VIDEO BREAK DETECTION BASED ON SIMILARITY ANALYSIS AND TEMPORAL STATISTICAL CHARACTERISTICS

Jianjun Huang

School of Mathematics and Computer Science, Fuzhou University, Fujian, P.R. China
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua University，Beijing, P.R. China

hngjnjn@163.com

ABSTRACT

Video Shot boundary detection is a fundamental task in any kind of video content manipulation and retrieval. The area of shot boundary detection has been extensively studied, but achieving highly accurate detection results remains a challenge. In this paper, we present a novel algorithm for video cut detection. The algorithm is implemented and evaluated on the TRECVID benchmark platform. The experimental results show the effectiveness of the proposed approach. The algorithm provides a clue to the gradual transition detection.

1. INTRODUCTION

The widespread availability and usage of digital video have created a need for automated video content analysis. Partitioning a video sequence into shots is the first step toward video content analysis, browse and retrieval.

To date, numerous techniques have been proposed for shot boundary detection [9]-[10], etc. Hanjalic [5], U. Gargi [6] and Lienhart [7]-[8] have provided comprehensive and comparative surveys on the most representative approaches.

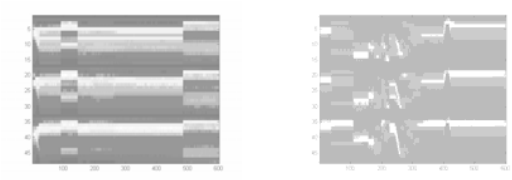
There are some algorithms in existence for shot boundary detection based on similarity analysis [1]-[4]. The common approach consists of three major steps: to construct a similarity matrix first, then calculate a frame-indexed score by traversing the frame series with a template of varying width under multi-resolutions, and finally submit the score to a learning machine, or to an adaptive or global threshold judgment. These approaches are usually expensive in calculation, and leave room for improvements of results.

In this paper, we follow a different approach. Additional step is added to the common process before working with both global and adaptive thresholds. Possible cut candidates are chosen first with a view of the statistical detection theory. The selection of cuts candidates is vital, because it not only reduces the total amount of computation, but also contributes greatly to the final outcomes. These candidates are the centers of calculation in the next filtering stage. False cuts are sifted by calculating a score which carries magnified information about the data distribution difference between the real cut with false cut. This amplification is done by making use of a similarity matrix. The way we construct the similarity matrix is also a little different. The similarity matrix we constructed is fit to represent the similarity between frames. The algorithm proposed is implemented, tested by extensive experiments and evaluated on TRECVID platform. The experimental results outperform many others.

The paper is organized as follows. In section 2, our approach to the shot boundary detection problem is presented. In section 3, the experimental results are provided. We conclude this paper with a discussion in section 4.

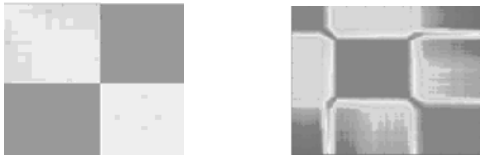
2. DETECTION ALGORITHM

The main idea of our approach for shot boundary detection is to find the possible cut candidates first, and then sift the false shot breaks. From the statistical point of view, the number of cut shot transitions per hundred frames couldn't be too big, usually no more than two. In view of disturbances, we set the cut candidate quotas equal five (i.e., the number of abrupt transition candidates we pick up is five per hundred frames). In a way, this simple technique reduces computation time, and increases precision rate without at a cost of reducing the recall rate. There are several different ways to calculate the similarity matrix S. The cosine distance and squared Euclidean vector distance are used to measure similarity for each pair of video frames [3], [4] respectively. Histogram inter-section method is adopted in [1], [2]. We take a different measurement.



(a) (b)

Figure 1: (a) Abrupt transitions; (b) Dissolve & FOI transitions



(a) (b)

Figure 2: Similarity diagrams of shot boundary transitions. (a): CUT transition. (b): Dissolve transitions.

2.1 Similarity matrix and its computation

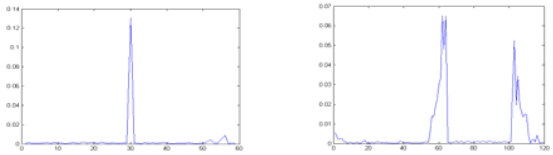
Low-level features are calculated to represent each frame first. Throughout this paper, we extract a global RGB-48 histogram (16 bins for each channel of the RGB color space), although histograms in other color spaces, and other global or block-wise low-level features also serve the purpose well. Let X = {Xi | i=1,2,…,N } denote the feature matrix where Xi is a frame-indexed feature vector (a column vector of 48 dimensions in this case, because we extract a global RGB-48 histogram feature), and N the number of frames of a given video. The X, as a feature matrix, contains ample information about shot boundary transition. For example, Figure 1 (a) corresponding to the feature sub-matrix from the video "19981004_ABCa.mpg" shows the presence of typical cut transitions. From Figure 2 (b), the presence of gradual transitions can be easily spotted.

In this paper, we define the similarity matrix S=(*Sij*) as a two-dimensional array with size R×R (R is a positive number) that satisfies

$$s_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}} \qquad (1)$$

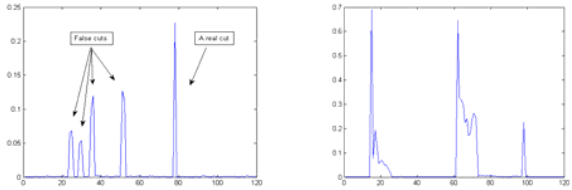
where *Cij*=E[(Xi- μ i) (Xj- μ j)], E is the mathematical expectation and μ j=E(Xj), for i,j=1,2,...,R. Such-defined similarity matrix is well fit in describing the similarity between frames. The similarity matrix contains almost as much shot transition information as the feature matrix. This is illustrated in Figure 2 which shows the typical cut and gradual transition appearances. Moreover, the similarity matrix S is symmetric, and has maximum correlation value (which equals one) along the main diagonal where each frame is compared to itself.

2.2. The selection of cut boundary candidates



(a) (b)

Figure 3: The curves show the presence of shot boundary. (a): A cut. (b): Gradual transitions.



(a) (b)

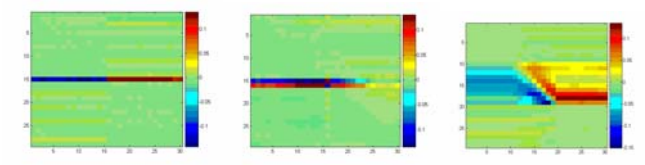
Figure 4: Peaks caused by false cuts. (a): Four are flashlights and the one on right is a real cut. (b): Some OTHs exhibit similar peaks.

To pick up possible candidates, we adopt the first diagonal elements above the main diagonal of the similarity matrix as the score of the corresponding frame. The score represents the correlation of a frame with its next neighbor. It is independent of the position and the number of the frames we choose to compute the correlation. Thus, the cut candidate determination doesn't involve much computation. In practice, we usually take 100 frames and computer their similarities each time. The cut break candidates can be easily picked up by means of adaptive thresholds, since an abrupt transition produces an isolated high peak and we have the quotas of five cut candidates per hundred. In the implementation, we combine a heuristically chosen global threshold (0.01) with an adaptive threshold. A cut candidate can only be chosen when its value (the approximate derivative or the difference of the correlation) exceeds the global threshold and more than three times of the values of its direct neighboring frames.

2.3 The cut boundary determination

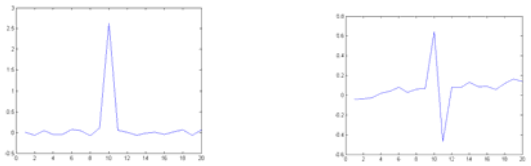
The false cuts (disturbances caused by fast motions of object and flashlights) and some OTHs have to be filtered, because they produce similar curves. Figure 4 shows peaks caused by false shot breaks. Some OTHs do involve instantaneous shot change from one black screen to a normal scene, or vice verse. To deal with disturbances caused by fast motions of object and flashlights, we employ a technique computing the difference of the similarity matrix. By the difference of matrix, we mean a matrix of row differences. If M= (m_{ij}) is a K×K matrix, then the difference of M is a (K−1)×K matrix D= (d_{ij}) satisfying:

$$d_{ij} = m_{(i+1),j} - m_{i,j} \qquad (2)$$



(a) (b) (c)

Figure 5: Different data distribution patterns. (a): Left half elements of a cut frame are negative; the right half elements are positive. (b) A data pattern of flashlight. (c): The data pattern of a gradual transition (negative in certain low triangular positions).



(a) (b)

Figure 6: The difference between real and false cuts is amplified in one dimension by making use of the frame score.
(a): A real cut curve. (b): A false cut curve.

for each i=1,2,…,(k-1), j=1,2,…,k. We define a new frame-indexed score as the subtraction of the sum of the up-triangular corresponding row elements and the sum of the low-triangular corresponding row elements of the difference matrix. To make it more clearly, suppose D= (d_{ij}) is the difference of the similarity matrix, then the frame-indexed score is calculated by

$$framescore(i) = \sum_{j=i+1}^{k} d_{ij} - \sum_{j=1}^{i-1} d_{ij} \qquad (3)$$

where k is the second dimension of D (the column number).The motivation of the above formula comes from the observation that there is a big data distribution pattern difference between the real cut transition and the one caused by disturbances. Figure 5 illustrates different data distribution patterns that go with the occurrences of a real abrupt transition, a false transition and a gradual transition, respectively. Notice that the right half elements of the center row (a cut boundary) are positive, and the left half elements are negative (Figure 5(a)). The different data patterns can also be observed from other two pictures.

By triangular matrix subtraction, the difference between a real cut and a false cut is greatly magnified, and thus easily recognized in one dimension, as illustrated in Figure 6. The frame score is the result of the magnification, which makes it easier to filter false cuts by working with both global and adaptive thresholds. These thresholds are heuristically chosen. Actually, we set the global threshold equal 0.24 when radius r=10, and 0.14 when r=5. We employ multi-scale sifting scheme. A cut candidate can

Recall	0.968	0.933	0.897	0.930	0.953	0.860
Precision	0.951	0.945	0.923	0.975	0.972	0.937
F-Score	0.959	0.939	0.910	0.952	0.962	0.897

Recall	0.970	0.947	0.911	0.929	0.950	0.959
Precision	0.953	0.928	0.964	0.916	0.884	0.917
F-Score	0.961	0.937	0.937	0.923	0.916	0.938

Table 1:The experimental results on TRECVID 2004 data set (12 video clips).

only be chosen when its value (the framed score defined above) exceeds the global threshold and three times of the values of its direct neighboring frames and ten times of mean values of its left and right neighboring frames.

There is another problem need to be tackled. Some OTHs that involve an abrupt transition from a black screen to a normal scene, or the vice verse, could be easily go unidentified with cuts. Therefore, a black screen detector and OTH/FOI filters have to be designed to discriminate the cut shot boundary from these OTHs. The employment of these special filters contributes to a reduction of false detection rate. In short, our algorithm for the cut transition detection consists of the following steps:

1. Extract certain features from a video data set, and construct a feature matrix;
2. Pick up cut candidates. This can be done by choosing the first diagonal elements of the similarity matrix as a score before working with thresholds and with statistical quotas. The similarity matrix is obtained by computing normalized correlation coefficients of the feature matrix.
3. Filter false cuts by calculating the subtraction of the sum of the up-triangular and that of the low-triangular of the difference matrix of the similarity matrix (Formula (3)).
4. Design and incorporate special detectors (such as a monochrome, screen-in-screen and OTH detectors) into the framework to reduce false detection rate.

3. EXPERIMENTAL RESULTS

The algorithm proposed above is implemented and tested by extensive experiments. For testing, we use the TRECVID 2004 data corpus and its revaluation tools. The data for TRECVID 2004 shot boundary detection consists of 12 randomly selected 30-minute videos with 618,409 frames and 4,806 transitions including 2774 cut transitions. The performance of an algorithm is evaluated by recall, precision, and F-score which are common in information retrieval.

Our experiments were carried out mainly on the 19981004_ABCa.mpg file and tested on all the 12 video clips. The results are showed in Table 1. The recall × precision results of cut detection from TRECVID-2004 17 participants, together with our results, are displayed in Figure 7.

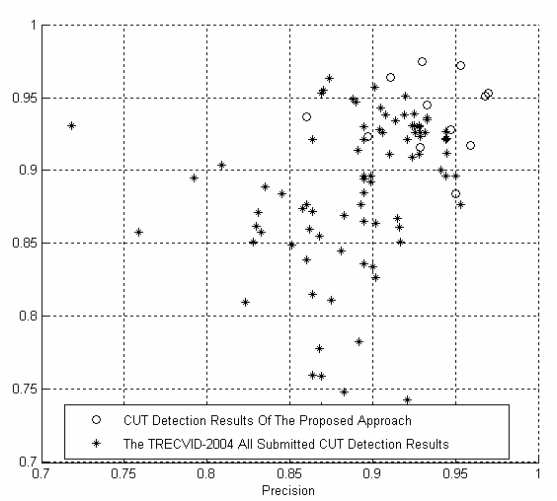


Figure 7: Our experimental results (marked in circle) and the results of participants of TRECVID-2004(in star).

The performance on some video clips suffered somewhat due to the absence of a screen-in-screen detector and a general monochrome detector. The false detection rate could have been reduced further if other detectors (such as a picture-in-picture detector) had been incorporated into the system. In spite of this insufficiency, our results are still very competitive when compared with the results from TRECVID 2004.

4. CONCLUSION AND FUTURE WORK

In this paper, we present a novel algorithm for shot boundary detection, focusing primarily on the identification of abrupt transitions or cuts. A pre-processing step is added to the common procedure of the approaches based on similarity analysis for shot boundary detection. The similarity between frames is measured in a new way. A novel frame-indexed score, calculated by the triangular subtraction method, is proposed and applied in false cut sifting. The new frame-indexed score shows promise for application in the gradual transition detection (Figure 5 (c)). The proposed algorithm is implemented and evaluated on TRECVID benchmark platform. Compared with results reported by others, our results are among the best. The evaluation demonstrates the effectiveness of the proposed approach on the TRECVID-2004 data collection. To some extent, the algorithm is robust because it handles disturbances (flashlights, object motions) within a shot. Moreover, the approach is computationally lightweight. The novelty scores do not involve much computation. Compared with other methods, our algorithm is also competitive computationally.

There are still some problems that ask for future work. For instance, we didn't incorporate general monochrome screen detector and picture-in-picture detector in the framework. As a result, our results are somewhat suffered. The usage of other features or block-wise features leaves room for further improvements. Overall, we believe that the proposed algorithm is effective for abrupt boundary detection, and can throw light on the investigation of the gradual shot boundary detection.

5. ACKNOWLEGEMENT

The author is grateful to the members of the research group in the State Key Laboratory of Intelligent Technology and Systems, Tsinghua University. Among them are Bo Zhang, Jinhui Yuan, Fuzhong Lin, and Jianmin Li. They offered me a lot of help and very useful discussions during my visit to Tsinghua. Without their assistances, my work presented here wouldn't be possible.

6. REFERENCES

[1] J. Yuan, J. Li, etc, "A Unified Shot Boundary Detection Framework Based on Graph Partition Model", In *ACM Multimedia*, October, 2005

[2] J. Yuan, B. Zhang, and F. Lin, "Graph partition model for robust temporal data segmentation", In *PAKDD*, pp.758-763, 2005.

[3] M. Cooper, "Video segmentation combining similarity analysis and classification," In *ACM Multimedia*, October, 2004.

[4] Cooper, M., Foote, J., Adcock, J. & Casi, S., "Shot boundary detection via similarity analysis", In *Proceedings of the TRECVID 2003 Workshop*, Gaithersburg, Maryland, USA, pp. 79-84,2003.

[5] Alan Hanjalic, Shot-Boundary Detection: Unraveled and Resolved ?, In *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, NO.2, pages 90-104, February 2002.

[6] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods", In *IEEE Transaction on Circuits and Systems for Video Technology*, 10(1), February, 2000.

[7] R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *International Journal of Image and Graphics* 1(3), pp. 469-486, 2001.

[8] R. Lienhart, "Comparison of automatic shot boundary detection algorithms", In *Image and Video Processing,* VII, 3656, pp. 290-301, Proc. SPIE, Jan. 1999.

[9] Ramin Zabih, Justin Miller, Kevin Mai, "A feature-based algorithm for detecting and classifying production effects", *Multimedia Systems*, pp 119-128, 1999.

[10] C.W. Ngo, T.C. Pong, and R. T. Chin, "Camera breaks detection by partitioning of 2D spatio-temporal images in MPEG domain", In *Proc. IEEE Intl. Conf. Multimedia Computing and Systems*, vol.1, pp. 750-755, 1999.