39	24,551	2	129	131
5	NAD28	352 × 240	29.25	52,925	163	94	257
6	NAD31	352 × 240	29.08	52,394	155	29	184
7	NAD52	352 × 240	14.30	26,087	91	121	212
8	NAD53	352 × 240	14.31	26,114	83	80	163
9	NAD55	352 × 240	14.31	26,126	108	72	180
10	NAD57	352 × 240	6.57	12,509	45	25	70
11	NAD58	352 × 240	7.35	13,649	37	42	79
12	BG_22677	352 × 288	10.26	15,672	55	22	77
13	BG_35145	352 × 288	23.22	35,067	121	20	141
14	BG_35146	352 × 288	22.45	34,130	160	39	199
15	BG_37721	352 × 288	28.07	42,117	107	1	108
	Total		253.32	435,575	1583	925	2508

For the case where the original segment length is less than th_{seg}, the long segment is considered to contain at most one GT. The motion areas of the first three frames and the last three frames of the segment are extracted. And the following three feature similarities of the motion areas are calculated: the normalized Euclidean distance of the grayscale histogram calculated with Eq. (12), the main color similarity calculated with Eq. (13), and the SIFT matching score. If the normalized Euclidean distance is less than th_{Euro}, the segment is considered as a GT; otherwise, if the main color similarity is less than th_{color} and the SIFT matching score is less than th_{SIFT}, the segment is considered as a GT.

For the case where the length of the long segment is greater than or equal to th_{seg}, the long segment is thought to contain two GTs; thus, it is evenly divided into two sub-segments. We use the following rule to confirm GT.

(1) The three motion areas of the two sub-segments and the long segment are extracted, and the normalized Euclidean distance (E_h) of the motion areas is calculated with Eq. (12). We perform GT detection with the following three conditions: If E_h of the first sub-segment is less than th_{Euro}, the sub-segment is considered as a GT; otherwise, if E_h of the second sub-segment is less than th_{Euro}, the second sub-segment is considered as a GT; otherwise, if E_h of the long segment is less than th_{Euro}, the long segment is considered as a GT.

(2) If all the above three conditions are not met, the main color similarity of the two motion areas of the two sub-segments is calculated also the SIFT matching scores. Then, GT detection is performed with the following two conditions: For the first sub-segment, if the similarity is less than th_{color} and the matching score is less than th_{SIFT}, the first sub-segment is considered as a GT; otherwise, for the second sub-segment, if the similarity is less than th_{color} and the matching score is less than th_{SIFT}, the second sub-segment is considered as a GT.

(3) If all the above two conditions are not met, and the SIFT matching score of the first sub-segment and the second sub-segment is less than th_{SIFT}, the main color similarity and the SIFT matching score of the motion area of the long segment are calculated, respectively. Then, GT detection is performed with the following condition: If the main color similarity is less than th_{color} and the SIFT matching score is less than th_{SIFT}, the long segment is considered as a GT.

3 Experiments results and analysis

3.1 Experiment environment and evaluation standards

The experiments are performed by the Visual Studio 2012 platform and the OpenCV function Library with C++. And the same evaluation standards as in [16] are used.

3.2 The test dataset

The proposed model is tested on TRECVid2001 and TRECVid2007 datasets (Table 1). These videos have diverse characteristics, such as illumination effects of indoor and out-

Table 2 Thresholds of shot boundary detection

Symbol	Threshold definition	Value
th$_{\text{Y-RGB}}$	Y-RGB histogram similarity	0.93
th$_{\text{SURF}}$	SURF matching score	0.5
th$_{\text{D-RGB}}$	Pixel average difference	0.3
th$_{\text{I}}$	Inter-frame distance	5
th$_{\text{seg}}$	Candidate GT segment length	38
th$_{\text{Euro}}$	Histogram Euclidean distance	0.94
th$_{\text{color}}$	Main color similarity	0.75
th$_{\text{SIFT}}$	SIFT matching score	0.54

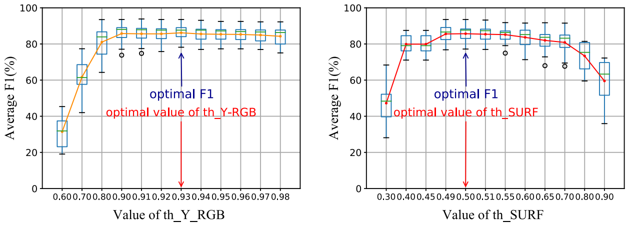

Fig. 4 *F* score of threshold th$_{\text{Y-RGB}}$ and th$_{\text{SURF}}$ with different values

door, intermingling of simulated and real frames, and camera movement or zoom. In this paper, the segments which fade out a shot to a shot without visual content and fade in a shot with visual content are considered as two GTs.

3.3 Analysis of threshold selection

In the proposed model, there are eight thresholds, as shown in Table 2. We mainly justify and explain two thresholds' selection, the th$_{\text{Y-RGB}}$ and th$_{\text{SURF}}$, and the other six thresholds' selections are qualified in the same way. For a threshold, by comparing the boundary detection effects (F1) with different threshold values, we get the most appropriate value of the threshold for different videos (Table 2).

To improve the robustness of each threshold on different videos, we performed lots of experiments on 10 different types of videos to get the trend and the appropriate range of each threshold with the control variable method, not the adaptive threshold method. The appropriate range of threshold th$_{\text{Y-RGB}}$ and th$_{\text{SURF}}$ for different videos is shown in Fig. 4. As shown in the two box plots in Fig. 4, when th$_{\text{Y-RGB}}$ is 0.93 and th$_{\text{SURF}}$ is 0.5, the mean value of $F1$ for different videos is the highest, with lowest discrete degree and without outliers. It can be seen that the optimal value range of the two thresholds converges to 0.93 and 0.5. Experimental results show that the threshold range is robust for datasets.

3.4 Statistic of experiment result

The experiment results are evaluated on TRECVid2001 and TRECVid2007. The average recall, precision, and $F1$ of boundary detection are shown in Table 3. And the bold in Table 3 represents the average F1 value of our proposed method on the data set. The experimental results are obtained with the optimal thresholds.

3.5 Comparison with other methods

To show the superiority of our proposed method, we compare our proposed method with the PS method [7], the

Table 3 Experiment results on TRECVid2001 and TRECVid2007

Video	Cut			Gradual			Total		
	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)
anni005	85.0	85.0	85.0	77.8	84.0	80.8	81.4	84.5	82.9
anni009	84.6	94.3	89.2	69.7	88.5	78.0	77.2	91.4	83.6
BOR08	83.6	96.3	89.5	53.8	81.7	64.9	68.7	89.0	77.2
BOR12	100.	100.	100.	69.0	95.7	80.2	84.5	97.9	90.1
NAD28	92.0	97.4	94.6	79.8	47.8	59.8	85.9	72.6	77.2
NAD31	91.6	98.6	95.0	79.3	48.9	60.5	85.5	73.8	77.8
NAD52	98.9	93.8	96.3	76.9	84.6	80.5	87.9	89.2	88.4
NAD53	97.6	96.4	97.0	77.5	80.5	79.0	87.6	88.5	88.0
NAD55	99.1	98.2	98.6	90.3	86.7	88.4	94.7	92.4	93.5
NAD57	95.6	100.	97.7	84.0	72.4	77.8	89.8	86.2	87.8
NAD58	86.5	100.	92.8	83.3	81.4	82.4	84.9	90.7	87.6
BG_22677	90.9	98.0	94.3	59.1	54.2	56.5	75.0	76.1	75.4
BG_35145	97.5	99.2	98.3	90.0	42.9	58.1	93.8	71.0	78.2
BG_35146	95.0	85.4	89.9	66.7	57.8	61.9	80.8	71.6	75.9
BG_37721	99.1	100	99.5	100.	100.	100.	99.5	100.	99.8
Average	93.1	96.2	**94.6**	77.0	74.0	**75.5**	85.1	85.1	**85.1**

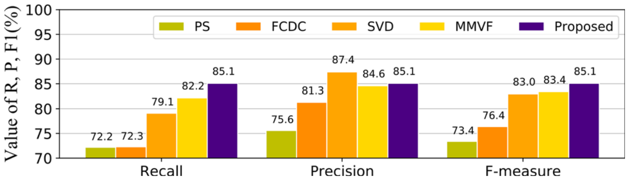

Fig. 5 Comparison results on TRECVid2001 and TRECVid2007

FCDC method [14], the SVD method [11], and the MMVF method [4] on the recall, precision, F-measure, and time cost. Figure 5 and Table 4 show the quantitative comparison results. And the bold font in Table 4 indicates the optimal value of our proposed method and all comparison methods on each evaluation index. As shown in Table 4 and Fig. 5, our proposed method has the highest $F1$ performance with an average CT and GT detection at 85%, while the method in [7] is 73%, 76% in [14], and 83% in [4,11]. Our proposed method has better recall, precision of CT detection, and better recall of GT detection compared with the four related methods, but the precision of GT detection is slightly lower than them. On the whole, our F1 is superior.

Compared with [8], we perform CT and GT detection. Fan et al. [14] constructs a kind of fuzzy color distribution chart, but not fully considers the rapid object movement, and lacks robustness to the adjacent shots whose color is similar. We consider it and improve it by image slice. For [11], it performs detection mainly based on color histogram and SVD, which lacks robustness to light, object, and camera motion. Fan et al. [4] tries to use SURF descriptors, color histogram, and frame difference to reduce the influence of logo, texts, ignoring the effects of dramatic illumination changes. We consider the interference of large object movement and dramatic flashlight changes with motion area extraction.

Our method improves the recall of GT detection and has the highest F-measure, with lower precision of GT detection, reducing boundary missing. In the comparison experiments, the optimal parameters are used as default; see related paper for details. Our aim is to propose a general method to perform CT and GT detection with traditional features.

Figure 6a shows the average processing time of our proposed model, which shows that the highest and lowest average processing time is 47 ms and 6 ms. Figure 6b shows the comparative results of average processing time. Our proposed method has a low time cost which is about 31 ms. Although the time cost of our method is higher than [14], ours have higher F-measure. Moreover, the average processing time of our proposed method is lower than [4,7,11]. Compared with [7], we do not perform SURF matching of all adjacent frames of each candidate segment frame by frame. For [11], it constructs feature matrix of almost all frames and perform singular value decomposition. [4] repeatedly performed SURF matching and extracted color histogram of all frames. Those make the average processing time of [4,11] much higher than ours, which are, respectively, about 36 times and 21 times.

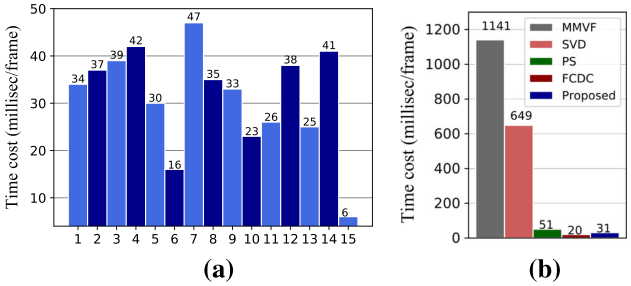

Fig. 6 **a** Average processing time of our proposed method, **b** comparison of average processing time

4 Conclusion

In this paper, a video shot boundary detection approach based on multi-level features collaboration is proposed, considering CT and GT. We perform effective shot boundary detection mainly based on color feature and feature descriptors, as well as a kind of frame motion area extraction method and a kind of top-down zoom rule. The experimental results have shown the effectiveness of our proposed method. Compared with some other related shot boundary detection methods, our proposed method has an efficient detection performance, simple detection process, and low time cost.

Table 4 Comparison results on TRECVid2001 and TRECVid2007

Methods	Cut			Gradual			Total		
	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)
Proposed	**93.1**	**96.2**	**94.6**	**77.0**	74.0	**75.5**	**85.1**	85.1	**85.1**
PS [7]	72.1	88.0	79.2	72.3	63.3	67.5	72.2	75.6	73.4
FCDC [14]	82.2	86.1	84.1	62.4	76.5	68.7	72.3	81.3	76.4
SVD [11]	89.2	95.9	92.5	68.9	**78.9**	73.6	79.1	**87.4**	83.0
MMVF [4]	91.5	94.2	92.8	73.0	75.1	74.0	82.2	84.6	83.4

References

1. Shekar, B., Kumari, M.S., Holla, R.: Shot boundary detection algorithm based on color texture moments. In: International Conference on Advances in Communication, Network, and Computing, pp. 591–594. Springer (2011)
2. Jiang, X., Sun, T., Liu, J., Chao, J., Zhang, W.: An adaptive video shot segmentation scheme based on dual-detection model. Neurocomputing **116**, 102–111 (2013)
3. Li, Z., Liu, X., Zhang, S.: Shot boundary detection based on multilevel difference of colour histograms. In: 2016 First International Conference on Multimedia and Image Processing, pp. 15–22. IEEE (2016)
4. Tippaya, S., Sitjongsataporn, S., Tan, T., Khan, M.M., Chamnongthai, K.J.I.A.: Multi-modal visual features-based video shot boundary detection. IEEE Access **5**, 12563–12575 (2017)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
6. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer (2006)
7. Birinci, M., Kiranyaz, S.: A perceptual scheme for fully automatic video shot boundary detection. Signal Process. Image Commun. **29**(3), 410–423 (2014)
8. Mishra, R., Singhai, S., Sharma, M.: Video shot boundary detection using dual-tree complex wavelet transform. In: 2013 3rd International Advance Computing Conference, pp. 1201–1206. IEEE (2013)
9. Hannane, R., Elboushaki, A., Afdel, K., Naghabhushan, P., Javed, M.: An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram. Int. J. Multimed. Inf. Retr. **5**(2), 89–104 (2016)
10. Shen, R.-K., Lin, Y.-N., Juang, T.T.-Y., Shen, V.R., Lim, S.Y.: Automatic detection of video shot boundary in social media using a hybrid approach of HLFPN and keypoint matching. IEEE Trans. Comput. Soc. Syst. **5**(1), 210–219 (2017)
11. Youssef, B., Fedwa, E., Driss, A., Ahmed, S.J.C.V., Understanding, I.: Shot boundary detection via adaptive low rank and svd-updating. Comput. Vis. Image Underst. **161**, 20–28 (2017)
12. Kavitha, J., Jansi Rani, P.A., Sowmyayani, S.: Wavelet-based feature vector for shot boundary detection. Int. J. Image Graph. **17**(01), 1750002 (2017). https://doi.org/10.1142/S0219467817500024
13. küçüktunç, O., Güdükbay, U., Ulusoy, Ö.: Fuzzy color histogram-based video segmentation. Comput. Vis. Image Underst. 114(1), 125–134 (2010)
14. Fan, J., Zhou, S., Siddique, M.A.: Fuzzy color distribution chart-based shot boundary detection. Multimed. Tools Appl. **76**(7), 10169–10190 (2017)
15. Chakraborty, B., Bhattacharyya, S., Chakraborty, S.: An unsupervised approach to video shot boundary detection using fuzzy membership correlation measure. In: 2015 Fifth International Conference on Communication Systems and Network Technologies, pp. 1136–1141. IEEE (2015)
16. Bhaumik, H., Bhattacharyya, S., Chakraborty, S.: A vague set approach for identifying shot transition in videos using multiple feature amalgamation. Appl. Soft Comput. **75**, 633–651 (2019)
17. Thounaojam, D.M., Khelchandra, T., Singh, K.M., Roy, S.: A genetic algorithm and fuzzy logic approach for video shot boundary detection. Comput. Intell. Neurosci. **2016**(1), 14 (2016)
18. Yazdi, M., Fani, M.: Shot boundary detection with effective prediction of transitions' positions and spans by use of classifiers and adaptive thresholds. In: 2016 24th Iranian Conference on Electrical Engineering (ICEE), pp. 167–172. IEEE (2016)
19. Mondal, J., Kundu, M.K., Das, S., Chowdhury, M.: Video shot boundary detection using multiscale geometric analysis of nsct and least squares support vector machine. Multimed. Tools Appl. **77**(7), 8139–8161 (2018)
20. Xu, J., Song, L., Xie, R.: Shot boundary detection using convolutional neural networks. In: 2016 Visual Communications and Image Processing (VCIP), pp. 1–4. IEEE (2016)
21. Juan, L., Gwon, O.: A comparison of sift, pca-sift and surf. Int. J. Image Process. **8**(3), 169–176 (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.