Reducing False Positives in Video Shot Detection Using Learning Techniques

Nithya Manickam, Aman Parnami, and Sharat Chandran

Department of Computer Science and Engineering
Indian Institute of Technology Bombay
`http://www.cse.iitb.ac.in/~{mnitya, nsaaman, sharat}`

Abstract. Video has become an interactive medium of daily use today. However, the sheer volume of the data makes it extremely difficult to browse and find required information. Organizing the video and locating required information effectively and efficiently presents a great challenge to the video retrieval community. This demands a tool which would break down the video into smaller and manageable units called shots.

Traditional shot detection methods use pixel difference, histograms, or temporal slice analysis to detect hard-cuts and gradual transitions. However, systems need to be 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.

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.

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

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 421–432, 2006.

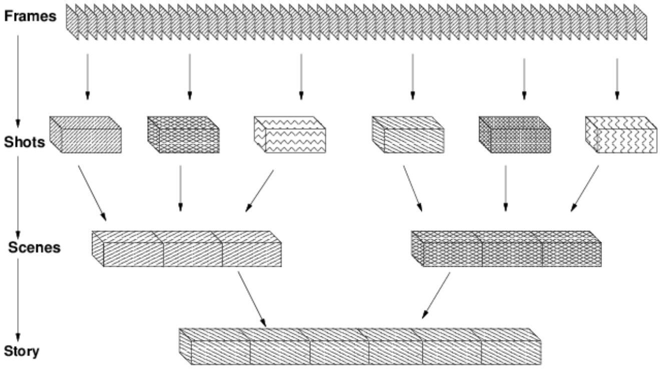


Fig. 1. Hierarchical structure of video

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 *pairwise pixel similarity* [1,2], where the intensity 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* [3,4] 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* [4,5,6,7,8]. For example, a 16 bin normalized HSV color histogram is used in [6] to perform histogram intersection. In [7] 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. [9] 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* [10,11] handles slow transitions by looking for similar edges in the adjacent frames and their ratios. [11] addresses the problem with illumination changes. Three-dimensional *temporal-space methods* [12,13] are better, but still sensitive to sudden changes in illumination. *Cue Video* [14] 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 Problem Statement

As mentioned earlier, our main interest is in reducing false positives in challenging situations enumerated below.

1. *Illumination changes*: An example of this situation (inter-reflections, user-driven light changes, flash photography) is illustrated in Fig. 2. In the movie excerpt, lighting causes the actress to appear different. It is natural to the human, but confuses shot detection algorithms and even the camera as seen in the third frame!
2. *Camera effects*: These include effects such as zooming and tilting of objects of interest, shaky handling of amateur video, fast object motion, and fast camera motion. An example is illustrated in Fig. 3.
3. *Special effects*: An example of this situation (explosion) is illustrated in Fig. 4. Split screen is another possibility shown in the last figure.

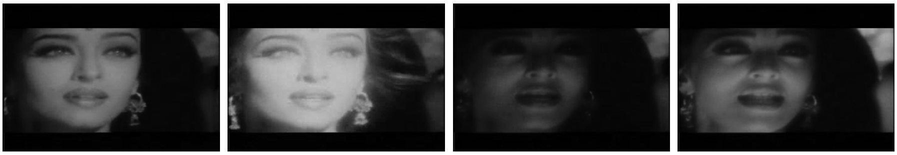


Fig. 2. A movie excerpt featuring Aishwarya Rai. Lightning creates unpredictable lighting changes.

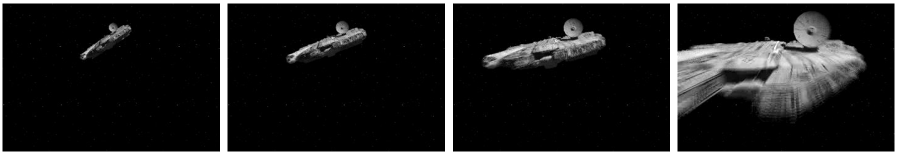


Fig. 3. Fast camera motion makes individual frames undecipherable

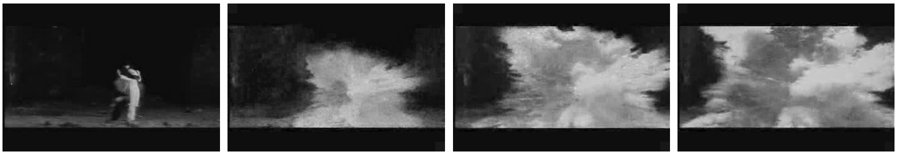


Fig. 4. Explosion in a dimly lit scene causes considerable change in color and intensity

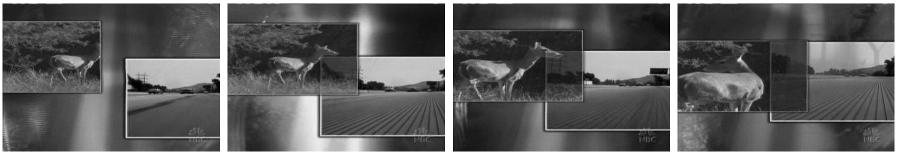


Fig. 5. Two different scenes are displayed simultaneously using split-screen methods. However, a shot break may be observed in only one of them.

1.2 This Paper and 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 above effects 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. 6 and essentially consists of the following three steps.

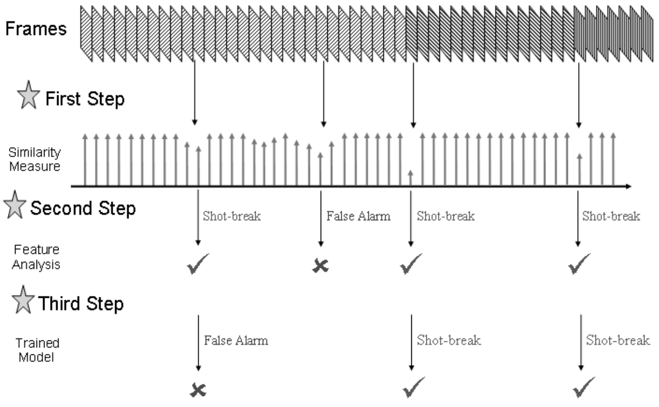
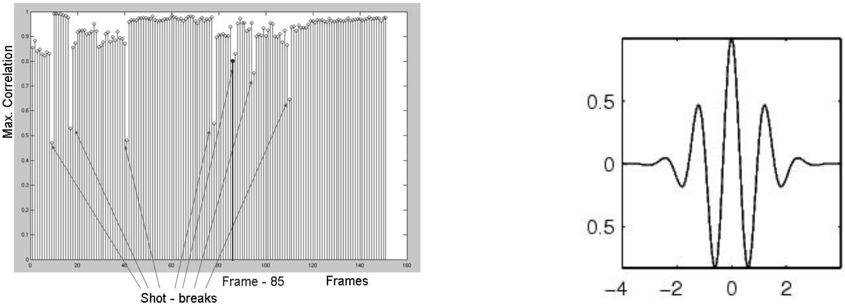


Fig. 6. Our filtering approach

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. 6 (First Step). This is further elaborated in Section 2.1 (for hard cuts) and Section 2.4 (for gradual transitions).



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

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

Fig. 7. Similarity features and the Morlet mother wavelet

2. Analyzing features to detect plausible shot breaks. As shown in Fig. 6 (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) and Section 2.5 (for gradual transitions). 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. 6 (Third Step), the first candidate is now dropped. This technique is elaborated in Section 2.3 (for hard cuts) and Section 2.6 (for gradual transitions).

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

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

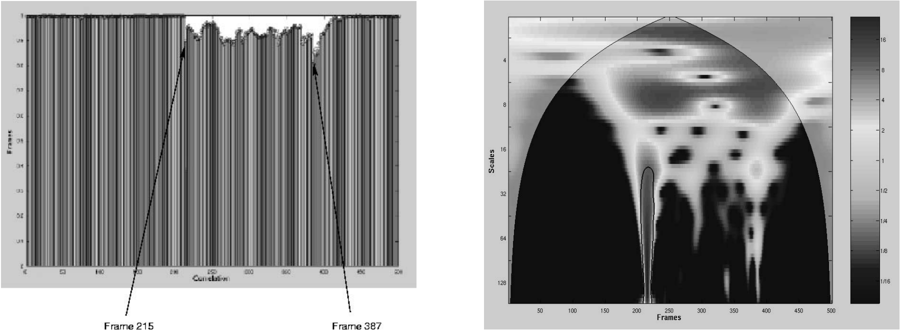
where m_f and m_g are the mean intensity values of frame f and g respectively. A high correlation signifies similar frames, probably belonging to the same shot; a low value is an indication of an ensuing shot break.

The correlation values between successive frames are plotted as in Fig. 7(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 ad-hoc 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. 7(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.



(a) A sample correlation sequence.

(b) A visualization of the relevant wavelet transform

Fig. 8. Using the Morlet wavelet

We graphically illustrate the power of the wavelet in Fig. 8. Fig. 8(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. 8(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.

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.

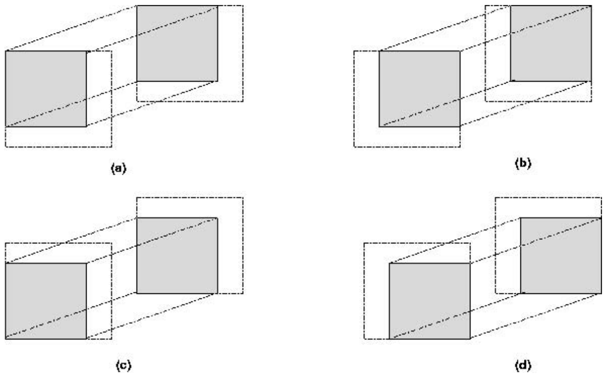


Fig. 9. Computing correlation of corresponding sub-windows

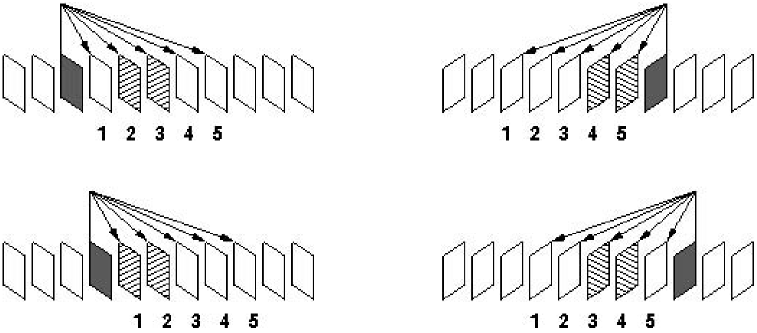


Fig. 10. Recomputing correlation in the frames around the shot-break. The dashed window indicates a shot break and the frame under focus is darkened. The correlation between the dark frame and other frames indicated by arrows is computed. The maximum of these values replaces the value computed earlier.

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. 9 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 shown in Fig. 10. 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 videos containing the challenging problem presented in Section 1.1, news videos, and movie clips. 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.

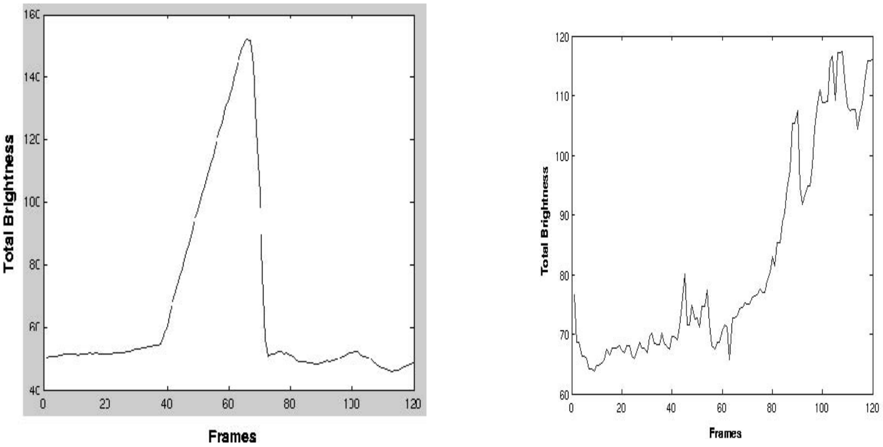
2.4 Gradual Transitions

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 [16,1,17,18,19] use a hard threshold to detect the shot transitions. Tuning these thresholds to improve the accuracy of the gradual detection system is a critical and important task. We use machine learning algorithms to solve this task.

A primary feature used in gradual detection is in the change in the brightness value of frames. 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. 11(a) shows a sample situation.



(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.

Fig. 11. Sample brightness values around a gradual transition (a) can sometimes be predictable. At other times, the pattern can result in false positives.

2.5 Gradual Shot Prediction

The possible gradual transitions are predicted by detecting steady increase or decrease in the brightness values of the image sequence. Unfortunately, as exemplified in Fig. 11(b), false positives are produced. We improve the accuracy by analyzing the frames around predicted shot breaks in greater detail. The following measures are used.

1. Analysis by synthesis: The detected gradual transitions are checked for the dissolve linear property [17] thereby eliminating some of the false positives.
2. Edge Energy: Sequences containing illumination changes affects the total brightness value but do not affect the edge energy computed as

$$\sum_i \sum_j [\mathrm{edge}(f(i,j))]^2 \tag{3}$$

As a result, a few false positives are eliminated.

2.6 Training

The following features are used for classifying gradual transitions. The videos in the training set are rife with characteristics mentioned in Section 1.1.

1. The normalized brightness difference between the start and end of the gradual transition.

2. The differences of brightness value between starting and middle point of the gradual transition and the difference between the brightness value of middle and end value is computed. The sum of these two differences is calculated.
3. The difference between the minimum value and the maximum value in the pixel difference sequence computed in the previous step.
4. Average edge energy value of the dissolve interval.

By this training process, we eliminate most of the hard-coded thresholds and make our system more flexible to use.

3 Experimental Results and Discussion

Our training data consists of a large number of videos. These involving the challenging problems stated in our problem statement and other videos which do not have too many unusual situations. Normal videos are included in the training set to avoid the over fitting problem in SVM.

We have tested our system on the data comprising of

- News videos each having around 500 hard cuts, containing different types of events. These are in multiple languages (notably Chinese and English).
- Short videos taken from motion pictures and from NASA. These involve some of the challenging problems mentioned in Section 1.1.
- Low-quality home video with varying lighting conditions and fast, shaky motion.
- Clips from motion picture containing illumination changes, fast camera motion, and object intrusion.

The ground truth for these experiments is from Trecvid 2005 [20].

Table 1 shows the experimental results on a news video which is a synthetically combined video of various challenging problems like fast camera motion, illumination change, flash light, explosion, and low video quality. We present results which shows the efficacy of our filtering approach. In the first attempt, we detect shot-breaks using only the first step (see Section 2). In the second attempt, we imposed the constraints to remove false positives as noted earlier, but do not use any learning methods. We note that the precision improves, but the recall drops. Many of the true positives were also eliminated. The third row shows our result on cross-validation with a split of 33% training data and 66% test data. The precision and recall go up.

Table 1. Result from a news video

Method	Precision	Recall	F-Measure
Without training, without filtering	0.686	1.0	0.814
Without training, with filtering	0.761	0.768	0.764
Cross-validation(33-training, 66-test)	0.892	0.992	0.939

Table 2. Result from an unseen video containing fast camera motion, object intrusion, and unpredictable illumination changes

Method	Precision	Recall	F-measure
Pixel Difference	0.926	0.571	0.707
Histogram Comparison	0.259	0.195	0.222
Correlation Value	0.785	0.734	0.759
Temporal Slice	0.750	0.604	0.669
Our Method	0.923	1.000	0.960

Table 2 shows the experimental results on unseen test data from motion video containing problems like fast camera motion, shaky handling of camera, object intrusion and illumination changes. The ground truth for these experiments was generated manually. As the results reflect, our system is successful in reducing the false positives considerably.

4 Conclusions

We have discussed in this paper the characteristics of videos that make shot detection a challenging problem. We have presented our framework that improves the accuracy of shot detection in such cases. 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. Processes more than 30 frames per second with the accuracy required for the normal usage.
3. Presents a unique solution to solve all the problems, instead of combining different problem specific solutions.
4. Introduces a new wavelet based feature based on extensive experiments.

References

1. Zhang, H., Kankanhalli, A., Smoliar, S.: Automatic partitioning of full-motion video. ACM Multimedia Systems **1** (1993) 10–28
2. Bang, C., Chenl, S.C., Shyu, M.L.: Pixso: a system for video shot detection. Fourth International Conference on Information, Communications and Signal Processing (2003) 1320–1324
3. Shahraray, S.: Scene change detection and content-based sampling of video sequence. In: SPIE Storage and Retrieval for Image and Video Databases. (1995) 2–13
4. Swanberg, D., Shu, C., Jain, R.: Knowledge guided parsing in video database. In: SPIE Storage and Retrieval for Image and Video Databases. (1993) 13–24

5. Funt, B., Finlayson, G.: Color constant color indexing. Pattern Analysis and Machine Intelligence, IEEE **17** (1995) 522–529
6. Rasheed, Z., Shah, M.: Scene detection in Hollywood movies and TV shows. In: IEEE Conference on Computer Vision and Pattern Recognition. (2003) II: 343–348
7. Patel, N., Sethi, I.: Video shot detection and characterization for video databases. Pattern Recognition **30** (1997) 583–592
8. Li, D., Lu, H.: Avoiding false alarms due to illumination variation in shot detection. IEEE Workshop on Signal Processing Systems (2000) 828–836
9. Lu, H., Tan, Y.: An effective post-refinement method for shot boundary detection. CirSysVideo **15** (2005) 1407–1421
10. Zabih, R., Miller, J., Mai, K.: Feature-based algorithms for detecting and classifying scene breaks. Technical report, Cornell University (1995)
11. Yuliang, G., De, X.: A solution to illumination variation problem in shot detection. TENCON 2004. IEEE Region 10 Conference (2004) 81–84
12. Ngo, C., Pong, T., Chin, R.: Detection of gradual transitions through temporal slice analysis. In: IEEE Conference on Computer Vision and Pattern Recognition. (1999) I: 36–41
13. Yeo, C., Zhu, Y.W., Sun, Q., Chang, S.F.: A framework for sub-window shot detection. In: MMM '05: Eleventh International Multimedia Modelling Conference (MMM'05). (2005) 84–91
14. et. al., A.A.: IBM Research TRECVID-2005 Video Retrieval System. In: TREC Proc. (2005)
15. Vlachos, T.: Cut detection in video sequences using phase correlation. Signal Processing Letters **7** (2000) 173–175
16. Yoo, H.W., Ryoo, H.J., Jang, D.S.: Gradual shot boundary detection using localized edge blocks. Multimedia Tools and Applications **28** (2006) 283–300
17. Petersohn, C.: Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. In: TREC Proc. (2004)
18. Covell, M., Ahmad, S.: Analysis by synthesis dissolve detection. In: International Conference on Image Processing. (2002) 425–428
19. Lienhart, R., Zaccarin, A.: A system for reliable dissolve detection in videos. In: International Conference on Image Processing. (2001) III: 406–409
20. NIST: TREC Video Retrieval Evaluation.
`www-nlpir.nist.gov/projects/trecvid` (2005)