

An Integrated Method for Video Shot Boundary Detection

Lei Zhu, Junfeng Qu, Muhammad Asadur Rahman, and Weihong

*College of Information and Mathematical Sciences, Clayton State University
2000 Clayton State Blvd, Morrow, GA 30260, U.S.A.*

{LeiZhu, jqu, mrahman, Weihong}@mail.clayton.edu

Abstract— Video shot boundary detection, which segments a video by detecting boundaries between camera shots, is usually the first and important step for content-based video retrieval. This paper investigates methods which are effective in detecting abrupt transitions and gradual transitions, respectively, and proposes an integration scheme to combine their results, aiming to detect both types of transitions.

Keywords—video shot boundary detection, abrupt transitions, gradual transitions, cuts, dissolves, fades, wipes

I. INTRODUCTION

With the rapid advance of multimedia and Web technologies, video data in various formats are becoming available at an explosive rate. For example, based on a Yahoo! Answers post dated on June 2009 [1], there were over 240,000,000 videos on YouTube, which is the most popular online video sharing Website. The time required to view all these videos was over 800 years. More amazingly, around 500,000 new videos were uploaded to YouTube everyday!

With such enormous video data resources, sophisticated video database systems are highly demanded to enable efficient browsing, searching and retrieval. However, the traditional video indexing method, which uses human beings to manually annotate or tag videos with text keywords, is time-consuming, lacks the speed of automation and is hindered by too much human subjectivity. Therefore, more advanced approaches such as content-based video retrieval are needed to support automatic indexing and retrieval directly based on videos' content, which provide efficient search with satisfactory responses to the scenes and objects that the user seeks.

Video shot boundary detection, which segments a video by detecting boundaries between camera shots, is usually the first and important step for content-based video retrieval. A video consists of a sequence of images (often being called frames), which can be played consecutively at the speed of around 20 to 30 frames per second in order to view smooth motion. To index and retrieval a video, shot boundary detection is usually conducted to segments the video into shots by detecting boundaries between camera shots.

As illustrated in Figure 1, a shot, the basic syntactic unit of a video, may be defined as a sequence of frames captured by "a single camera in a single continuous action in time and space" [2]. For example, a video sequence showing a soccer game may be composed of a number of interleaved shots

which show players playing the ball inside the soccer field, as well as the shots showing the re-actions of the audience in the stadium. A scene is a logical grouping of shots into a semantic unit which focuses on a certain object or objects of interest.

There are a number of different types of transitions or boundaries between shots. The simplest transition is a cut: an abrupt transition that occurs between two adjacent frames. More sophisticated transitions are gradual transitions which include dissolves, fades and wipes, etc. A fade is a gradual change in brightness, either starting or ending with a black frame. A dissolve is similar to a fade except that it occurs between two shots. The frames of the first shot get dimmer and those of the second shot get brighter until the second replaces the first. So fades can be regarded as special cases of dissolves. Other types of shot transitions include wipes and computer generated effects such as morphing. A robust segmentation algorithm should be able to detect all of these different boundaries with a high degree of accuracy.

Although there are many methods of detecting shot boundaries, relatively fewer methods focus on detecting both abrupt and gradual transitions. This paper investigates methods which are effective in detecting abrupt transitions and gradual transitions, respectively, and proposes an integration scheme to combine their results, aiming to detect both types of transitions. The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents the integration method which detects both abrupt transitions and gradual transitions. In Section 4, experimental results are presented to demonstrate the effectiveness of our approach. The conclusion and future work are provided in Section 5.

II. RELATED WORK

The detection of a cut is based on the fact that consecutive frames on either side of a shot boundary generally display an abrupt change in content. Therefore, the general method is to use a quantitative measure such as color histogram or edge change of frames to represent the content of frames, and then calculate the content difference between such a pair of frames. If the difference exceeds a given threshold, it indicates a shot boundary. In [2], Nagasaka and Tanaka compared several statistical techniques using grey level and color histograms. Zhang et al [3] used a running histograms method to detect gradual as well as abrupt shot boundaries. An approach which can decrease false/miss caused by fast object motion and abrupt screen flash was proposed in [11]. In [4], after comparing several different methods of shot boundary

detection using a variety of video content types, Borezky and Rowe concluded that histogram-based methods were more successful than others, but that shot boundary detection thresholds must be guided by the target application. To address the difficulties of setting up thresholds, authors of [5] proposed several methods to detect cuts using adaptive thresholding. Our method for cut detection is mainly based on the above work, using color histograms and adaptive threshold as the first round of detection, which generates a number of shot boundary candidates.

It is more challenging to detect gradual transitions because gradual transitions usually stretch over a number of adjacent frames of two shots. Lo and Wang [6] used clustering algorithms to monitor frame similarity. When certain frames are identified as belonging to a scene change, adjacent frames are marked as gradual transitions while the remaining frames are detected as cuts. In [7], methods for detecting shot boundaries in video sequences and for extracting key frames using metrics based on information theory are proposed. The method for shot boundary detection relies on the mutual information (MI) and the joint entropy (JE) between the frames. It can detect cuts and fades. Another algorithm based on a sliding window Singular Value Decomposition approach is proposed in [10]. In [8], a moving query window of frames is maintained and the average frame similarities of the frames of the left side and right side of the center frame are calculated, respectively. Then the ratio of these two similarities are being monitored and used to detect gradual transitions. We use a similar method proposed in [8] as the second round of detection, and combine the results of both rounds to determine the boundaries' location and types.

III. METHODS

Our methods maintain a sliding window which consists of a predefined number $(2m+1)$ of frames, where m is the half-window size and f_c is the center frame (located in the middle of the window). Figure 2 shows an example of a sliding window with 5 frames where $m=2$. When the center frame f_c is identified as a shot boundary, the window will slide m frames to the right (which means no new decision will be made until half-window size number of frames have elapsed); otherwise, the window will slide one frame to the right. For each frame f_i in the sliding window, color histogram h_i based on RGB or HSV space is calculated and is used as the feature vector of the frame.

Method 1: Shot Boundary Detection Based on Hard Threshold

For each frame f_i in the sliding window, we calculate $d_i = \|h_i - h_{i-1}\| / \|1 + h_i + h_{i-1}\|$, which is the dissimilarity of histogram with the previous frame (the immediately adjacent frame at the left side) f_{i-1} . If $d_i > Tb$, where Tb is the cut off threshold, then f_i is marked as a shot boundary. This detection method is based on the fact that consecutive frames on either side of a shot boundary (especially cut) generally display a significant change in content. Therefore, a significant change of dissimilarity of histogram of a frame may indicate a shot boundary (more likely to be cuts). As illustrated in Figure 3,

six cuts are detected when $Tb = 300$. However, the major drawback of this method is that it is very difficult to detect the majority of shot boundaries for different videos by using a single cut off threshold, even for videos of the same content type. Thus an adaptive threshold is more appropriate.

Method 2: Shot Boundary Detection Based on Adaptive Threshold

For each frame f_i in the sliding window, we calculate $d_i = \|h_i - h_{i-1}\| / (1 + h_i + h_{i-1})$, which is the dissimilarity of histogram with the previous frame f_{i-1} . For all d_i in the sliding window, calculate the mean μ and standard deviation σ . Suppose d_c is the dissimilarity of histogram of the center frame f_c with the previous frame f_{c-1} , if d_c is the maximum value in the window, and it is greater than the threshold $\mu + Td * \sigma$, where Td is an empirical tuning parameter, then f_c is marked as a shot boundary. In this way, the threshold is computed dynamically and statistically when the window is sliding, based on all the dissimilarities of the frames in the local window. If the center frame's dissimilarity of histogram d_c reaches the local maximum value and exceeds this adaptive threshold, then it may indicate a shot boundary. Therefore, this method is more adaptive and effective in detecting shot boundaries, especially for cuts.

Method 3: Shot Boundary Detection Based on Ratio of Average Frame Dissimilarity

Because gradual transitions are more sophisticated and subtle, usually stretching over a number of adjacent frames of two shots, methods effective for cut detection may not be effective in detecting gradual transitions. We use the similar method originally presented in [8] which is relatively effective in detecting dissolves. In this method, for the center frame f_c , the average histogram dissimilarity of the frames in the left side of the window (Pre-dissimilarity) and the average histogram dissimilarity of the frames in the right side of the window (Post-dissimilarity) are calculated, respectively. Then the ratio of these two dissimilarities (Pre/Post ratio) is calculated and monitored. Figure 4 describes an example of dissolve where the Pre/Post ratio shows certain pattern of changing before, in, and after the dissolve. As stated in [8], cuts usually are indicated by peaks in the Pre/Post ratio, with steep slopes. Dissolves are also indicated by peaks in the pre/post ratio, usually at the end of the transition. In addition, the slopes of these peaks are often moderately steep, as opposed to the very quick rise found for cuts.

Based on these observations, when the center frame f_c 's Pre/Post ratio reaches the local maximum and exceeds a threshold T_h (e.g., 1.3), we first check if there is a big Pre/Post ratio drop T_r (e.g., 70%) for the frame f_{c-1} which is the immediately adjacent frame at the left side of the center frame f_c . If there is such a big drop, then f_{c-1} may indicate a shot boundary for cut. Otherwise, we check the Pre/Post ratios of the past frames in the sliding window and look for local minimum. If the local minimum value is achieved at frame f_s which is least 6 frames before the center frame f_c , and its Pre/Post ratio drops significantly and below a threshold T_l (e.g., 0.7), then f_s may indicated the start of the dissolve.

Method 4: Integrated Two Round Shot Boundary Detection

Aiming to detect both types of transitions, we propose an integration scheme: first, we use method 2 and method 3 to conduct detections independently, in two rounds, which generate two lists of shot boundary candidates: L2 (mainly cuts) and L3 (cuts and gradual transitions). Secondly, in the result merge process, we start from L2, and then check each candidate in L3: If the distance of a boundary in L2 and a cut boundary in L3 is within a threshold (T_c) number (e.g., 3) of frames, we keep the boundary in L2 and remove the boundary in L3. If the distance of a boundary in L2 and a gradual boundary in L3 is within a threshold (T_g) number (e.g., 20) of frames, we keep the gradual boundary in L3 and remove the boundary in L2. And finally we merge L2 and L3 to get the final boundary list.

IV. EXPERIMENTS

The experiments are conducted on a Dell Inspiron E1705 Laptop with Intel Centrino Duo Core 1.66G Hz processor, 1.5 GB memory and 70GB harddisk, running on Windows XP Professional operating system. The C/C++ programs are developed using Visual Studio 2005 with FFmpeg library [9] for converting and decoding video streams.

The dataset consists of 19 videos downloaded from Youtube, with total length of 66 minutes. These 320x240 videos have 30 frames per second. Although the content of these videos are diversified, we roughly categorized them into six content types: TV News, Situation Comedy, Sports Game, Music Video, Lip Synchronization, and User Generated Video. Each type has three or four videos. We have viewed these videos frame by frame, and manually marked types and positions of shot boundaries which will be used for evaluation purpose. Table 1 lists number of frames, number of cuts, number of gradual transitions, and the ratio of gradual transitions to cuts, by content types. Note that videos of Sports Game and Music Video have relatively more gradual transitions and higher ratio of gradual transitions to cuts, which makes the task of shot boundary detection more difficult. Observing that around 90% (216 out of 241) gradual transitions are dissolves and fades which are special cases of dissolves, we focus on detecting dissolves for gradual transitions in this dataset.

We use recall and precision to evaluate system performance. Recall is the ratio of shot boundaries correctly identified by the system to the total number of shot boundaries existing in the dataset. Precision is the ratio of correct shot boundaries identified by the system to the total number of shot boundaries identified by the system. Ideally, both recall and precision should equal 1 which indicates that the system has identified all existing shot boundaries correctly, without identifying any false boundaries. In this research, we are more concerned about recall because content-based video retrieval applications generally prefer high recall if high precision and high recall can't be achieved at the same time.

In the experiments, we observe that HSV color space achieved slightly better performance than RGB color space. Therefore, we only report the performance based on 8x8x8

HSV space in this paper. Major parameters and thresholds we use are: For Method 2, the half-window size $m=12$, $T_d=3$; for Method 3, $m=20$, $T_h=1.3$, $T_l=0.70$, $T_r=70\%$; for Method 4, $T_c=3$, $T_g=20$.

Table 2 lists the precision and recall achieved by different methods, when all types of transitions are considered. Compared with Method 3, Method 2 achieves higher precision and recall in the whole dataset as well as most content types. The precision values of Method 3 are consistently lower, which indicates that Method 3 tends to generate more false boundaries. With the result merge process proposed in Method 4, recall improved around 1 to 3 percent with some sacrifice of precision, as illustrated by Figure 5.

Table 3 further lists the detection ratios of cuts (recall for cuts) and the detection ratios of gradual transitions (recall for gradual transitions) achieved by different methods. The detection ratios of cuts by Method 2 are higher than those of Method 3 consistently, which indicates method 2 may be more reliable in detecting cuts for this dataset. On the other hand, the detection ratios of gradual transitions by Method 3 are higher than those of Method 2 in the whole dataset as well as most content types, even for videos of Sports Game and Music Video which have relatively more gradual transitions and higher ratio of gradual transitions to cuts. This suggests that method 3 might be more reliable in detecting gradual transitions for this dataset. Method 4 combines some strength of Method 2 and Method 3, and improves the detection ratios of cuts and gradual transitions, which indicates the effectiveness of the proposed integration scheme.

TABLE 1: DATASET DESCRIPTION

Content Type	# of Frames	# of Cuts	# of Gradual Transitions	Gradual/Cuts
TV News	25,923	91	21	0.23
Situation Comedy	19,439	103	27	0.26
Sports Game	20,711	90	66	0.73
Music Video	23,411	352	111	0.32
Lip Sync	12,178	17	2	0.12
User Generated	14,881	65	14	0.22
ALL	116,543	718	241	0.34

V. CONCLUSIONS AND FUTURE WORK

In this paper, an integration method for video shot boundary detection is presented. By combining boundary detection based on adaptive threshold and ratio of average frame dissimilarity, the integration scheme has improved the detection ratio of cuts and gradual transitions effectively. Currently we are improving our algorithms (especially those in Method 3 and Method 4) and conducting extensive experiments to achieve better performance. In the future, we will investigate more detection methods, with larger datasets and more video content types.

REFERENCES

- [1] Roughly how many videos are found on youtube?. [Online]. Available: <http://answers.yahoo.com/question/index?qid=20090523195808AAFd4pe>. Yahoo Answer, June 2009.
- [2] A. Nagasaka and Y. Tanaka, *Automatic video indexing and full-video search for object appearances*, in Visual Database Systems II, Elsevier Science Publishers, pages 113-117, 1992.
- [3] H. J. Zhang, A. Kankanhalli and S. W. Smoliar, *Automatic partitioning of full-motion video*, in Multimedia Systems, volume 1, pages 10-28, 1993.
- [4] J. Boreczky and L.A. Rowe, *Comparison of video shot boundary detection techniques*, in IS&T/SPIE proceedings: Storage and Retrieval for Images and Video Databases IV, volume 2670, pages 170-179, February 1996.
- [5] Y. Yusoff, W. Christmas, and J. Kittler, *Video Shot Cut Detection Using Adaptive Thresholding*, in Proceedings of The Eleventh British Machine Vision Conference (BMVC 2000), University of Bristol, September 11-14, 2000.
- [6] C. Lo and S. Wang, *Video segmentation using a histogram-based fuzzy C-Means clustering algorithm*, in Journal of Computer Standards and Interfaces, 23:429-438, 2001.
- [7] Z. Černeková, I. Pitas, S. Member and C. Member, *Information theory-based shot cut/fade detection and video summarization*, in IEEE Transactions on Circuits and Systems for Video Technology, volume 16, pages 82-91, 2006.
- [8] T. Volkmer, S. Tahaghoghi and H. Williams, *Gradual Transition Detection Using Average Frame Similarity*, in Proceedings of Multimedia Data and Document Engineering 2004, Washington, DC, USA, July 2, 2004.
- [9] FFmpeg. [Online]. Available: <http://ffmpeg.org>. November 2009.
- [10] Wael Abd-Almageed, *Online, Simultaneous Shot Boundary Detection and Key Frame Extraction for Sports Videos Using Rank Tracing*, in Proceedings of 15th IEEE International Conference on Image Processing, San Diego, CA, October 12-15, 2008.
- [11] Shien-Tang Chiu, Guo-Shiang Lin, Min-Kuan Chang, *An Effective Shot Boundary Detection Algorithm for Movies and Sports*, in Proceedings of 3rd International Conference on Innovative Computing Information and Control, 2008.

TABLE 2: PERFORMANCE COMPARISON

Content Type	Method 2		Method 3		Method 4	
	Pre	Rec	Pre	Rec	Pre	Rec
TV News	0.29	0.80	0.16	0.82	0.27	0.82
Situation Comedy	0.51	0.82	0.26	0.81	0.45	0.83
Sports Game	0.33	0.47	0.28	0.45	0.30	0.48
Music Video	0.84	0.58	0.65	0.48	0.81	0.60
Lip Sync	0.18	0.89	0.10	0.95	0.15	0.92
User Generated	0.40	0.80	0.31	0.75	0.37	0.83
ALL	0.47	0.64	0.29	0.59	0.43	0.67

TABLE 3: DETECTION RATIO ON CUTS AND GRADUAL TRANSITIONS

Content Type	Method 2		Method 3		Method 4	
	Cuts	Grad	Cuts	Grad	Cuts	Grad
TV News	0.99	0.00	0.97	0.19	0.97	0.19
Situation Comedy	0.92	0.40	0.86	0.59	0.90	0.55
Sports Game	0.70	0.15	0.64	0.18	0.72	0.20
Music Video	0.71	0.15	0.58	0.17	0.75	0.18
Lip Sync	1.00	0.00	1.00	0.50	1.00	0.50
User Generated	0.88	0.43	0.85	0.29	0.90	0.45
ALL	0.80	0.18	0.71	0.23	0.83	0.20

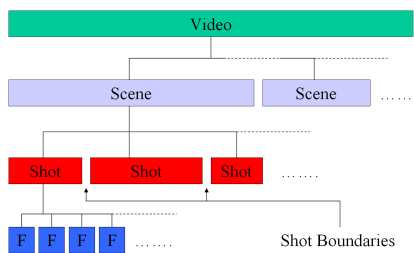


Fig. 1 Video sequence hierarchy.

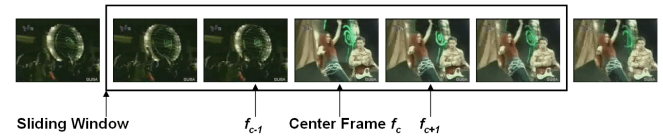


Fig. 2 A sliding window with 5 frames ($m = 2$). The URL of the video is <http://www.youtube.com/watch?v=mumsuRf0bsI>.

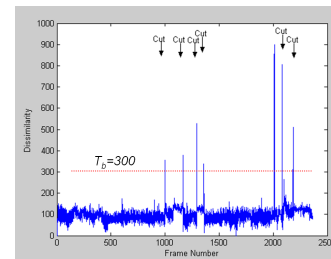


Fig. 3 Plot of dissimilarity against frame number. Six cuts are detected when $T_b = 300$.

