IJMI
International Journal of Machine Intelligence
ISSN: 0975–2927 & E-ISSN: 0975–9166, Volume 3, Issue 4, 2011, pp-371-375
Available online at http://www.bioinfo.in/contents.php?id=31

VIDEO CUT DETECTION USING CHROMATICITY HISTOGRAM

SHEKAR B.H.[1]*, RAGHURAMA HOLLA K.[1], SHARMILA KUMARI M.[2]
[1]Department of Computer Science, Mangalore University, Karnataka, India
[2]Department of Computer Science and Engineering, P A College of Engineering, Mangalore, Karnataka, India
*Corresponding Author: Email- bhshekar@gmail.com, raghu247@gmail.com, sharmilabp@gmail.com

Received: November 06, 2011; Accepted: December 09, 2011

Abstract- This paper introduces video shot detection method based on chromaticity histogram. The Chromaticity diagram provides a two dimensional representation of the image, and a corresponding two dimensional histogram can be constructed. The Horizontal Projection Profile (HPP) and Vertical Projection Profile (VPP) are obtained from the histogram. This feature vector is compared between successive frames to detect the presence of cut in the video. Experiments have been conducted on TRECVID video database to evaluate the effectiveness of the proposed model.
Key words - Shot detection, video segmentation, chromaticity histogram, horizontal projection profile, vertical projection profile.

Introduction

In recent days, as a consequence of advances in data capturing, storage and communication technology, the digital video has become an important part of many applications such as digital libraries, security, distance learning, advertising, electronic publishing, broadcasting, interactive TV, video-on-demand entertainment, 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. As video files are large in size and contain huge amount of information, it is required to organize them properly for efficient access and retrieval. Shot detection is the fundamental step in video indexing and retrieval. The video 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. Normally, it is a group of frames that have constant visual attributes, such as color, texture, and motion. In shot boundary detection, the aim is to detect the boundaries by computing and comparing similarity or difference between consecutive frames. So, the video shot detection provides a basis for video segmentation and abstraction methods [4]. The various approaches differ concerning the used features.

Shot boundaries can be broadly classified into two categories [2, 4]: abrupt/sharp shot transitions, which are also called 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 several 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 cuts.

The paper is organized as follows. Section 1 begins with the introduction. Section 2 presents the review of existing models for shot detection. The proposed model is introduced in section 3.Experimental Results and comparison with other models are presented in section 4 and conclusion is provided in section 5.

Review of existing work

We can see several models in the literature developed for video shot boundary detection. Frame level features are extracted and compared between successive frames. Each time the frame difference measure exceeds the threshold, a shot boundary is said to be detected.

We can see several models in the literature developed for video shot boundary detection. Frame level features are extracted and compared between successive frames. Each time the frame difference measure exceeds the threshold, a shot boundary is said to be detected. An easiest way of detecting the hard cut is based on pair-wise pixel comparison [2], where the differences in intensity 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. Methods based on simple pixel comparison are sensitive to object and camera movements. So an improved approach is proposed based on the use of a predefined threshold to determine the percentage of pixels that are changed [15]. This percentage is compared with a second threshold to determine the shot boundary.

As histograms are rotation invariant, histogram based methods are used in [15]. Whenever the histogram difference between two successive frames exceeds some threshold, a shot boundary is said to be detected. When the color images are considered, it is required to assign some weights to the histogram of each color component depending on the importance of color space, so weighted histogram based comparison was proposed [6].

To improve the accuracy in shot identification, low level frame features such as edges and their properties are analyzed in some of the techniques. The feature extraction can be performed either in the spatial domain or in the frequency domain. In [14], Zabih et al., (1999) proposed a method for shot detection based on analyzing edge change ratio (ECR) between consecutive frames. The percentage of edge pixels that enter and exit between 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 are sensitive to object, camera motions. To overcome this limitation, an algorithm for motion compensation was developed [15,1]. 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 [1,6]. The image is segmented into regions or blocks and the statistical measures like mean and standard deviation of pixels in the region are computed. These features are extracted from successive frames and compared using Euclidean distance or the sum of absolute differences 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 [6]. Even though likelihood ratio based methods give better results, extracting statistical features is computationally prohibitive in an unstructured environment.

In [7], Le et al., (2008) suggested a text segmentation based approach for shot detection. The combination of features such as, color moments and edge direction histogram are used for representing visual content of each frame and distance metrics are employed to identify shots in video scene. Gabor filter based approach was used for cut detection by convolving each video frame with bank of Gabor filters corresponding to different orientations [12].

In[11], dominant color features in the HSV color space are identified and a histogram is built. Block wise histogram differences are obtained and used for detecting the shot boundary. In [3], they developed a shot segmentation method based on the concept of visual rhythm. The video sequence can be viewed in three dimensions: two in the spatial coordinates and one in the temporal i.e. corresponding to frame sequence. The visual rhythm approach can be 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. The combinations of various features were used to accurately detect cuts [9]. The inter-frame difference values were computed separately for each feature to address different issues such as flash light effect, object or camera motion. The features include pixels, histogram, motion features etc.

Although several techniques are available for shot detection, the proposed technique is different from the existing approaches in terms of reducing time complexity without losing the accuracy and hence suitable for real time video database processing applications. In the proposed work, we focused on hard cuts. The details of the proposed model are given below.

Proposed Model

The proposed model is based on the projection profile of two dimensional chromaticity histogram obtained by transforming the image into *XYZ* color space [10]. The image obtained from each video frame is transformed from *RGB* to *XYZ* color space as follows:

$$\begin{aligned} X &= 0.607 * R + 0.174 * G + 0.200 * B \\ Y &= 0.299 * R + 0.587 * G + 0.114 * B \\ Z &= 0.066 * R + 0.111 * B \end{aligned} \quad (1)$$

The *(x, y)* chromaticities can be derived from the *XYZ* color space using the following transformation:

$$x = \frac{X}{X+Y+Z} \quad (2)$$

Similarly, we can define, $z = \frac{Z}{X+Y+Z}$ but it does not contain useful information since $x + y + z = 1 \Rightarrow z = 1 - x - y$.

We obtain the Chromaticity diagram, which is a 2D representation of an image where each pixel produces a

pair of *(x, y)* values. So, for the given image *I* of dimension L_x, L_y, the trace of the chromaticity diagram is defined as follows:

$$T(x,y) = \begin{cases} 1 & if\ \exists (i,j): I(i,j)\ producing\ (x,y) \\ 0 & otherwise \end{cases} \quad (3)$$

$$0 \le i \le L_x - 1, 0 \le j \le L_y - 1$$

There is a possibility that more than one pixel may produce the same *(x, y)* pair. So, we have a 2 dimensional Histogram associated with the chromaticity diagram.

$$D(x,y) = count\ of\ pixels\ producing\ (x,y). \quad (4)$$

We compute horizontal and vertical projection profiles for the Histogram *D*. The horizontal projection profile is defined as the vector of the sum of the pixel intensities over each row denoted by HPP_D. Similarly, the column projection profile is defined as the vector of the sum of the pixel intensities over each column denoted by VPP_D. The feature vector for frame n is given by:

$$F_n = [HPP_D, VPP_D] \quad (5)$$

For better accuracy, we compute the second order frame difference. The frame difference between two consecutive frames is calculated as follows:

$$D(n) = d(F_n, F_{n-1}) - d(F_{n-1}, F_{n-2}) \quad (6)$$

where,

$$D(n) = d(F_n, F_{n-1}) - d(F_{n-1}, F_{n-2}) \quad (7)$$

Where *N* is the length of the feature vector, i.e. *N=200*. 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. The difference vector *D* containing all the consecutive frame differences is compared against a threshold to detect the cut. Selecting a single global threshold is not adequate to detect all the cuts present in the video. Several approaches have been used for *adaptive thresholding* [13, 8]. Here, we use the *local adaptive thresholding* technique as suggested in [5]. The threshold *T* is calculated as follows:

$$T = \frac{\sum_{i=c+1}^{f_c - 1} \mathbf{D}(i)}{f_c - c + 1} w \quad (8)$$

where *c* is the frame number of previous cut, f_c is the current frame number, *w* represents the threshold weighting. Before calculating *T*, all the frame difference values which are below 10% of the maximum value in *D* are suppressed. In our experiments *w* = 10, gave better result for all the videos tested. It is possible to estimate the value of *w* automatically [5]. We have fixed the value of *w* as 10 empirically.

Experimental Results

This section presents the results of the experiments conducted to validate the success of the proposed model. We have conducted experimentation on TRECVID video database. We specifically chose this 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 models. All experiments are performed on a P-IV 2.99GHz Windows machine with 2GB of RAM. The proposed model has been tested with several videos from the database and produced better results for all the videos. Some of the results have been included in this paper.

Experimentation on senses111.mpeg video: We have considered first 5000 frames. Fig. (1) shows the pair of consecutive frames corresponding to cuts taken from the video segment *Senses111.mpeg*.The cut detection result obtained from the proposed model is shown in Fig. 2(a) There is a cut in frame numbers: 128, 933, 1063, 1167, 1569, 1703, 2082, 2202, 2370, 3615, 3757, 3891, 4584 and 4911 as obtained from the ground truth data. It shall be observed from Fig. 2(a) that the frame difference becomes high for these frames. The proposed model detected all 14 cuts.

Fig. 1. Pair of consecutive frames showing cuts for the video segment *Senses111. mpeg*

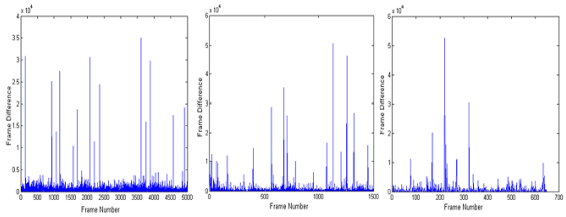

Fig. 2. Plot of frame number vs. frame dissimilarity measure for the video segment (a)senses111.mpeg (b) BOR03.mpeg (c) BOR08.mpeg

Experimentation on BOR03.mpeg video: We have considered the first 1500 frames. The cut detection results are shown in Fig. 2(b). From the ground truth data, the cuts are seen in frame numbers: 165, 317, 391, 561, 677, 708, 787, 951, 1075, 1135, 1198, 1252, 1317 and 1448. The frame numbers obtained from the experimental results indicating cuts are, Frame numbers: 19, 64, 165, 317, 391, 561, 677, 708, 787, 951, 1075, 1135, 1198, 1252, 1317, and 1448.

Experimentation on BOR08.mpeg video: We have considered first 650 frames. There is a cut in frame numbers: 77,168, 220,271,322 and 635 as seen from the ground truth data. The cut detection result is shown in Fig. 2(c). The shot cuts obtained from the experimental results are, Frame numbers: 77,168,220,271, 322, 340 and 635.

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 similar experimentation on the same video segments using other models based on Pixel difference [1], Colour Histogram [6], Edge Change Ratio (ECR) [14], chromaticity moments [10] and the results are reported in Table-1.

Table-1: *Precision and Recall metrics of the proposed model for cut detection on TRECVID video segments*

Video Segment	Metrics	Proposed model	Pixel difference based model [1]	Colour histogram based model [6]	ECR based model[14]	Chromaticity moments[10]
Senses111	Precision	1.000	1.000	1.000	0.632	0.867
	Recall	1.000	0.929	1.000	0.857	0.929
	Combined measure (F_1)	**1.000**	**0.963**	**1.000**	**0.727**	**0.897**
BOR03	Precision	0.875	0.933	0.875	1.000	0.889
	Recall	1.000	1.000	1.000	0.930	0.615
	Combined measure (F_1)	**0.933**	**0.966**	**0.933**	**0.964**	**0.727**
BOR08	Precision	0.857	1.000	0.750	0.600	0.714
	Recall	1.000	0.500	1.000	0.429	0.833
	Combined measure (F_1)	**0.923**	**0.667**	**0.857**	**0.500**	**0.769**

Further, to demonstrate the computational efficiency, computing time taken by the proposed model for feature extraction (match score) for any single frame is given in Table-2. We have also given in Table-2, the computing time taken by the other models.

Table-2: *Comparison of computation time of the proposed model with other methods*

Shot detection method	Per frame feature extraction time (sec.)
Proposed Model	0.0524
Gabor Filtering [12]	0.3647
Pixel difference[1]	0.0050
Colour Histogram[6]	0.0219
ECR[14]	0.1978
Chromaticity moments[10]	0.0545

Conclusion

We have developed an accurate model for video cut detection based on Horizontal Projection Profile (HPP) and Vertical Projection Profile (VPP) of the chromaticity histogram. The proposed model produced better results for cut detection. The experimental results on TRECVID video database show that the proposed model can be used for video cut detection purpose

References

[1] Boreczky J., Rowe L. (1996) *Journal of Electronic Imaging* 5(2), 122_128.

[2] Del Bimbo A. (1999) *Morgan Kaufmann Publishers Inc.*,San Francisco, CA, USA.

[3] Guimarães S. J. F., Couprie M., Araújo A. D. A., Leite N. J. (2003) *Pattern Recognition Letters*. 24, 947-957.

[4] Hanjalic A. (2002) *IEEE Transactions* on 12(2), 90 -105.

[5] Kim W.H., Moon K.S., Kim and J.N. (2009) *In: Advanced Communication Technology*, *ICACT 2009. 11th International Conference on (2009)*

[6] Koprinska I., Carrato S. (2001)*Image Communication* 16(5), 477 _ 500.

[7] Le D. D., Satoh S., Ngo T. D., Duong D.A. (2008) *In: Multimedia Signal Processing, 2008 IEEE 10th Workshop* on. pp. 702 -706.

[8] Li S., Lee M. C. (2005) *In: SIP'05*. pp. 464-468.

[9] Lian S. (2011) *Soft Computing* - A Fusion of Foundations, Methodologies and Applications 15, 469-482.

[10] Paschos G., Radev I., Prabakar N. (2003) *Knowledge and Data Engineering, IEEE Transactions* on 15(5), 1069-1072.

[11] Priya G. G. L., Domnic S. (2010) *In: Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*. pp. 130-134. *IITM '10, ACM, New York, NY, USA* .

[12] Tudor Barbu (2009) *Computers & Electrical Engineering* 35(5), 712-721.

[13] Yusoff Y., Christmas W.J., Kittler J. (2000) *In: Proceedings of the British Machine Vision Conference 2000, BMVC 2000*, Bristol, UK, 11-14 .

[14] Zabih R., Miller J., Mai K. (1999) *Multimedia Syst*. 7, 119-128.

[15] Zhang H., Kankanhalli A., Smoliar S.W. (1993) *Multimedia Systems* 1, 10-28.