Shot Boundary Detection Using Sorted Color Histogram Polynomial Curve

Awais Adnan, Muhammad Ali

Computer Department, Institute of Management Sciences Peshawar, Pakistan
Email: awaisadnan@gmail.com

Abstract: Multimedia is going more and more popular day by day and the major reason is the rapid advancement in technology and continuous reduction in price. Boundless amount of data in all media formats are now available locally in our own computers and online over the internet, most of these are however unorganized and scattered. There is an emerging need of classification, summarization and indexing on multimedia data but because of its diverse and heterogenic nature no remarkable achievement could be achieved so far. Shot detection is the basic and probably the most important step in almost all video processing activities. Shot is a group of frames sharing some spatial features captured by a single camera action. Lot of work is done in this domain but no single solution can solve the problem completely. Different video editing techniques like zooming, fade-in, fade-out and gradual translations make the problem even more crucial. In this work, we are presenting a different approach of finding shot boundaries using third order polynomial cure. In this method instead of using multiple frames, features are extracted from single frames; these features collected for boundary detection can also be used efficiently for shot classification.

[Awais Adnan[1], Muhammad Ali[1]. **Shot Boundary Detection Using Sorted Color Histogram Polynomial Curve.** *Life Sci J* 2013;10(4):1965-1972]. (ISSN:1097-8135). http://www.lifesciencesite.com. 260

Keywords: polynomial curve; Video scene; shot boundary; histogram.

1. Introduction

Because of the advancement in capturing device and low capacity-price ratio of storage, multimedia and especially videos are getting popularity very rapidly and now becomes an essential part of our today's digital life.

Digital video can be considered as collection of various frames whereas group of connected frames that are captured by a single camera action is called a shot. Usually frames in a shot share some visual similarities that can be identified by low-level feature analysis. Any shot can be summarized as one frame known as key-frame. A scene is a collection of connected shots having some sematic similarity. Scene is a part of story that contains many shots sharing semantics and contents. Story contains scenes having a logical relationship. Same story may be disjoined, scattered over the length of vides segment. Stories are present in structured videos only while other types of unstructured videos do not contain this element however by relaxing the typical concept of stories, we can still find some types of stories in many such videos. Meng Wang and Hong-Jiang Zhang (Wang & Zhang, 2009) arranged these unite of video in a hierarchy that can be shown as Figure 1.

Shot detection is the most important and usually the first step of any video processing applications. Process of shot boundary detection can be analyzed as mixture of three core elements: content representation, evaluation of continuity function and classification of this continuation function (I. Yuan et al., 2007) however topographies of the later stages are determined by the first element in this mix.

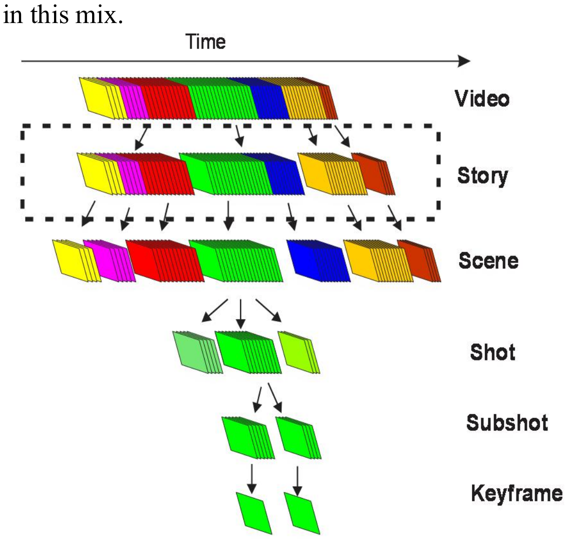


Figure 1: Video contents representation and decomposition in a hierarchy

Sensitivity that is the level of detail to be captures, and in variances, opposite on sensitivity, the tolerance to change in the feature, are two factors influence the selection of contents to be used for shot detection. Trade-off between these two is important decision to achieve the required performance. At one extreme, all the pixels in an image can be taken as feature. This option is more sensitive but has less tolerance to changes geometric transformations. Moreover, this need more computation time. On the other hand some features can be extracted from set of frames to detect the changes in continuatively function. Many features are used by researchers to detect shot boundaries in video that can be grouped into four major categories i.e. text based, visual based, audio based, and combination where all of these three are used.

In most of the solutions presented so far features are collected from a group of frames, mostly two neighboring frames, as the difference in some attributes however we in this paper are presenting a different approach where features are extracted from frame directly rather than the difference of the frames. This approach successfully detects cut boundaries and can be used for shot classifications and scene identifications.

2. Related Work

As mentioned above shot boundaries detection is an important step in different video processing activities where lot of work is done in this area especially in last decade after the launch of TRECVID (TREC Video Retrieval Evaluation) in 2001. All work in this area can broadly be divided into four major categories i.e. pixel based methods, histogram based methods, edge based methods, and methods based on temporal features.

Pixels are probably the oldest and the simplest features used to detect shot boundaries in videos. Shot are detected by identifying continuity and discontinuity points R and $\overline{R}$ in a signal generated by the comparing of corresponding pixels in a set of frames, mostly the neighboring frames. one of the pioneers, used pixels to detect shot boundaries are Kikukawa and Kawafuchi (Kikukawa & Kawafuchi, 1992) who calculated absolute difference of all the corresponding pixels of adjacent frames and used their mean for continuity function. This technique requires more computation and is highly sensitive for camera motion and geometric transformations. In summarized form this work can be represented by following equation.

$$z(k) = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y}|I_k\,(x,y) - I_{k+l}(x,y)| \quad (1)$$

where $\mathbf{I_k}\,(\mathbf{x},\mathbf{y})$ is the intensity value of pixels at location $(\mathbf{x},\mathbf{y})$ in frame k and $\mathbf{l} \geq \mathbf{1}$ is the distance of frames neighbourhood. Otsuji et al.(Otsuji, Tonomura, & Ohba, 1991) improve this concept and uses only those pixels for calculation that have intensity greater than a threshold. similar approaches are used by Zhang et al. (Zhang, Kankanhalli, & Smoliar, 1993) and some others (Nagasaka & Tanaka, 1992; Yeo & Liu, 1995) with few small modification to reduce computation complexity and time. Gyo Chung et.al (Chung, Kim, & Song, 2000) presented the concept of visual rhythm in pixel based methods. All these techniques share common problem of high sensitivity and less invariance.

To solve this problem some researchers, instead of using pixels directly, used other statistical techniques like histogram for shot boundaries detection. These techniques ignores the small changes in spatial information however for some editing effects and especially for changes in light and contrast affects, its sensitivity remains high. Histogram based methods for gray-level and color image can be summarized as following equations (2) and (3).

$$z(k) = \sum_{j=1}^{N_{Bins}} |H_k(j) - H_{k+1}(j)| \quad (2)$$

$$z(k) = \sum_{C}\sum_{j=1}^{N_{Bins}} \left|H_k^C(j) - H_{k+1}^C(j)\right| \quad (3)$$

Where $\mathbf{H_k(j)}$ is the number of pixels for intensity level **j** in frame **k**. similarly $\mathbf{H_k^c(j)}$ in value in histogram for frame **k** in color channel **c** and **j** is the intensity level.

M. Cooper (Cooper, 2004) uses $\chi 2$ test to compare histogram of adjacent frames to detect shot boundaries between them with a better accuracy then other histogram based methods (Choubeya & Raghavanb, 1997). In most of the histogram based techniques spatial information are ignored which is tried to solve by Hongjiang Zhang at.al (Zhang, Low, & Smoliar, 1995) by dividing frames into 16 blocks. Histogram is calculated and compared for each block to detect shot translation and other effects in the video.

Liu et al. (Liu, Liu, Ren, & Chan, 2008) use histogram based techniques for shot boundary detection in their work for video summarization. Local color histogram is used as feature for shot boundary detection. To improve the performance each frame is divided into 4 by 4 sub mages. 16-bin color histogram in HSV space is then computed for each sub image of the same size and shape. MPEG-7 HSV color model is used in this work by combining 16 bins each from 16 sub-images; as a result feature-vector of l256 length is obtained. Difference and accumulative difference are used for decision.

Edges are another features used by many researchers for boundary detection in videos. By applying some mask first edge are identified and then using these edges of neighbor frames continuity function is formulated by using two important features i.e. Edge Change Ratio (ECR) and Edge-based Contrast (EC) (Ahmed, Karmouch, & Abu-Hakimaˆ, 1999; Zabih, Miller, & Mai, 1995).

Different types of translations and geometric transformation can easily be handled by techniques based on edge but these methods are generally multi-phase, more complex and involve more computations. Rainer Lienhart (Lienhart, 1999) improve this basic techniques by using edge based contrast features that reduces computational complexity.

Besides features from spatial domain, some researchers have used temporal features like motion for shot translations. Akutsu, et al in their research (Akutsu, Tonomura, Hashimoto, & Ohba, 1992) used motion smoothness for this purpose using a threshold where Jinhui (J. Yuan, Li, Lin, & Zhang, 2005) used other temporal information with Support Vector Machine (SVM) for classification. Many other classification techniques are discussed by M. Albanese, et al. in (Albanese, Chianese, Moscato, & Sansone, 2004). That can be used to classify temporal features of videos.

Sleit et al. (Sleit, Almobaideen, Qatawneh, Al-Asir, & Al-Megdadi, 2008) use objects to understand the type of movies, moments in movies and activities. 1st they identify objects and observe the motion of different components.

D. Borth, et al. (Borth, Ulges, Schulze, & Breuel, 2008) use MPEG-7 color layout descriptor (CLD) as feature for shot boundary detection. First, these color and texture features are calculated for each frames and then difference for these consecutive frames are calculated. Adaptive threshold is used to detect shot boundaries. This method works best for cut shot however, for fade shots and other transitions, this shows poor results. After shot boundary, they detected key-frame by applying k-mean algorithm on CLD features extracted for shot boundaries. Bayesian Information Criterion (BIC) is used to estimate the number of clusters.

Hsiao et al (Hsiao, Huang, Tsai, & Chiang, 2010) use DCT to extract features for image retrieval using contents. Using YUV color space, they first divide image into number of regions their marking locations. Local and global matching are used for searching. This object based CBIR system give improved results.

People counting is used in (Arif, Saqib, Basalamah, & Naeem, 2012) for video surveillance. this proposed system works for crowd management. First background and foreground are identified and separated. Foreground contain people that are separated using texture analysis. For classification they used neural network that give about 96% accuracy.

3. Shot Boundary Detection using sorted histograms

In this method, features are extracted from two visual channels Chroma channel that represent light and Luma channel that represent color information of the frame. Novelty of this method is the reduction in the three mappings that are supposed to be fundamental steps of the process (Hanjalic, 2002, 2004; I. Yuan et al., 2007) as shown in table 1. In this method feature extracted from the frame, construct the continuity function without any additional step of mapping from feature-space into continuity-space. By eliminating the mapping steps, features extracted at frame level can be used directly for decision space. Same single value can be used for length of the windows while in most of other proposed method this value has to recalculated for each window length.

TABLE 1: Novelty of The Proposed Solution

Three mappings in standard solution	Two mappings in proposed solution
1. Ψ : $Q \to F$, $Q_t \to V_t$	1. Ψ : $Q \to F$, $Q_t \to V_t$
2. φ : $F^{2\times D} \to S$, $A_t^d \to s_t$	2. Φ : $\boldsymbol{F^{2\times r+1} \to W}$, $\boldsymbol{A_t^d \to w}$
3. Ξ : $S^{2\times r+1} \to W$, $B_t^r \to w$	

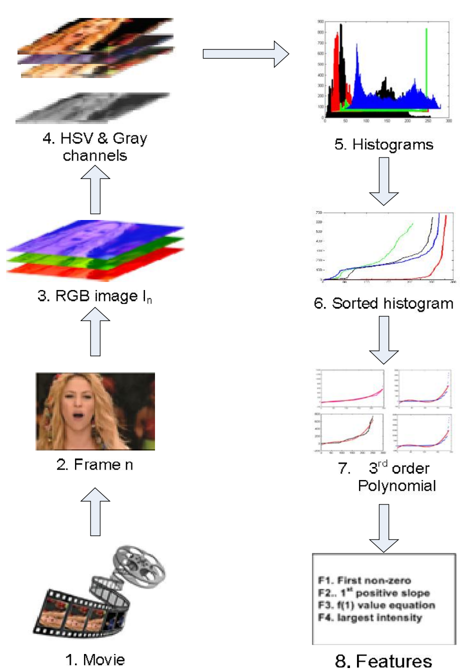


Figure 2: Graphical summary of the algorithm

To extract Chroma information, HSV color model is used while for Luma information are extracted form gray version on the frame. Selection of these two channels separately makes this solution compatible with most of the component and composite video signals like S-video, YP_bPr, and RGB.

Main steps of the proposed method can be summarized as shown in Figure 2. All the entities can be grouped into four main steps.

a. Image Decomposition into Luma and Chroma

In first step movie is taken frame by frame and then scaled into size 240×180. This size is selected because this maintains the aspect ratio of 4×3 and contains total pixels 43200 that can easily be fitted into 16-bit integer, the default size in most of the programming tool and hardware architecture. This step is shown in Figure 3 and can be summarized in following two equations.

$$Movie() \xrightarrow{for\ Frame = t} f_t \tag{4}$$

$$\Re(f_t) \xrightarrow{240 \times 18(} I_t \tag{5}$$

Normalized image that is in RGB format is then converted into Chroma and Luma channels. For Chroma RGB image is converted into HSV channels using following equations, **(6)**, **(7)**, and (8).

$$H(x,y) = \cos^{-1}\frac{\frac{1}{2}\left(2R_{(x,y)} - G_{(x,y)} - B_{(x,y)}\right)}{\sqrt{\left(R_{(x,y)} - G_{(x,y)}\right)^2 - \left(R_{(x,y)} - B_{(x,y)}\right)\left(G_{(x,y)} - B_{(x,y)}\right)}} \tag{6}$$

$$S_{(x,y)} = \frac{\max\left(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}\right) - \min\left(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}\right)}{\max\left(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}\right)} \tag{7}$$

$$V_{(x,y)} = \max\left(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}\right) \tag{8}$$

For x = 1.. 240 and y = 1.. 180

For Luma i.e. brightness information RGB image is converted into gray-level using the basic relation of the equation (9). This equation use lot of floating point computation that slows down the process where we have to perform 43200 calculations in each frames, one for each pixel. To speed up the process we have actually used the modified version of basic equation of weighted sum that involves only integer calculations. We have added 500 in equation (10) to get the result of division in round figures (Jesus, Guillermo, M., M, & Julian, 2005).

$$\mathrm{Gray} = \mathrm{R} * 0.299 + \mathrm{G} * 0.587 + \mathrm{B} * 0.114 \tag{9}$$

$$Gray_{(x,y)} = \frac{[R_{(x,y)} * 299 + G_{(x,y)} * 587 + B_{(x,y)} * 114] + 500}{1000} \tag{10}$$

Where x = 1.. 240 and y = 1.. 180

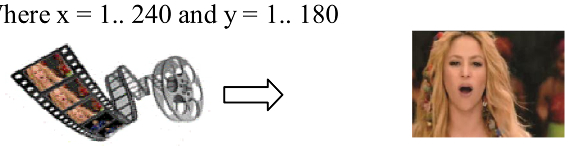


Figure 3. Frame from Video

Image in normalized size is actually in RGB format that is converted into Chroma and Luma domain. for Chroma RGB image is converted into HSV color model using following equations (11), (12) and (13). For Images from mono-chromic source, this remains blank.

$$H(x,y) = \cos^{-1}\frac{\frac{1}{2}\left(2R_{(x,y)} - G_{(x,y)} - B_{(x,y)}\right)}{\sqrt{\left(R_{(x,y)} - G_{(x,y)}\right)^2 - \left(R_{(x,y)} - B_{(x,y)}\right)\left(G_{(x,y)} - B_{(x,y)}\right)}} \tag{11}$$

$$S_{(x,y)} = \frac{\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) - \min(R_{(x,y)}, G_{(x,y)}, B_{(x,y)})}{\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)})} \quad (12)$$

$$V_{(x,y)} = \max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) \quad (13)$$

For x = 1. 240 and y = 1. 180

For Luma range RGB image is converted into gray-level. for this conversion a modified version on equation (14) is used.

$$\text{Gray} = \text{R} * 0.299 + \text{G} * 0.587 + \text{B} * 0.114 \quad (14)$$

This equation use lot of floating point computation that slows down the process where we have to perform 43200 calculations in each frames, one for each pixel. To speed up the process we have actually used the modified version of basic equation of weighted sum that involves only integer calculations. We have added 500 in equation (15) to get the result of division in round figures (Jesus et al., 2005).

$$Gray_{(x,y)} = \frac{[R_{(x,y)} * 299 + G_{(x,y)} * 587 + B_{(x,y)} * 114] +}{1000} \quad (15)$$

Where x = 1.. 240 and y = 1.. 180

This process can be summarized as Figure 4 below.

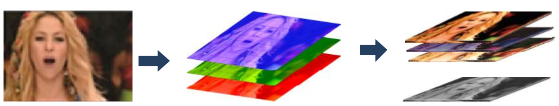


Figure 4. Decomposition into HSV and Gray-level

b. Sorted Histogram and 3rd order polynomial

In next step histograms for each of the four channels using 256 bins are calculated, one for each intensity level.

$$\begin{aligned} H_h(k) &= N_{k,h} \\ H_s(k) &= N_{k,s} \\ H_v(k) &= N_{k,v} \\ H_{Gray}(k) &= N_{k,Gray} \end{aligned} \quad (16)$$

Where $H_c(k)$ is histogram in channel c that is number of pixels on k^{th} level in channel c. then these histograms are sorted in non-decreasing order such that $H_n \leq H_{n+1}$ for all n = 1.. 256

$$\begin{aligned} SH_h &= SORT(H_h) \\ SH_s &= SORT(H_s) \\ SH_v &= SORT(H_v) \\ SH_{Gray} &= SORT(H_{Gray}) \end{aligned} \quad (17)$$

Where SORT is function that arrange H_c such that $H_c(n) \leq h_c(n+1)$ for all n where c is the channel under consideration.

In next step a third order polynomial curve is fitted from the sorted histogram. Third order curve is selected because it is the nearest approximation of the sorted histogram. best fit 3rd order polynomial curve us selected using least-square error method. General form of mth order polynomial curve can be represented as equation (18).

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + .. + a_m x^m \quad (18)$$

Where a_0, a_1,.. a_m are unknown coefficients

To obtain minimum square error, first order derivative must be zero (Adnan, Ali, & Dar, 2007). These unknown coefficients can be obtained by solving fowling equations.

$$\left| \begin{aligned} \frac{\partial \Pi}{\partial a_0} &= 2 \sum_{i=1}^{n} [y_i - (a_0 + a_1 x_i^1 + a_2 x_i^2 + .. + a \\ \frac{\partial \Pi}{\partial a_1} &= 2 \sum_{i=1}^{n} x_i [y_i - (a_0 + a_1 x_i^1 + a_2 x_i^2 + .. + \\ \frac{\partial \Pi}{\partial a_2} &= 2 \sum_{i=1}^{n} x_i^2 [y_i - (a_0 + a_1 x_i^1 + a_2 x_i^2 + .. + \\ &\qquad .... \\ \frac{\partial \Pi}{\partial a_m} &= 2 \sum_{i=1}^{n} x_i^m [y_i - (a_0 + a_1 x_i^1 + a_2 x_i^2 + .. + \end{aligned} \right. \quad (19)$$

By solving above equations for m = 3, we obtained 3rd order polynomial curve as

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \; where \; x = 1, 2..255 \quad (20)$$

Results of this step in four channels are shown in Figure 5.

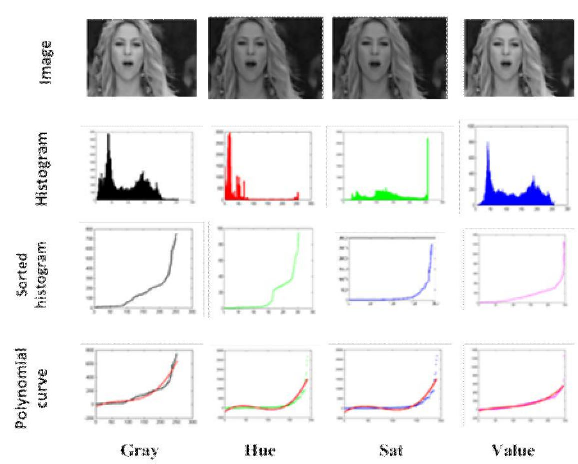


Figure 5. Polynomial curve fitting

c. Feature extraction

At the end we extracted our feature set. Our feature set F contains four elements that can be simplified as.

$$F = \{f_1, f_2, f_3, f_4\} \quad (21)$$

Details in these fours elements are as

$\boldsymbol{f_1}$: First feature is the 1st non-zero value in the sorted histogram curve. $y = f(x) fror\ x = 1..256$ then f_1 = n such that $f(n) \neq 0\ and\ f(n-1) = 0$.

$\boldsymbol{f_2}$: Second feature is the first intensity level in sorted curve having non-zero slope. If $\frac{dy_k}{dx}$ is slope at *k* then our second feature is *n* such that

$$\frac{dy_n}{dx} \neq 0\ and\ \frac{dy_{n-1}}{dx} = 0 \quad (22)$$

In our case $\Delta x = x_{n+1} - x_n = 1$ and $y_{n+1} > y_n$ as curve is in ascending order hence $\boldsymbol{f_2}$ can simply be calculated as $\boldsymbol{f_2} = n$ such that $(y_{n+1} - y_n) > 0$ and $y_n \neq 0$

$\boldsymbol{f_3}$ **:** Third feature is value of the polynomial curve foe x = 1 i.e *f(1)* as shown in equation (23).

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (23)$$
$$f(1) = a_0 + a_1 + a_2 + a_3$$

We selected f(1) for this feature because shapes of all the curves for other higher valve of x is almost similar as for x = 1 but calculating value of polynomial for x = 1 is simple, its only addition of the four coordinates of the ploy.

$\boldsymbol{f_4}$**:** Final feature point is the of the intensity level with highest value in the histogram curve. if $H_{k,c}$ is the value at k in histogram for channel c then $\boldsymbol{f_4}$ for this channel is $H_{k,c}$ if $H_{k,c} \geq H_{n,c}$ for all n = 1.. 256. And c = 1, 2 & 3

To generate continuity function we combine these four features according to following equation.

$$F = [w_1f_1 + w_2f_2 + w_3f_3]\, f_3^{w_3} \quad (24)$$

d. Shot boundary identification

Shot boundary is identified where there is a sudden change in the continuation function F composed of four features. Boundary is identified at point where absolute difference in the continuation function is above threshold. Δf is the absolute difference as

$$\Delta f_n = |F_{n+1} - F_n| \quad (25)$$

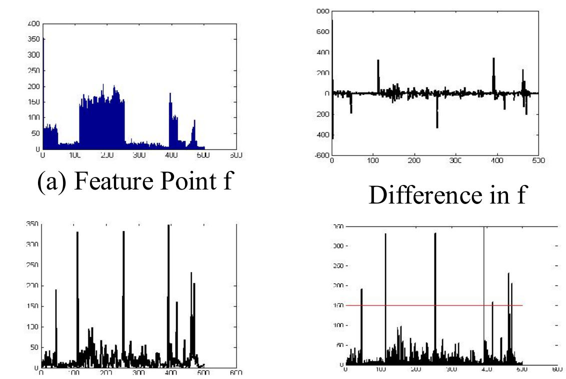
and the point $\delta(f)$ where this difference is above threshold is call the shot boundary.

$$\delta(f) = \begin{cases} Yes, \Delta f \geq th \\ No, \Delta f \geq th \end{cases} \quad (26)$$

Where th is the threshold value. Continuation function and threshold value is shown in Figure 6.

4. Results and conclusion

Precision and recall are used to evaluate the performance of this proposed algorithm. Precision in the number of shot boundaries detected by the algorithm that are actually boundaries (true-positive) divide by the total boundaries detected. Recall on the other hand is the true -positive values divides by the total values calculated by the algorithm (Buckland & Gey, 1994).



(a) Feature Point f (b) Difference in f

(c). Absolute difference (d) Thresholding

Figure 6. Feature using thresholding on continuity function

To evaluate performance of this proposed algorithm for precision and recall, group of videos are selected, some of which are local and are manually tagged while some videos obtained from National Institute of Standards and Technology (NIST, http://www.nist.gov) project. As in both cases only cut-shots are marked, we used only this type of scene transitions for evaluation. TABLE 2:shows the performance of this proposed method.

This table shows that on average more than 98% recall is obtained and even in some long videos it's about 100%. Precision is about 82%, this low value is due to the fact that this is a sensitive method that detect gradual translation as series of frame in transition while manual results used for this paper contain only cut shots.

Figure 7 shows the sequences of frame near shot boundaries and show the shot boundaries.

a.
b.
c.
d.
e.
f.
g.
h.
i.

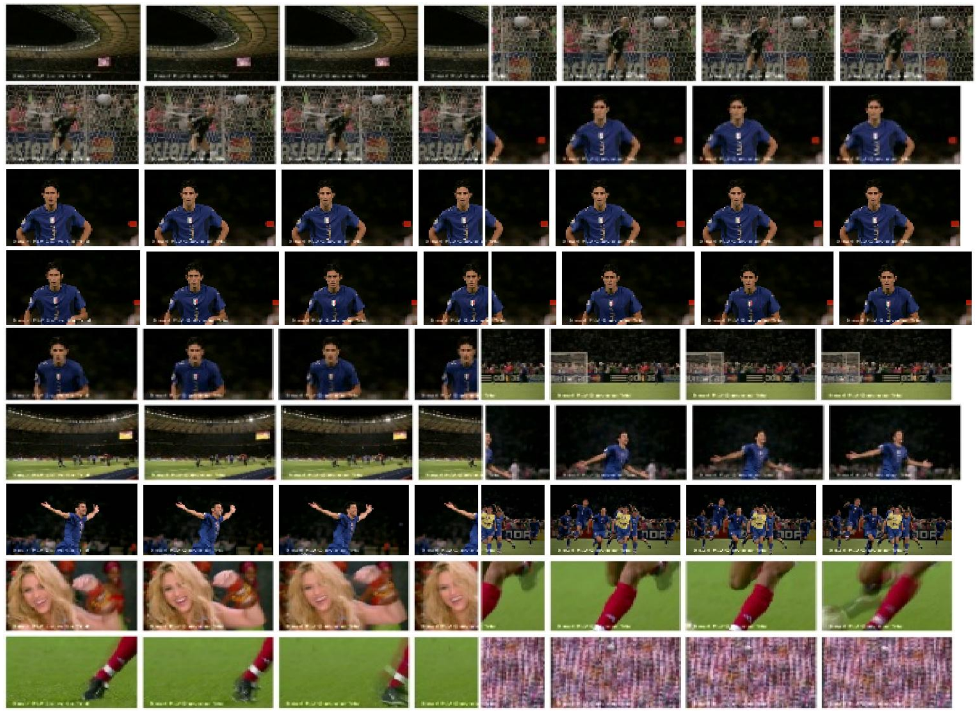


Figure 7. Shot Boundaries if frame sequences

TABLE 2: Precision and Recall, for 5 videos using Sorted Histograms

Video	Number of frames	Shots in video	Shots using sort Hist	True Positive	False Positive	False Negative	Precision	Recall
Wild_Life	902	6	6	6	0	0	100.00%	100.00%
Waka_1000	1000	31	38	30	8	1	78.95%	96.77%
Roman_02	1403	32	39	30	9	2	76.92%	93.75%
Roman_01	1491	29	40	29	11	0	72.50%	100.00%
BG_34901	28000	179	214	179	35	0	83.64%	100.00%
Average							82.40%	98.10%

Corresponding Author:
Awais Adnan
Computer Department
Institute of management Sciences
Peshawar, 25000, Pakistan
E-mail: awaisadnan@gmail.com

Reference:

1. Adnan, Awais., Ali, Mumhammad, & Dar, Amir Hanif. (2007). Image Search by Features of Sorted Gray Level Histogram Polynomial Curve. *International Journal of Applied Mathematics and Computer Sciences, 4*, 280-285.
2. Ahmed, Mohamed, Karmouch, Ahmed, & Abu-Hakimaˆ, Suhayya. (1999). Key frame extraction and indexing for multimedia databases. *Proc. Vis. Interface Conf. 1999*, 506-511.
3. Akutsu, Akihito, Tonomura, Yoshinobu, Hashimoto, Hideo, & Ohba, Yuji. (1992). *Video indexing using motion vectors.* Paper presented at the Proceedings of SPIE Visual Communications and Image Processing,.
4. Albanese, Massimiliano, Chianese, Angelo, Moscato, Vincenzo, & Sansone, Lucio. (2004). A formal model for video shot segmentation and its application via animate vision. *Multimedia Tools Appl., 24*(3), 253-272.
5. Borth, Damian, Ulges, Adrian, Schulze, Christian, & Breuel, Thomas M. (2008). *Keyframe Extraction for Video Tagging &*

Summarization. Paper presented at the Informatiktage 2008. Fachwissenschaftlicher Informatik-Kongress.

6. Buckland, Michael, & Gey, Fredric. (1994). The Relationship between Recall and Precision. *Journal of The American Society For Information Science, 45*(1), 12-19.
7. Choubeya, Suresh K., & Raghavanb, Vijay V. (1997). Generic and fully automatic content-based image retrieval using color. *Pattern Recognition Letters, 18*(11), 1233–1240.
8. Chung, Min Gyo, Kim, Hyeokman, & Song, S.M.-H.. (2000). *A scene boundary detection method*. Paper presented at the Image Processing, 2000. Proceedings. 2000 International Conference on.
9. Cooper, Matthew. (2004). *Video segmentation combining similarity analysis and classification.* Paper presented at the Proceedings of the 12th annual ACM international conference on Multimedia MULTIMEDIA '04, New York
10. Hanjalic, Alan. (2002). Shot boundary detection: unraveled and resolved? *IEEE transactions on circuits and systems for video technology, 12*(2), 90-105.
11. Hanjalic, Alan. (2004). *Content-Based Analysis of Digital Video* (Springer; 2004 edition ed. Vol. 1). Boston: Springer
12. Jesus, Bescos, Guillermo, Cisneros, M., Martinez José, M, Menendez José, & Julian, Cabrera. (2005). A unified model for techniques on video shot transition detection. *IEEE Transactions on Multimedia, 7*(2), 293-307.
13. Kikukawa, T., & Kawafuchi, S. (1992). Development of an automatic summary editing system for the audio visual resources. *Transactions of the Institute of Electronics, Information and Communication Engineers, J75-A*(2).
14. Lienhart, Rainer. (1999). Comparison of automatic shot boundary detection algorithms. *In Storage and Retrieval for Image and Video Databases, No. SPIE 3656.*, 290-301.
15. Liu, Yang, Liu, Yan, Ren, Tongwei, & Chan, Keith C. C. (2008, October 31). *Rushes Video Summarization using Audio-Visual Information and Sequence Alignment.* Paper presented at the 2nd ACM TRECVid Video Summarization Workshop TVS '08, Vancouver, British Columbia, Canada.
16. Nagasaka, Akio, & Tanaka, Yuzuru. (1992). *Automatic video indexing and full-video search for object appearances,*. Paper presented at the Second Working Conference on Visual Database Systems II IFIP TC2/WG 2.6, Amsterdam, The Netherlands.
17. Otsuji, Kiyotaka, Tonomura, Yoshinobu, & Ohba, Yuji. (1991). *Video browsing using brightness data.* Paper presented at the Visual Communications and Image Processing '91, Kou-Hu Tzou, Toshio Koga.
18. Wang, Meng, & Zhang, Hong-Jiang. (2009, 21 October 2011). Video Content Structuring. *Scholarpedia.* 2012, from http://www.scholarpedia.org/article/Video_Content_Structuring
19. Yeo, Boon Lock, & Liu, Bede. (1995). Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology, 5*(6), 533-544
20. Yuan, Inhui, Wang, Huiyi, Xiao, Lan, Zheng, Wujie, Li, Jianmin, Lin, Fuzong, & Zhang, Bo. (2007). A Formal Study of Shot Boundary Detection. *IEEE Transactions on Circuits And Systems For Video Technology, 17*(2), 168-186.
21. Yuan, Jinhui, Li, Jianmin, Lin, Fuzong, & Zhang, Bo. (2005). *A unified shot boundary detection framework based on graph partition model.* Paper presented at the Proceedings of the 13th annual ACM international conference on Multimedia MULTIMEDIA '05, Singapore.
22. *Zabih, Ramin, Miller, Justin, & Mai, Kevin. (1995). A feature-based algorithm for detecting and classifying scene breaks,. Paper presented at the Proceedings of ACM Multimedia '95,, San Francisco.*
23. *Zhang, HongJiang, Kankanhalli, Atreyi, & Smoliar, Stephen W. (1993). Automatic partitioning of full-motion video. Multimedia Systems, 1(1), 10-28.*
24. *Zhang, Hongjiang, Low, Chien Yong, & Smoliar, Stephen W. (1995). Video parsing and browsing using compressed data. Multimedia Tools Appl, 1(1), 89-111.*

10/12/2013