Shot Boundary Detection Using Convolutional Neural Networks

Jingwei Xu [1], Li Song [1,2], Rong Xie [1,2]
[1] *Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University*
[2] *Cooperative Medianet Innovation Center, Shanghai, China*
Email: {superxu, song_li, xierong}@sjtu.edu.cn

***Abstract*— Video shot boundary detection (SBD) is necessary for further video analysis like video retrieval and annotation. Great efforts have been made to develop SBD algorithms for speed and accuracy. Most works implement frame histogram as features to measure similarity for detection. However, when changes between consecutive shot boundaries are small and backgrounds of them are highly similar, most state-of-the-art methods miss these boundaries thus cannot achieve high accuracy of detection. In this paper we propose a novel SBD framework with Convolutional Neural Networks (CNNs). Firstly we adopt a candidate segment selection method to locate the positions of shot boundaries coarsely using adaptive thresholds and eliminate most non-boundary frames. Then CNN is implemented to extract representative features of frames in candidate segments. Finally cut and gradual transitions can be obtained by using a novel pattern-matching method based on a new similarity strategy. Experiments on TRECVID 2001 test data demonstrate that the proposed scheme outperforms the state-of-the-art methods and achieves high accuracy of detection.**

***Index Terms*— Shot boundary detection, adaptive thresholds, convolutional neural networks, cut transition detection, gradual transition detection, pattern matching.**

I. INTRODUCTION

With the rapid development of multimedia and network technology, digital information has an explosive growth in both quantity and quality. To better manage and analyze the large amount of data, automatic video content analysis has been urgently needed for subsequent proposes such as video retrieval and annotation. Video shot boundary detection (SBD), i.e., to segment a video sequence into several meaningful shots, is the first and fundamental process for further applications of video processing. A video shot is defined to be a sequence of images which are captured by a single camera in an uninterrupted run [1], and shot boundaries can be categorized into two types: cut transition (CT) and gradual transition (GT). In CT, the next shot appears immediately after the last frame of previous shot. While in GT, the transition in two adjacent shots usually consists of several interrelated frames and these frames change in a mild way such as fade in/out, dissolve, wipe out and other effects.

Until now great efforts have been made in SBD, and most of them focus on CT detection. Considering that two frames in CT have great discontinuities, these approaches usually extract features of consecutive frames and measure the similarity between features. Thus a CT is detected when similarity exceeds a threshold. Y. Li et al. [2] proposed an adaptive threshold employing the intensity difference between frames. Compared with CT, GT detection is more difficult because a GT usually lasts for a while and the difference between consecutive frames is less obvious. In [3], SVD was used to simplify the frame-feature matrix and an effective triangle pattern was proposed to detect GT. In [4], Tong et al. used CNN to extract semantic information and semantics were employed as auxiliary information for similarity measurement. However, in some cases when changes in GT are small and the background is similar, semantics do not change at all thus they cannot achieve high detection of accuracy.

In this paper, we propose to implement CNN to extract representative features rather than semantics. In CNNs, low-level features can be roughly learned from low-stage layers. With the network go deeper, higher-level features are obtained by composing lower-level ones [5]. Thus high-level features are powerful and suitable to distinguish transitions. Meanwhile the candidate segment selection is employed to help reduce computation. Finally we devise a novel CNN-based framework to detect shot boundaries. Experiments on standard video datasets show that proposed framework can achieve high accuracy in detections of both CT and GT.

II. SBD WITH CNN

As illustrated in Fig. 1, SBD with CNN is divided into three stages: candidate segment selection, CNN based feature extraction and SBD framework. At the beginning, to eliminate most non-boundary frames and reduce the computation complexity, we implement candidate segment selection to extract candidate segments from the entire video. Then deep features can be extracted from the fc-6 layer of CNN and used to measure the similarity between frames. Afterwards we implement proposed SBD framework to detect cut and gradual transitions respectively based on segments in different length.

A. Candidate Segment Selection

Candidate segment selection is based on the fact that consecutive frames within one shot always have high correlations [2]. In one segment, if the similarity between the first

978-1-5090-5316-2/16/$31.00 ©2016 IEEE

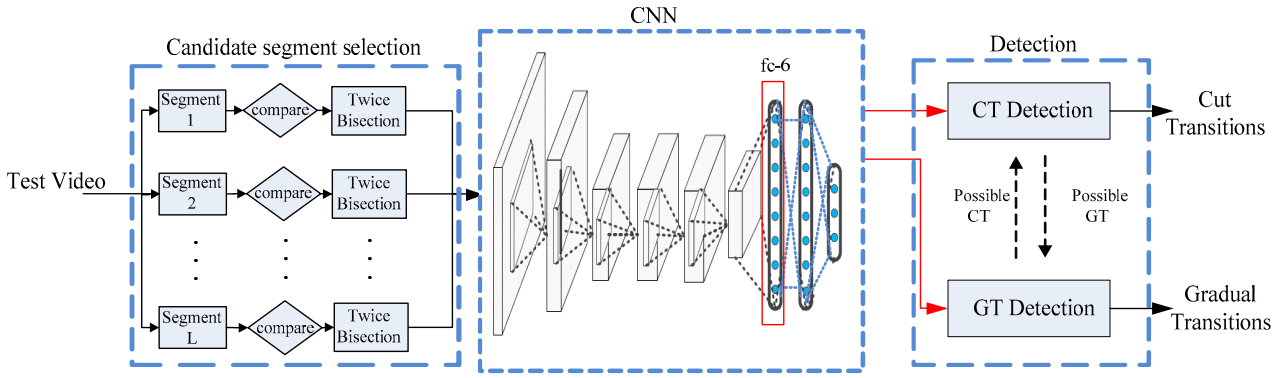

Fig. 1 The main architecture of proposed scheme

frame and the last frame exceeds the preset threshold, frames in this segment are all non-boundary frames. We adopt the pre-processing steps in [3] and basic components are shown in Fig. 1. For completeness we briefly explain it as follows:

We first cut the entire video into segments in length 21 and calculate the intensity distance between the first frame and the last frame of each segment. Ten segments are gathered into one group and local threshold T_n^L is defined as (1):

$$T_n^L = \mu_L + p\left(1+\ln\left(\frac{\mu_G}{\mu_L}\right)\right)\sigma_L \tag{1}$$

where μ_G is the global mean of all intensity distances, μ_L and σ_L are the local mean and deviation of intensity distances in one group respectively. Afterwards we compare intensity distance of each segment with local threshold to consider whether it can be a candidate segment or not. Then we implement twice bisection comparisons to locate the candidate CT segments. As a result, we get candidate CT segments in length 6 and candidate GT segments in length larger than 6.

B. CNN Based Feature Extraction

Deep features learned from CNN have been verified to be able to filter background noise and abstract representative information effectively. Thus we adopt a powerful CNN architecture which won the ILSVRC-2012 competition [6].

In this paper we train the network on ImageNet dataset with 80000 iterations. Then extract feature vectors from the sixth layer, i.e. fc-6, to measure the similarity between frames.

Cosine distance $\psi(f_i, f_j)$ is implemented to measure the similarity between frame f_i and frame f_j. It calculates the cosine angle of the feature vectors β_i and β_j of two frames as follows:

$$\psi(f_i, f_j) = \cos(\beta_i, \beta_j) = \frac{(\beta_i, \beta_j)}{\|\beta_i\| \cdot \|\beta_j\|} \tag{2}$$

It ranges from 0 to 1, which is more suitable to set thresholds. The more similar between two vectors, the distance is closer to 1. On the contrary, for vectors with more difference, the distance is closer to 0. Euclidean distance can also be used to measure similarity, but it needs normalization in further steps, which takes more computation.

There are some reasons why we do not choose features from the fifth layer (pool-5) or the seventh layer (fc-7): first, features from pool-5 are maps. Stacking these maps and stitching them to a long vector take more time and memory. Second, features from fc-7 are less representative than those from fc-6. As shown in Fig. 2, a segment containing CT is used to compare the effectiveness of features from fc-6 and fc-7. In this segment, frame 6520 belongs to a CT. From Fig. 3, it is observed that cosine distance between frame 6520 and frame 6521 using fc-6 features is approximately 0.1, which is distinctly smaller than that between other consecutive frames. However, the cosine distance has no abrupt changes when using fc-7 features. Based on discussions above, we use fc-6 features as similarity measurement.

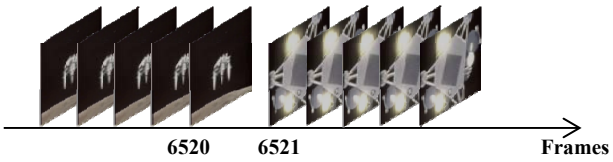

Fig. 2 A CT segment of video anni009 [7]

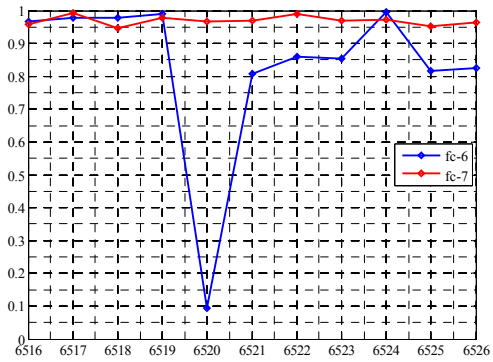

Fig. 3 Curves of cosine distance between consecutive frames in a CT segment: the blue curve and red curve denote cosine distance calculated by fc-6 and fc-7 features, respectively. Notice the obvious valley value at frame 6520 on the blue curve.

C. SBD Framework

In this section we propose a novel SBD framework with CNN, which is partially inspired by [3]. We mainly concentrate on the reduction of miss rate and the increase of accuracy, rather than the reduction of time.

1) *CT Detection*: To each candidate N-frame CT segment, feature vectors, i.e., $\{\beta_i\}$ $(i=0,...,N-1)$, are extracted from fc-6 layer of [6]. Let $\psi(t)=\psi(f_t,f_{t+1})$ denote the cosine distance between frame f_t and frame f_{t+1} and let $D_1=\psi(f_0,f_{N-1})$ denote an adaptive parameter of each segment to measure the general similarity. A CT in the t-th frame is declared if the following three criteria are all satisfied.

$$D_1 < 0.9 \tag{3}$$

$$\min\left(\psi(t)\right) < kD_1 + (1-k) \tag{4}$$

$$\max\left(\psi(t)\right) - \min\left(\psi(t)\right) > T_c \tag{5}$$

where $t=0,...,N-2$ and k is a parameter from 0 to 1.

If (3) is not satisfied, discard the segment and check the next candidate CT segment. Combination of (4) and (5) are used to measure the difference between consecutive frames as an indicator of CT. If (3), (4) and (5) are all satisfied, the t-th frame is declared as a CT. For cases that no frames satisfy (4) and (5) simultaneously, GT detection is needed because similarity between consecutive frames in a GT is always higher. Thus we add 10 frames before and after the segment respectively and consider it as a candidate GT segment.

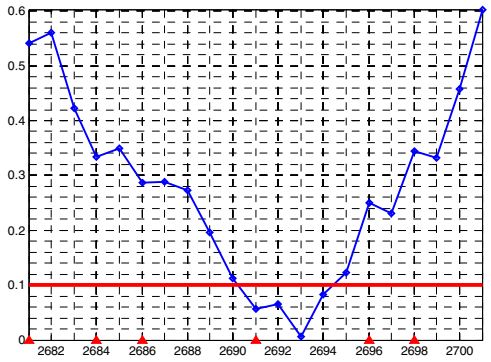

Fig. 4 Comparison of GT detection criteria between proposed scheme and [3] in a GT segment of video anni009 [7]. The blue curve denotes $diff(t)$ in the segment. The red triangles mark the abnormal frames [3]. And the red line is the threshold to determine proposed dissimilar frames.

2) *GT Detection:* GT detection is implemented to candidate segments in length larger than 6. According to [3], the distance between two consecutive frames does not show distinct characteristics in detection of gradual transitions. However, two frames over the transition segment which belong to different shots exhibit difference definitely. Thus *absolute distance difference* is defined as follows:

$$diff(t) = |\psi(f_s,f_t) - \psi(f_t,f_e)| \tag{6}$$

where f_s and f_e denote the first frame before and after the candidate segment, respectively.

While calculating $diff(t)$ from $t=0$ to $N-1$ of an ideal GT, it is found that the value approximately decreases linearly with t in the first half of the candidate segment while linearly increases with t in the second half, which illustrates that the curve of $diff(t)$ displays as an inverted isosceles triangle.

Thus detection of GT is equivalent to do pattern matching. Proposed GT detection framework is partially based on the method in [3]. The differences are that we implement a new criterion which improves the accuracy and design a more efficient detection process. In [3], one criterion is the restriction of the number of abnormal points. However, in some cases when changes through a GT are small and backgrounds are highly similar, the number of abnormal points is usually larger than the preset threshold as shown in Fig. 4, thus the GT segment is missed. During experiments we find that in most gradual transitions the proportion of frames close to the middle of a GT is limited, which can be implemented as a GT indicator. Thus we present a new criterion, i.e., the normal ratio of dissimilar frames. A dissimilar frame, which is not similar to frames before and after the segment, is close to the middle of a GT. Here we consider a frame whose $diff(t)$ is smaller than 0.1 as a dissimilar frame, and the ratio of dissimilar frames should be limited as:

$$N_d / N < T_r \tag{7}$$

where N_d denotes the number of dissimilar frames. Considering the computational complexity and the reduction of miss rate, T_r is set to 0.25 by analysing statistics from [7]. Other criteria of pattern matching are listed below:

a) Distinct peak-peak value: the difference between the maximum and the minimum values of $diff(t)$ should be distinct:

$$\max\left(diff(t)\right) - \min\left(diff(t)\right) > T_p \tag{8}$$

b) Approximate symmetry: the position of the minimum value of $diff(t)$, i.e., t^*, should be at the middle of the candidate segment, thus the bias of position should be limited.

$$(t^* - (N+1)/2) / N < T_b \tag{9}$$

For the completeness of detection, we implement the strategy of position adjustment. Concretely, if criterion (7) and (8) are satisfied while (9) is not, assume the bias between t^* and $(N+1)/2$ is M, then the candidate segment should be adjusted by M frames with its length unchanged.

Our GT detection framework can be summarized as follows:

GT Detection Framework

Input: N-frame candidate GT segments
Output: **GT** Results
Process
For each segment **do**
Extract **fc-6** feature vectors of $N+2$ frames and calculate D_1.
If $D_1 < 0.85$ is satisfied, **then**
Calculate $diff(t)$ $(t=0,...,N-1)$.
Check
If (8) is satisfied, **then**
If (7) is satisfied, **then**
If (9) is satisfied, **then**
Declare the segment as a **GT**.
Else
Adjust the position and go **Check**.
Else
Employ **CT detection** to the segment.
Else
Discard the segment.
Else
Discard the segment.
End for
Merge the GT segments that overlap with each other.

TABLE II
PERFORMANCE COMPARISON OF CT AND GT DETECTION

Transitions	CT									GT								
Videos	Recall			Precision			F1			Recall			Precision			F1		
	[3]	[4]	Proposed	[3]	[4]	Proposed	[3]	[4]	Proposed	[3]	[4]	Proposed	[3]	[4]	Proposed	[3]	[4]	Proposed
anni001	—	—	—	—	—	—	—	—	—	0.889	**1.000**	**1.000**	0.800	**1.000**	**1.000**	0.842	**1.000**	**1.000**
anni005	0.974	0.895	**1.000**	0.881	**1.000**	**1.000**	0.925	0.932	**1.000**	**0.963**	0.889	0.826	0.426	0.857	**0.974**	0.591	0.873	**0.894**
anni007	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.455	**1.000**	**1.000**	0.625	**1.000**	**1.000**
anni008	**1.000**	**1.000**	**1.000**	0.667	**1.000**	**1.000**	0.800	**1.000**	**1.000**	0.833	0.917	**0.929**	0.500	0.917	**0.929**	0.625	0.917	**0.929**
anni009	0.737	0.821	**0.935**	0.875	**1.000**	0.967	0.800	0.901	**0.951**	0.797	0.734	**0.917**	0.662	0.940	**1.000**	0.723	0.825	**0.956**
BOR10-001	—	—	—	—	—	—	—	—	—	**1.000**	0.909	**1.000**	0.612	0.909	**1.000**	0.742	0.909	**1.000**
BOR10-002	—	—	—	—	—	—	—	—	—	0.800	0.800	**1.000**	0.706	0.842	**1.000**	0.828	0.821	**1.000**
Average	0.929	0.929	**0.984**	0.856	**1.000**	0.992	0.891	0.958	**0.988**	0.896	0.893	**0.953**	0.594	0.924	**0.986**	0.810	0.906	**0.968**

TABLE I
DETAILS OF TEST VIDEOS

Videos	Frames	Transitions			Sources
		Total	CT	GT	
anni001	914	8	0	8	7 documentaries from TRECVID 2001 test data[7]
anni005	11363	84	38	46	
anni007	1590	11	5	6	
anni008	2775	14	2	12	
anni009	12306	103	31	72	
BOR10- 001	1815	11	0	11	
BOR10- 002	1795	10	0	10	
Total	32558	241	76	165	

III. EXPERIMENT RESULTS

To evaluate the performance of our scheme, we compare our SBD framework with state-of-the-art methods including the SVD based pattern matching method [3] and the semantics based method [4] on standard test videos as listed in Table I.

A. Evaluation Metrics and Parameters Selection

Similar to SBD schemes in most papers, we use recall, precision and F_1 as evaluation metrics as follows:

$$\text{recall} = \frac{N_C}{N_C + N_M} \tag{10}$$

$$\text{precision} = \frac{N_C}{N_C + N_F} \tag{11}$$

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{12}$$

where N_C , N_M , and N_F respectively denote the number of shot boundaries detected correctly, missed and falsely. F_1 value is a general measurement considering both recall and precision, and higher value means better performance.

In our scheme, T_r is a parameter as a GT indicator and it is set to 0.25. The remaining parameters are set by experiments while considering both speed and accuracy as follows:
$p = 0.7, k = 0.55, T_c = 0.6, T_p = 0.25, T_b = 0.2$.

B. Comparisons and Analysis

The recall, precision and F_1 measure for SBD detection with [3], [4] and proposed scheme are listed in Table II. It is observed that with proposed scheme the mean value of F_1 in CT detection and GT detection are 0.988 and 0.968, respectively. Results demonstrate that our SBD framework outperforms state-of-the-art methods.

Concretely, there are several gradual transitions in video anni009 where changes are small and backgrounds are almost the same. Feature vectors used in [3] and semantic labels extracted from CNN in [4] do not change in highly similar cases thus cannot help locate gradual transitions accurately. Our scheme incorporates powerful fc-6 features extracted from CNN and a novel GT detection framework to achieve a high F_1 value as 0.917, which is superior to 0.732 with [3] and 0.825 with [4].

IV. CONCLUSION

We present a novel SBD framework based on representative features extracted from CNN in this paper. The proposed scheme is suitable for detection of both CT and GT boundaries. Experiments show that it outperforms the state-of-the-art methods and achieves excellent accuracy.

ACKNOWLDGMENT

This work was supported by NSFC (61521062, 61527804), and Shanghai Zhangjiang national independent innovation demonstration zone development fund (201501-PD-SB-B201-001).

REFERENCES

[1] C. Cotsaces, N. Nikiolaidis, and I. Pitas, "Video shot detection and condensed reprsentation. A review," *IEEE Signal Process. Mag*, Vol. 23, no. 2, pp. 28-37, Mar. 2006.

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

[3] Z. Lu and Y. Shi, "Fast video shot boundary detection based on SVD and pattern matching," *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 5136-5145, Dec. 2013.

[4] W. Tong, L. Song, X. Yang, H. Qu and R. Xie, "CNN-based shot boundary detection and video annotation," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Ghent, 2015, pp. 1-5.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, May. 2015.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, Harrahs and Harveys, 2012, pp. 1097-1105.

[7] (2001) TREC video retrieval test collection. [Online] Available: http://www.open-video.org/collection detail.php?cid=7.