Indian Institute of Technology, Bombay at TRECVID 2006

Nithya Manickam Neela Sawant Aman Parnami Srikanth Lingamneni Sharat Chandran
Computer Science & Engineering Department
Indian Institute of Technology, Bombay
India 400076
http://www.cse.iitb.ac.in/~{mnitya,neela,nsaaman,srikanthl,sharat}

Abstract

This year, Indian Institute of Technology, Bombay participated in TRECVID 2006 in the task of shot detection. We observe that, even though large number of available shot detection methods perform well under normal conditions (as evinced by the results at Trecvid for the last couple of years), they are not robust to sequences that contain dramatic illumination changes, shaky camera effects, and special effects such as fire, explosion, and synthetic screen split manipulations. Traditional systems produce false positives for these cases; i.e., they claim a shot break when there is none.

We propose a shot detection system which reduces false positives even if all the above effects are cumulatively *present in one sequence. Similarities between successive frames are computed by finding the correlation and is further analyzed using a wavelet transformation. A final filtering step is to use a trained Support Vector Machine (SVM). As a result, we achieve better accuracy (while retaining speed) in detecting shot-breaks when compared with other techniques.*

Keywords: *Video Shot Detection, Correlation, Wavelet*

1 Introduction

In recent times, the demand for a tool for searching and browsing videos is growing noticeably. This has led to computer systems internally reorganizing the video into a hierarchical structure of frames, shots, scenes and story. A frame at the lowest level in the hierarchy, is the basic unit in a video, representing a still image. *Shot detection techniques* are used to group frames into shots. Thus, a shot designates a *contiguous sequence of video frames recorded by an uninterrupted camera operation*. A scene is a collection of shots which presents different views of the same event and contain the same object of interest. A story is a collection of scenes that defines an unbroken event. Fig. 1 illustrates this paradigm.

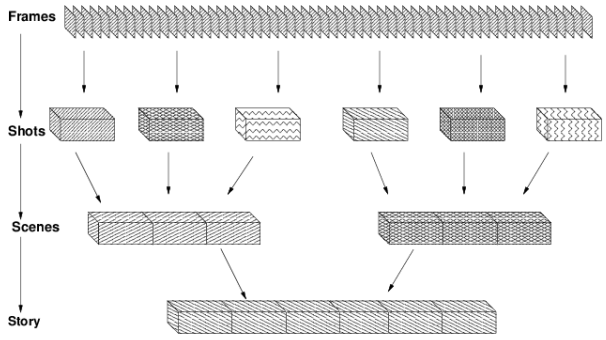


Figure 1. Hierarchical structure of video.

Video shot detection forms the first step in organizing video into a hierarchical structure. Intuitively, a shot captures the notion of a single semantic entity. A *shot break* signifies a transition from one shot to the subsequent one, and may be of many types (for example, fade, dissolve, wipe and hard (or immediate)). Our primary interest lies in improving hard cut detection by *reducing the number of places erroneously declared* as shot breaks (false positives).

A wide range of approaches have been investigated for shot detection but the accuracies have remained low. The simplest method for shot detection is *pair-wise pixel similarity* [20, 1], where the inten-

sity or color values of corresponding pixels in successive frames are compared to detect shot-breaks. This method is very sensitive to object and camera movements and noise. A *block-based approach* [13, 14] divides each frame into a number of blocks that are compared against their counterparts in the next frame. Block based comparison is often more robust to small movements falsely declared as shot-break. Sensitivity to camera and object motion, is further reduced by *histogram comparison* [14, 4, 12, 10, 5]. For example, a 16 bin normalized HSV color histogram is used in [12] to perform histogram intersection. In [10] a combination of local and global histogram is used to detect shot-breaks. However, all these methods perform less than satisfactorily when there are deliberate or inadvertent lighting variations. [7] uses a statistical distribution of color histogram of the shot to refine shot-breaks.

At the cost of more processing, the *edge change ratio method* [19, 18] handles slow transitions by looking for similar edges in the adjacent frames and their ratios. [18] addresses the problem with illumination changes. Three-dimensional *temporal-space methods* [8, 16] are better, but still sensitive to sudden changes in illumination. *Cue Video* [3] is a graph based approach, which uses a sampled three-dimensional RGB color histogram to measure the distance between pairs of contiguous frames. This method can handle special issues such as false positives from flash photography.

1.1 Our Contributions

Our first attempt of detecting shot-breaks only from correlation value resulted in many false positives as the correlation value, when used as is, is unreliable. Therefore, *a multi layer filtering framework* as described in Section 2 is necessary. Based on a large number of experiments, we decided on the use of a Morlet wavelet based feature and a SVM to reduce false positives. It is significant to note that any framework should not increase errors if all unusual effects are cumulatively present in one sequence, or when gradual transitions are present. Our machine learning based scheme avoids this problem. Results of our experiments are given in Section 3 and we end with some concluding remarks in the last section.

2 Proposed method

We propose a shot detection system which reduces errors even if all the effects like dramatic illumination changes, shaky camera effects, and special effects such as fire, explosion, and synthetic screen split manipulations are cumulatively present in one sequence. Similarities between successive frames are computed by finding intensity-compensated correlation using ideas similar to the ones in [15]. We depart, by further analyzing these similarities using wavelet methods to locate the shot breaks and reduce false positives by analyzing the frames around the predicted shot-breaks. We further use learning techniques to refine our shot-breaks. The method is summarized in Fig. 2 and essentially consists of the following three steps.

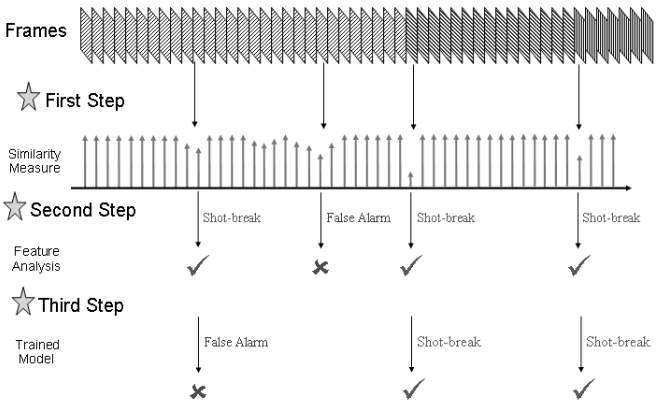


Figure 2. Our filtering approach.

1. Extracting features representing the similarity between the successive frames helps to determine candidate points for shot breaks. Candidate points for shot breaks are where similarity is low; four frames are indicated in the portion marked in Fig. 2 (First Step). This is further elaborated in Section 2.1 (for hard cuts) and Section 2.4 (for gradual transitions).

2. Analyzing features to detect plausible shot breaks. As shown in Fig. 2 (Second Step) the second predicted shot break is dropped because it is a false alarm. This is further elaborated in Section 2.2 (for hard cut). We then refine the

detected shot breaks using more involved techniques to further reduce the false positives.

3. Training the system using a support vector machine to further improve the accuracy. In Fig. 2 (Third Step), the first candidate is now dropped. This technique is elaborated in Section 2.3 (for hard cuts).

2.1 Hard Cut Feature Extraction

The similarity between two consecutive frames is computed using a normalized mean centered correlation. The correlation between two frames f and g is computed as

$$\frac{\sum_{i,j}(f(i,j)-m_f)(g(i,j)-m_g)}{\sqrt{\sum_{i,j}(f(i,j)-m_f)^2}\sqrt{\sum_{i,j}(g(i,j)-m_g)^2}} \quad (1)$$

where m_f and m_g are the mean intensity values of frame f and g respectively. A high correlation signifies similar frames, probably belonging to the same shot; a low value is an indication of an ensuing shot break.

The correlation values between successive frames are plotted as in Fig. 3(a). The locations of shot breaks as identified by a human annotator are also indicated. From this diagram, it is also clear that placing an adhoc value as threshold to detect shot breaks will not work. A delicate shot break, like the one at frame 85 is missed if a hard threshold is placed.

2.2 Hard Cut Shot Prediction

To overcome this difficulty, we consider the continuity of correlation values rather than the correlation values themselves, as an indicator of a shot. We achieve this using wavelet analysis. We have experimented with different wavelet transforms to detect this continuity and have observed that the Morlet wavelet results in a good discrimination between actual shot breaks and false positives.

The Morlet wavelet is a complex sine wave modulated with a Gaussian (bell shaped) envelope as shown in Fig. 3(b) . Note there are equal number of positive and negative values in the mother wavelet and the area sums to zero. Whenever there is no or little change in the correlation sequence, the wavelet transform returns zero value. If there is a hard cut, there is a discontinuity in the correlation value, which results in a distinctive PPNN pattern (two positive values followed by two negative values) in the lowest scale. At high scales the coefficient values are quite large. Hence hard cuts can be obtained by observing this pattern.

We graphically illustrate the power of the wavelet in Fig. 4. Fig. 4(a) shows a fluctuation in the correlation values from frames 215 up to 420. Out of these, frames 215 and 387 look like possible candidates for shot breaks. However, only frame 215 is an actual cut and frame 387 is a false positive (if reported as a cut).

In contrast, observe the corresponding Morlet wavelet transform in Fig. 4(b). The wavelet coefficients are high in all the scales around the frame 215, whereas the wavelet coefficients value around the frame 387 is not high at all the scales. Thus frame 215 is detected correctly as shot-break and frame 387 is dropped.

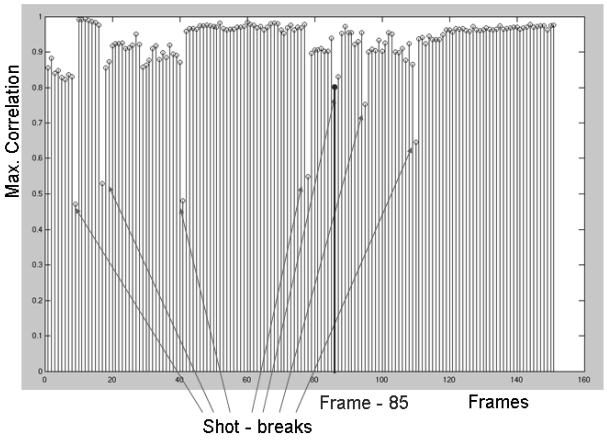
2.2.1 Filtering:

After detecting possible locations of shot breaks, we improve the accuracy by analyzing the frames around predicted shot breaks in greater detail. The following measures are used.

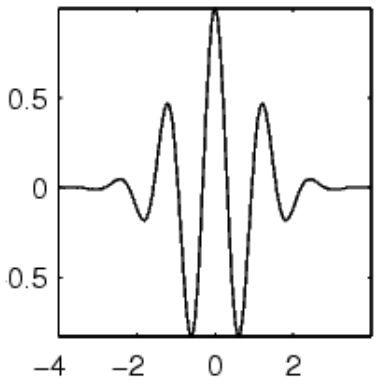
1. Due to random lighting variations, the gray-scale value of successive frames in a shot might differ considerably resulting in a low correlation value. We pass potential shot break frames through a median filter. As a result, false positives are decreased without increasing false negatives.

2. Synthetic manipulations such as animations or screen-split cause the correlation coefficient to become low resulting in false positives. We divide the frame into four overlapping sub-frames as shown in Fig. 5 and compute the correlation of corresponding sub-frames. One of these four correlation values reflect the desired relation. As a result, false positives are decreased.

3. MPEG errors and noise in the neighboring frame in low quality video can cause false positives in spite of recomputing the correlation value at shot-breaks. The correlation of the frames around the shot-break is recomputed in a window size as

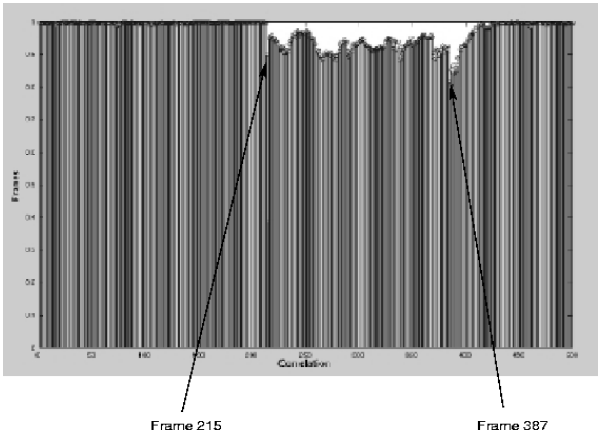


(a) A sample correlation sequence. Low values might indicate shot breaks.

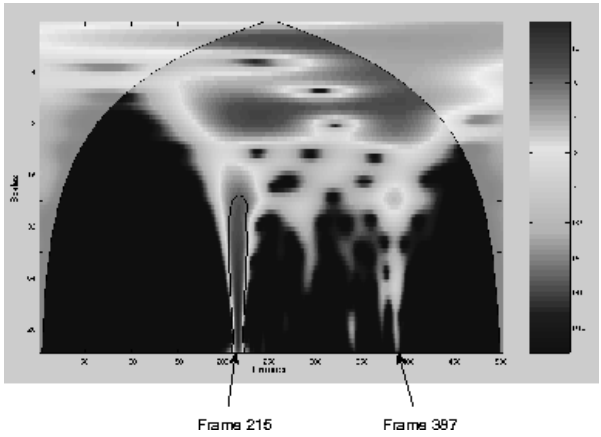


(b) $\psi(t) = Ce^{(\frac{-t^2}{2})}\cos(5t)$.

Figure 3. Similarity features and the Morlet mother wavelet.



(a) A sample correlation sequence.



(b) A visualization of the relevant wavelet transform

Figure 4. Using the Morlet wavelet.

shown in Fig. 6. This measure helps in reducing false positives due to noise in the subsequent frames from the same shot.

4. Camera or object motion may cause low correlation value resulting in false positives. For the predicted frames only, *cross-correlation* is computed.

We select the best correlation values generated using the above measures and rerun the process of computing wavelet coefficients and detecting discontinuities with these new values. Finally, by taking the intersection of the two sets of predicted shot breaks, we produce a pruned set.

2.3 Training

We now describe how to train a SVM to further improve our accuracy. As the features play an important role in the training, we mainly focus on the features used in this process. The features extracted in previous two steps contribute correlation and wavelet features. Apart from this, we also compute traditional features like pixel differences, histogram differences, and edge differences. The training set consists of news videos, and movie clips and videos containing chal-

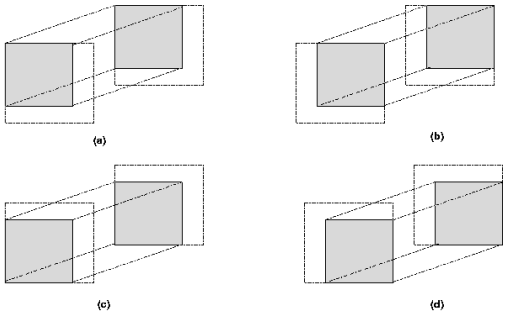


Figure 5. Computing correlation of corresponding sub-windows.

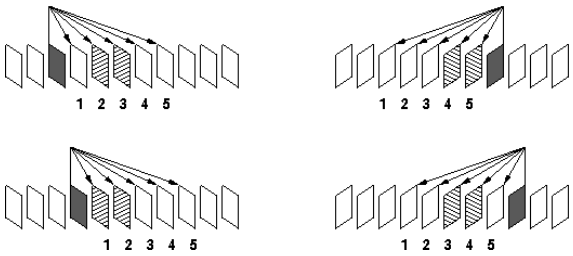


Figure 6. Recomputing correlation in the frames around the shot-break. The dashed window indicates a shot break and the frame under focus is darkened. The correlation between the dark frame and other frames indicated by arrows is computed. The maximum of these values replaces the value computed earlier.

lenging problems like illumination changes, fast camera and object motion. The features used in training the SVM are

1. Pixel differences which includes average pixel difference and Euclidean pixel difference
2. Histogram differences: Average histogram difference, histogram intersection, thresholded chi-square distance
3. Edge difference
4. Average intensity value
5. Correlation, Cross-correlation and maximum of the correlation values computed in the previous step
6. Presence of PPNN pattern in the lowest level of wavelet transform computed in the previous step
7. Lowest wavelet coefficient

Though our feature set contains some duplication, we use standard machine learning methods to select relevant features.

2.4 Gradual Transitions

Gradual transitions (or *graduals*) are shot transitions which occur over multiple frames resulting in smooth transition from one shot to another. As a result, gradual transitions are comparatively difficult to detect when compared to hard-cuts. The problem is increased with issues like uncertain camera motion common among amateurs resulting in false positives. Unfortunately, imposing more constraints to eliminate these false positives can eliminate the actual graduals as well. Most of the gradual detection algorithms [17, 20, 11, 2, 6] use a hard threshold to detect the shot transitions.

In this trecvid, we have attempted to detect dissolves using a simple method. Change in the brightness value of frames is taken as indication of presence of dissolve. The total brightness of a frame f is computed as

$$\sum_i \sum_j [f(i,j)]^2 \tag{2}$$

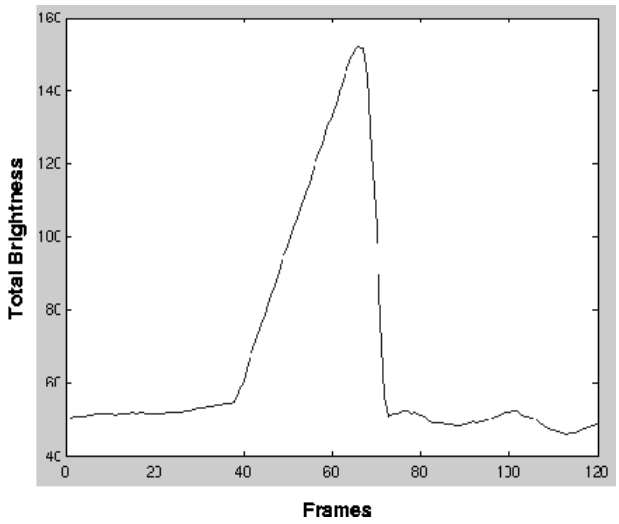
Within a shot, the total brightness remains predictable by and large. Upon encountering a gradual, we see a cone-like pattern. Fig. 7(a) shows a sample situation.

The change in brightness values are also caused due to camera motion and other changes as shown in Fig. 7(b). As the trecvid test data set contains such cases, our system has reported false positives. We have achieved a recall of 0.691 and a precision of 0.224, with frame recall and precision being 0.729 and 0.626 respectively.

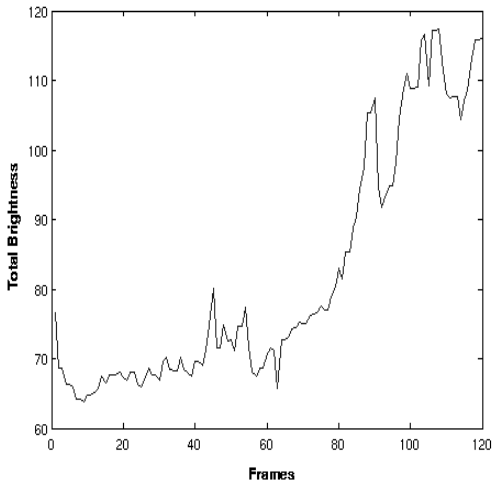
3 Experimental results and discussion

Our training data consists of following videos

- News videos from Trecvid [9] data set
- Short videos taken from motion pictures and from NASA.
- Low-quality home video with varying lighting conditions and fast, shaky motion.



(a) A fade-out from frame 40 to 65 results in increasing brightness and a fade-in from frame 65 to 75 results in decreasing brightness value.



(b) Multiple "cones" can be found even when there is no shot break. In this example, a moving object has caused the changes.

Figure 7. Sample brightness values around a gradual transition (a) can sometimes be predictable. At other times, the pattern can result in false positives.

Table 1 shows the experimental results on this years test data. In the first run (runid: tb1), we used our hard cut detection algorithm described in Section 2.1 - 2.3. In the training phase, we labeled only hard cuts (the ones in which transitions occurs immediately, as opposed to short graduals) as 1s and rest as 0s. In this method, we have achieved a recall of 0.771 and a precision of 0.918 for hard cut.

Our second run (runid: tb2) is similar to the first run, except that in the training process, we have labeled all correctly detected shot-breaks as 1s and others as 0s. Our accuracy improved to a recall of 0.815 and a precision of 0.872 for hard cut.

In our third run (runid: tb4), in addition to hard cut, we have also included the result from our gradual detection algorithm described in Section 2.4. Though cut recall has improved, cut precision has decreased due to the false positives introduced by gradual detection method. For gradual transitions, we have achieved a recall of 0.691 and a precision of 0.224, with frame recall and precision of 0.729 and 0.626.

As our hard cut detection algorithm performs better, we have used it to detect short gradual transitions too. We ran our shot detection method by dropping four in between frames, so that our algorithm can detect short graduals. We have used the frame accuracies detected by hard-cut and gradual detection algorithm, as this method has approximate shot boundaries. This method (runid: tb3) has performed well in detecting short graduals and improved the recall and precision to 0.839 and 0.830 respectively.

4 Conclusions

In summary, we use mean-centered correlation as the similarity measure and use Morlet wavelet to predict shot-breaks by capturing the discontinuity in the correlation sequence. We further improve our accuracy by using a support vector machine.

Our shot detection system achieves the following:

1. Reduces false positives in the event of challenging problems like unpredictable illumination changes, camera effect & special effects.

2. Presents a unique solution to solve all the problems, instead of combining different problem specific solutions.

	All		Cuts		Graduals			
							Frame	
Run ID	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
tb1	0.563	0.918	0.771	0.918	0.000	0.000	0.000	0.000
tb2	0.594	0.872	0.815	0.872	0.000	0.000	0.000	0.000
tb3	0.721	0.687	0.839	0.830	0.403	0.348	0.660	0.626
tb4	0.818	0.417	0.865	0.558	0.691	0.224	0.729	0.626

Table 1. Trecvid Results for test data

3. Introduces a new wavelet based feature based on extensive experiments.

References

[1] Chengcui Bang, Shu-Ching Chenl, and Mei-Ling Shyu. Pixso: a system for video shot detection. *Fourth International Conference on Information, Communications and Signal Processing*, pages 1320–1324, December 2003.

[2] M. Covell and S. Ahmad. Analysis by synthesis dissolve detection. In *International Conference on Image Processing.*, pages 425–428, 2002.

[3] A. Amir et. al. IBM Research TRECVID-2005 Video Retrieval System. In *TREC Proc*, November 2005.

[4] B.V. Funt and G.D. Finlayson. Color constant color indexing. *Pattern Analysis and Machine Intelligence, IEEE*, 17:522–529, 1995.

[5] DaLong Li and Hanqing Lu. Avoiding false alarms due to illumination variation in shot detection. *IEEE Workshop on Signal Processing Systems*, pages 828–836, October 2000.

[6] R. Lienhart and A. Zaccarin. A system for reliable dissolve detection in videos. In *International Conference on Image Processing*, pages III: 406–409, 2001.

[7] H. Lu and Y.P. Tan. An effective post-refinement method for shot boundary detection. *CirSysVideo*, 15(11):1407–1421, November 2005.

[8] C.W. Ngo, T.C. Pong, and R.T. Chin. Detection of gradual transitions through temporal slice analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I: 36–41, 1999.

[9] NIST. *TREC Video Retrieval Evaluation*. `www-nlpir.nist.gov/projects/trecvid`.

[10] N.V. Patel and I.K. Sethi. Video shot detection and characterization for video databases. *Pattern Recognition*, 30(4):583–592, April 1997.

[11] Christian Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. In *TREC Proc*, November 2004.

[12] Z. Rasheed and M. Shah. Scene detection in Hollywood movies and TV shows. In *IEEE Conference on Computer Vision and Pattern Recognition.*, pages II: 343–348, June 2003.

[13] S. Shahraray. Scene change detection and content-based sampling of video sequence. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 2–13, February 1995.

[14] D. Swanberg, C.F. Shu, and R. Jain. Knowledge guided parsing in video database. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 13–24, May 1993.

[15] T. Vlachos. Cut detection in video sequences using phase correlation. *Signal Processing Letters*, 7(7):173–175, July 2000.

[16] Chuohao Yeo, Yong-Wei Zhu, Qibin Sun, and Shih-Fu Chang. A framework for sub-window

shot detection. In *MMM '05: Eleventh International Multimedia Modelling Conference (MMM'05)*, pages 84–91, 2005.

[17] Hun-Woo Yoo, Han-Jin Ryoo, and Dong-Sik Jang. Gradual shot boundary detection using localized edge blocks. *Multimedia Tools and Applications*, 28(3):283–300, 2006.

[18] Geng Yuliang and Xu De. A solution to illumination variation problem in shot detection. *TENCON 2004. IEEE Region 10 Conference*, pages 81–84, November 2004.

[19] Ramin Zabih, Justin Miller, and Kevin Mai. Feature-based algorithms for detecting and classifying scene breaks. Technical report, Cornell University, 1995.

[20] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, 1:10–28, 1993.