Video Shot Detection Using Cumulative Colour Histogram

B. H. Shekar, K. Raghurama Holla and M. Sharmila Kumari

Abstract The shot boundary detection is the fundamental step in video indexing and retrieval. In this paper a new method for video shot boundary detection based on *slope* and *y-intercept* parameters of the straight line fitted to the cumulative plot of color histogram is proposed. These feature vectors are extracted from every video frames and the frame dissimilarity values are compared against a threshold to identify the *cuts* and *fades* present in the video sequence. Experiments have been conducted on TRECVID video database to evaluate the effectiveness of the proposed model. A comparative analysis with other models is also provided to reveal the superiority of the proposed model for shot detection.

Keywords Cumulative colour histogram · Cut detection · Fade detection · Video segmentation · Shot boundary detection

B. H. Shekar (✉) · K. Raghurama Holla
Department of Computer Science, Mangalore University,
Konaje, Mangalore, Karnataka, India
e-mail: bhshekar@gmail.com

K. Raghurama Holla
e-mail: raghu247@gmail.com

M. Sharmila Kumari
Department of Computer Science and Engineering, P. A. College of Engineering,
Mangalore, Karnataka, India
e-mail: sharmilabp@gmail.com

Mohan S. and S. S. Kumar (eds.), *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012)*, Lecture Notes in Electrical Engineering 222,
DOI: 10.1007/978-81-322-1000-9_34,

1 Introduction

As a consequence of rapid development in video technology and consumer electronics, the digital video has become an important part of many applications such as distance learning, advertising, electronic publishing, broadcasting, security, video-on-demand and so on, where it becomes more and more necessary to support users with powerful and easy-to-use tools for searching, browsing and retrieving media information. Since video databases are large in size, they need to be efficiently organized for quick access and retrieval. Shot detection is the fundamental step in video database indexing and retrieval. The video database is segmented into basic components called shots, which can be defined as an unbroken sequence of frames captured by one camera in a single continuous action in time and space. In shot boundary detection, the aim is to identify the boundaries by computing and comparing similarity or difference between adjacent frames. So, the video shot detection provides a basis for video segmentation and abstraction methods [7].

In general, shot boundaries can be broadly classified into two categories [3, 7]: abrupt shot transitions or *cuts*, which occur in a single frame where a frame from one shot is followed by a frame from a different shot, and gradual shot transitions such as *fades*, *wipes*, and *dissolves*, which are spread over multiple frames. A *fade-in* is a gradual increase in intensity starting from a solid color. A *fade-out* is a slow decrease in brightness resulting in a black frame. A *dissolve* is a gradual transition from one scene to another, in which the two shots overlap for a period of time. Gradual transitions are more difficult to detect than abrupt transitions.

The remaining part of the paper is organized as follows. Section 2 presents the review of existing models for shot detection. The proposed model is discussed in Sect. 3. Experimental results and comparison with other models are presented in Sect. 4 and conclusion is provided in Sect. 5.

2 Review of Existing Work

There are several models available in the literature for video shot boundary detection. To identify shot boundary, the low level visual features are extracted and the inter-frame difference is calculated using these extracted features. Each time the frame difference measure exceeds the threshold, the presence of shot boundary is declared. Some of the popular existing models for cut detection and fade detection are reported in the following section.

2.1 Cut Detection: A Review

The simplest way of detecting the hard cut is based on pair-wise pixel comparison [3], where the differences of corresponding pixels in two successive frames are computed to find the total number of pixels that are changed between two consecutive frames. The absolute sum of pixel differences is calculated and then compared against a threshold to determine the presence of cut. However, this method is sensitive to object and camera motions. So an improved approach is proposed based on the use of a predefined threshold to determine the percentage of pixels that are changed [22]. This percentage is compared with a second threshold to determine the shot boundary.

Since histograms are invariant to rotation, histogram based approaches are used in [22]. Whenever the histogram difference between two consecutive frames exceeds some threshold, a shot boundary is detected. When the color images are considered, some appropriate weights are assigned to the histogram of each color component depending on the importance of color space, so weighted histogram based comparison was suggested [9].

To improve the accuracy of shot identification process, low level visual features such as edges and their properties are used in some of the techniques. In [21] proposed a shot detection method by analysing *edge change ratio* (ECR) between consecutive frames. The percentage of edge pixels that enter and exit between two consecutive frames are calculated. The cuts and gradual transitions can be detected by comparing the ratio of entering and exiting edge pixels. The limitation of the edge based methods is that they fail to produce good results when the video sequence contains high speed object motions. To overcome this shortcoming, an algorithm for motion compensation was developed [2, 22]. The block matching procedure is used and motion estimation is performed. However, motion based approach is computationally expensive.

There are some models which use statistical features for shot detection [2, 9]. The image is segmented into a fixed number of blocks and the statistical measures like mean and standard deviation of pixels in the blocks are computed. These features are extracted from successive frames and compared against a threshold to detect a shot cut. As this type of methods may lead to false hits, more robust techniques using likelihood ratio (LHR) was proposed [9]. Even though likelihood ratio based methods produce better results, extracting statistical features is computationally expensive.

In [10] proposed a text segmentation based approach for shot detection. The combination of various features such as, color moments and edge direction histogram are extracted from every frame and distance metrics are employed to identify shots in a video scene. Gabor filter based approach was used for cut detection by convolving each video frame with a bank of Gabor filters corresponding to different orientations [17].

In [13], dominant color features in the HSV color space are identified and a histogram is constructed. Block wise histogram differences are calculated and the shot boundaries are detected. In [6], a shot segmentation method based on the concept of visual rhythm is proposed. The video sequence is viewed in three dimensions: two in the spatial coordinates and one in the temporal i.e. corresponding to frame sequence. The visual rhythm approach is used to represent the video in the form of 2D image by sampling the video. The topological and morphological tools are used to detect cuts. We can see that the combinations of various features such as pixels, histograms, motion features etc used to accurately detect shot transitions [12]. The frame difference values are computed separately for individual features to address different issues such as flash light effect, object or camera motion. For more review on shot boundary detection models see [15, 19].

2.2 Fade Detection: A Review

Once the abrupt transitions are identified, then it is very important to detect the gradual shot transitions such as *fade in* and *fade outs* which are present in the video for designing an effective video retrieval system. Usually, gradual transitions are hard to detect when compared to abrupt transitions.

In histogram based approaches, where the histogram differences of successive frames are computed and two thresholds [22] are used to detect the shot transitions: higher threshold T_h for detecting cuts and lower threshold T_l for detecting gradual transition. Initially, all the cuts are detected using T_h and then the threshold T_l is applied to the remaining frame difference values to determine the beginning of gradual transition. Whenever the start frame of the gradual transition is detected, the cumulative sum of consecutive frame difference is calculated until the sum exceeds T_h. Thus the end frame of the gradual transition is found. However, there is a possibility of false positives if the thresholds are not properly set.

In [1] suggested a fade detection model by analysing the first and second derivative of the luminance variance curves. This method has a limitation of being sensitive to motion and noise. Fernando et al. [4] used statistical features of luminance and chrominance components to detect fades. Zabih et al. [21] developed a model for shot detection based on analysing edge change ratio between adjacent frames. The percentage of edge pixels that enter and exit between consecutive frames are computed. The cuts and gradual transitions can be detected by comparing the ratio of entering and exiting edge pixels. Troung et al. [16] improves the fade detection using two step procedure: detecting the solid color frame in the first step and searching for all spikes in the second derivative curve. In another approach, fade in and fade out transitions were detected using histogram spans of video frame [5], which works based on the variations of the dynamic range during fade transitions. A fade detection method was proposed based on the principle that the horizontal span of histogram should increase for fade in and decrease for fade out. Each frame is divided into four regions and the histogram spans for these regions are computed to identify the beginning and end frame of fade transitions. In [6], Guimarães et al. proposed a shot detection algorithm based on Visual Rhythm Histogram (VRH). Fade regions are identified by detecting the inclined edges in VRH image. However, if the solid color frames are not solid black or white, then it results in two cross edges in the VRH image. In [18], localized edge blocks are used to identify gradual transition detection based on the concept of variance distribution of edge information in the frame sequences.

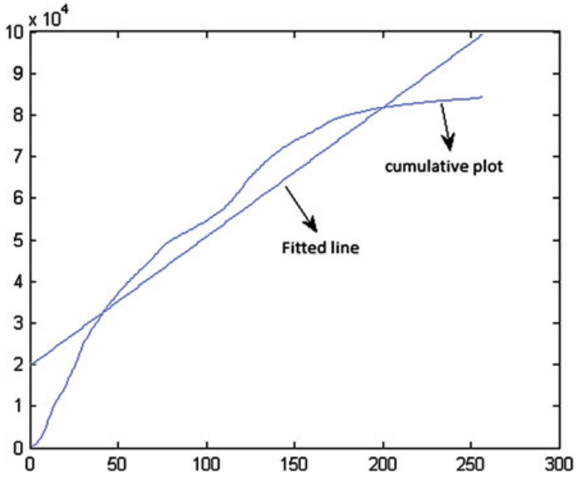

Fig. 1 Fitting a *straight line* to the cumulative plot

We have seen that several algorithms are available for shot boundary detection. The proposed technique is different from the existing models in terms of the reduced dimension of feature vector, hence suitable for real time video database processing applications. In the proposed work, we focussed on detecting both hard cuts and fade transitions present in the video. The details of the proposed model are given in Sect. 3.

3 Proposed Model

The proposed model is based on the *slope* and *y-intercept* parameters of the cumulative plot of the color histogram. Given a video containing n frames, the histograms are computed for all the three channels R, G and B separately. The cumulative histograms are obtained from each histograms. We fit a straight line to the cumulative histogram as shown in Fig. 1. Then the line parameters such as *slope* and *y-intercept* are calculated. So, it results in six element feature vector for each video frame. This feature vector is subsequently used to identify the shot transitions based on the frame differences. The frame difference between two consecutive frames is calculated as follows:

$$D(n) = d(F_n - F_{n-1}) \tag{1}$$

where

$$d(F_i - F_{i-1}) = \sum_{k=1}^{N} \|F_i(k) - F_{i-1}(k)\|^q \tag{2}$$

where N is the length of the feature vector, i.e. $N = 6$. When $q = 2$, $d(i,j)$ is the Euclidean distance between the frame i and frame j. The value of D indicates the change tendency of consecutive frames.

3.1 Cut Detection

The inter-frame dissimilarity values obtained in the previous stage are analysed to perform cut detection. We have used the *local adaptive thresholding* [20] since using a single global threshold is inadequate to detect all the cuts present in the video. In the first stage, global threshold T_g [11] is calculated using global mean μ_g and standard deviation σ_g obtained from the frame dissimilarity values D as follows:

$$T_g = \mu_g + \beta \cdot \sigma_g \tag{3}$$

where β is a constant which controls the tightness of T_g.

In the second stage, we consider only those frame dissimilarity values where, $D(V_n, V_{n-1}) \geq T_g$ for calculating the local threshold. Using a sliding window of size m, the mean and the standard deviation of left side frame difference(video frames: 1 ... $\frac{m}{2} - 1$) and right side frame difference(video frames: $\frac{m}{2} + 1$ ... m) from the middle frame difference($m/2$) are computed. The middle frame is declared to be the *cut* if the following conditions are completely satisfied:

(i) The middle frame difference value is the maximum within the window.
(ii) The middle frame difference value is greater than $max(\mu_L + \sigma_L T_d, \mu_R + \sigma_R T_d)$, where μ_L, σ_L and μ_R, σ_R represent mean and standard deviation values of the left side and right side of the middle frame difference value respectively. T_d is a weight parameter.

We have empirically fixed the values of β, m and T_d to be 1.5, 9 and 5 respectively.

3.2 Fade Detection

In order to detect fade transitions, we consider only the frame differences corresponding to non-cut frames, i.e. the fade-detection is followed by cut-detection process. Once the cuts are detected in a video segment, the associated frame difference values are not considered for further processing. The threshold selection for fade detection is done using local adaptive thresholding technique as suggested in [8]. A sliding window is placed preceding the current frame difference value. The mean (μ) and standard deviation (σ) within the window is calculated. The thresholds $\mu + \sigma k_1$ and $\mu + \sigma k_2$ are used to detect the beginning and end of fade transition respectively, where $k_1 = 2$ or 3 and $k_2 = 5$ or 6.

Fig. 2 Pair of consecutive frames showing cuts for the video segment *Lecture series*

4 Experimental Results

This section presents the experimental results to reveal the success of the proposed model. We have conducted experimentation on TRECVID video database as this is one of the standard database used by many researchers as a benchmark to verify the validity of their proposed shot detection algorithms. All experiments have been performed on a *Intel CORE* i5 processor, Windows operating system with 4 GB of RAM. We performed experiments with several videos from the database and obtained better results for all the videos. Some of the results are given in Sects. 4.1 and 4.2.

4.1 Cut Detection Results

We have considered the first 10,000 frames for the video segment *Lecture series*. Figure 2 shows the pair of consecutive frames corresponding to cuts for the first 5,000 frames. The cut detection result obtained from the proposed model is shown in Fig. 3a. The sharp peaks in Fig. 3a correspond to *cuts*. The proposed model accurately detected all the 27 cuts present in the video segment. The cut detection result for the first 2,500 frames of the video segment *Mechanical works* is shown in Fig. 3b. Figure 3c shows the result for the video segment *Central valley project*.

The performance of the proposed model is evaluated using precision and recall as evaluation metrics. The precision measure is defined as the ratio of number of correctly detected cuts to the sum of correctly detected and falsely detected cuts of a video data and recall is defined as the ratio of number of detected cuts to the sum of detected and undetected cuts. Also we compared our results by performing

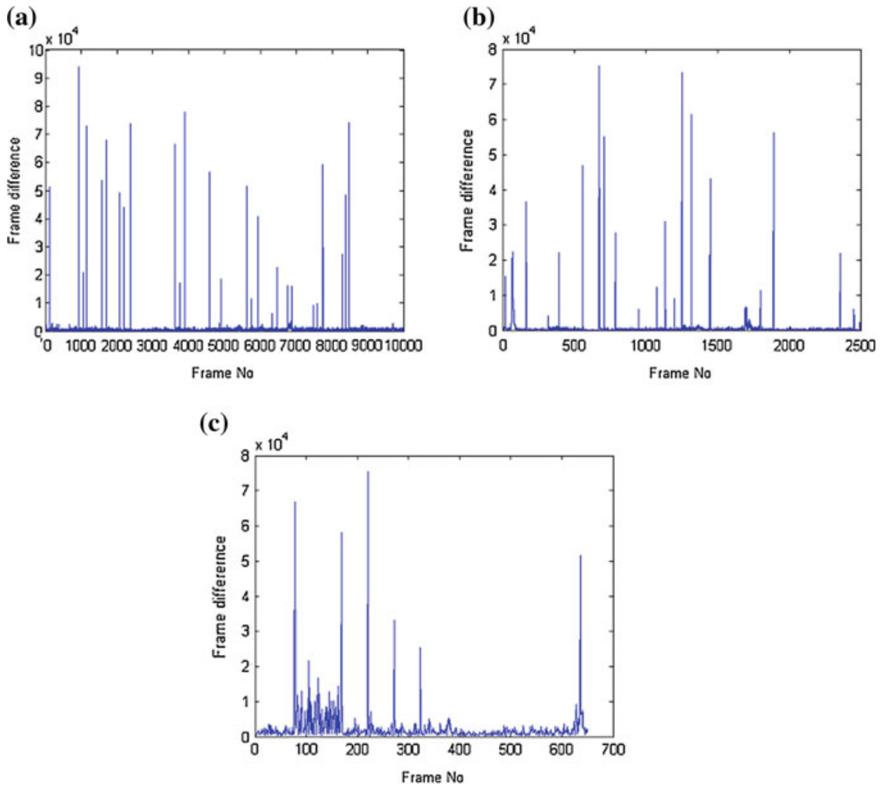

Fig. 3 Plot of *frame number* versus *frame dissimilarity measure* for the video segment: **a** *Lecture series*, **b** *Mechanical works*, **c** *Central valley project*

similar experimentation on the same video segments using other shot detection models based on *Pixel difference* [2], *Edge Change Ratio* (ECR) [21] and *chromaticity histogram* [14], and the results are reported in Table 1.

4.2 Fade Detection Results

The proposed model was tested with several videos containing fade transitions and produced satisfactory results. The fade transition regions for the *Tom and Jerry* video segment are shown in Fig. 4a. We considered the first 1,000 frames as they contained fade transitions. Figure 5 shows the frame numbers resulted from the experiment indicating the start and end of fade transitions. The fade detection result for the video segment *America's New Frontier* is shown in Fig. 4b.

The performance of the proposed model for fade detection is evaluated using precision and recall measures. These parameters were obtained for the proposed

Table 1 Precision and recall metrics of the proposed model for cut detection on TRECVID video segments

Video segment	Metrics	Proposed model	Pixel difference model [2]	ECR based model [9]	Chromaticity histogram [14]
Lecture series	Precision	1.00	0.96	0.78	1.00
	Recall	1.00	1.00	0.84	1.00
	F^1 value	**1.00**	**0.98**	**0.80**	**1.00**
Mechanical works	Precision	0.94	0.90	0.84	0.89
	Recall	0.89	1.00	0.88	0.89
	F^1 value	**0.91**	**0.95**	**0.86**	**0.89**
Central valley project	Precision	0.87	1.00	0.60	1.00
	Recall	1.00	0.50	0.43	0.75
	F^1 value	**0.93**	**0.67**	**0.50**	**0.86**

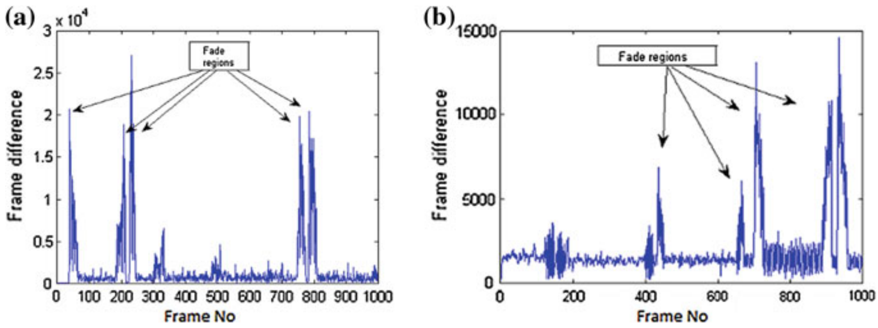

Fig. 4 Plot of *frame number* versus *frame dissimilarity measure* for the video segment: **a** *Tom and Jerry*, **b** *America's New Frontier*

Fig. 5 Pair of frames representing fade transition for the video segment *Tom and Jerry*

model on three different video segments. In order to compare our results, we have also conducted similar experimentation on the same video segments with another fade detection algorithm called *Twin comparison* method [22] and results of our model and *Twin comparison* based algorithm is reported in Table 2.

Table 2 Precision and recall metrics of the proposed model for fade detection on TRECVID video segments

Video segment	Metrics	Proposed model	Twin comparison based model
Tom and Jerry	Precision	1.000	1.000
	Recall	1.000	0.833
	F^1 measure	**1.000**	**0.909**
America's New Frontier	Precision	1.00	1.000
	Recall	1.000	0.909
	F^1 measure	**1.000**	**0.925**
Central Valley Project	Precision	1.000	1.000
	Recall	1.000	1.000
	F^1 measure	**1.000**	**1.000**

Table 3 Comparison of computation time of the proposed model with other methods

Shot detection method	Per frame feature extraction time (s)
Proposed Model	**0.0032**
Pixel difference [2]	0.0050
ECR [21]	0.1978
Chromaticity histogram [14]	0.0524

The feature extraction time taken by the proposed model and other shot detection models for a single frame are given in Table 3. It is observed from the Tables 1 and 2 that the proposed model is able to detect both hard cuts and fades. The superiority of the proposed model lies in identifying both types of shot boundaries with less computing time and produced accurate shot boundaries, and hence is suitable for video indexing and retrieval applications.

5 Conclusion

We proposed an accurate and computationally efficient model for video cut and fade detection based on cumulative color histogram. Since the dimension of the feature vector is very less, the matching time can be reduced. The proposed model produced better results for both cut and fade detection. The experimental results on TRECVID video database show that the proposed model can be used for real time video shot detection purpose.

References

1. Alattar A (1997) Detecting fade regions in uncompressed video sequences. In: IEEE international conference on acoustics, speech, and signal processing 1999 (ICASSP-97), vol 4. pp 3025–3028

2. Boreczky J, Rowe L (1996) Comparison of video shot boundary detection techniques. J Electron Imaging 5(2):122–128
3. Del Bimbo A (1999) Visual information retrieval. Morgan Kaufmann Publishers Inc., San Francisco
4. Fernando W, Canagarajah C, Bull D (1999) Automatic detection of fade-in and fade-out in video sequences. In: Proceedings of 1999 IEEE international symposium on circuits and systems (ISCAS'99), vol 4, pp 255–258
5. Fernando W, Canagararajah C, Bull D (2000) Fade-in and fade-out detection in video sequences using histograms. In: Proceedings of 2000 IEEE international symposium on circuits and systems (ISCAS 2000), Geneva, vol 4, pp 709–712
6. Guimarães SJF, Couprie M, Araújo AdA, Leite NJ (2003) Video segmentation based on 2d image analysis. Pattern Recogn Lett 24:947–957
7. Hanjalic A (2002) Shot-boundary detection: unraveled and resolved? IEEE Trans Circuits Syst Video Technol 12(2):90–105
8. Kong W, Ding X, Lu H, Ma S (1999) Improvement of shot detection using illumination invariant metric and dynamic threshold selection. In: Visual information and information systems. Springer, Berlin, pp 658–659
9. Koprinska I, Carrato S (2001) Temporal video segmentation: a survey. Signal Process: Image Commun 16(5):477–500
10. Le DD, Satoh S, Ngo TD, Duong DA (2008) A text segmentation based approach to video shot boundary detection. In: 2008 IEEE 10th workshop on multimedia signal processing, pp 702–706
11. Li S, Lee MC (2005) An improved sliding window method for shot change detection. In: SIP'05, pp 464–468
12. Lian S (2011) Automatic video temporal segmentation based on multiple features. Soft Comput—A Fusion Found Methodol Appl 15:469–482
13. Priya GGL, Dominic S (2010) Video cut detection using dominant color features. In: Proceedings of the first international conference on intelligent interactive technologies and multimedia (IITM '10). ACM, New York, pp 130–134
14. Shekar BH, Raghurama Holla K, Sharmila Kumari M (2011) Video cut detection using chromaticity histogram. Inter J Mach Intell 4(3):371–375
15. Smeaton AF, Over P, Doherty AR (2010) Video shot boundary detection: seven years of TRECVID activity. Comput Vis Image Understand 114(4):411–418
16. Truong B, Dorai C, Venkatesh S (2000) Improved fade and dissolve detection for reliable video segmentation. In: Proceedings of 2000 IEEE international conference on image processing, vol 3, pp 961–964
17. Tudor B (2009) Novel automatic video cut detection technique using Gabor filtering. Comput Electr Eng 35(5):712–721
18. Yoo H, Ryoo H, Jang D (2006) Gradual shot boundary detection using localized edge blocks. Multimedia Tools Appl 28(3):283–300
19. Yuan J, Wang H, Xiao L, Zheng W, Li J, Lin F, Zhang B (2007) A formal study of shot boundary detection. IEEE Trans Circuits Syst Video Technol 17(2):168–186
20. Yusoff Y, Christmas WJ, Kittler J (2000) Video shot cut detection using adaptive thresholding. In: Proceedings of the British machine vision conference 2000 (BMVC 2000), Bristol, 11–14 Sept 2000
21. Zabih R, Miller J, Mai K (1999) A feature-based algorithm for detecting and classifying production effects. Multimedia Syst 7:119–128
22. Zhang H, Kankanhalli A, Smoliar SW (1993) Automatic partitioning of full-motion video. Multimedia Syst 1:10–28