International Journal of Computer Applications (0975 – 8887)
Volume 64– No.4, February 2013

Shot Detection using Pixel wise Difference with Adaptive Threshold and Color Histogram Method in Compressed and Uncompressed Video

Upesh Patel
Department of Electronics & Communication Engg, CHARUSAT University, Changa, Gujarat, India

Pratik Shah, PhD.
PDPU, Gandhinagar, Gujarat,India

Pradip Panchal
Department of Electronics & Communication Engg, CHARUSAT University, Changa, Gujarat, India

ABSTRACT

To elaborate a video in terms of its content, it needs to be partitioned into its smallest visual unit called shot. To segment a video into shots, shot boundary is needed. Here pixel wise difference is adapted for uncompressed and compressed video shot transition detection with adaptive threshold. Color histogram is also used for the cut and dissolve boundary. For cut boundary, traditional threshold technique works well but it fails in some cases like camera flash, fast zooming etc. To improve its performance 2nd derivative method is accepted for cut detection and for the dissolve boundary histogram difference with twin comparison method is used. Pixel intensity based approach is utilized for fade detection.

Keywords

Pixel wise difference, adaptive threshold color histogram, gradual boundary detection, hard cut detection, shot boundary detection, twin comparison**.**

1. INTRODUCTION

In today's advance world, the rapid advance of multimedia and web technologies, video data in various formats are available. To enable efficient browsing, searching and retrieval with these huge video data resources, the video database systems are needed. The traditional method is time consuming, lacks the speed because it uses human beings to manually explain the videos with text keywords. Therefore, more advanced methods are needed to support automatic indexing and retrieval directly based on videos content, which provide information related to video without consuming the time and with higher speed. Shot boundary detection is one of the key techniques for digital video analysis. Video shot boundary detection is usually the first and important step for content-based video retrieval, which helps to segment a video by detecting boundaries between camera shots. A digital video sequence consists of group of scene. A scene is a collection of one or more shots focusing on one or more objects of interest. In other words it is a set of images (frames) taken from a single camera. A shot boundary separates two consecutive shots when one shot changes to another shot. It means that if *n* number of shots taken from different cameras with different Video shot boundary detection has various applications in different domains like video indexing, video compression, video access and others. Many techniques have been developed [4-6] and compared [8] to detect frame transitions in video sequences. One of the simplest ways of detecting shot Transition is to compare the corresponding pixels between two consecutive frames. Another way is by using grayscale or color histograms of two frames. There are several other methods like edge changes and some predefined models, objects, regions to detect shot changes. Hybrid of these techniques has also been investigated [7].

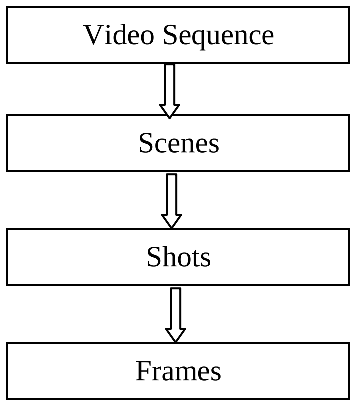

Fig.1 Illustration of video Sequence

In this paper hybrid approach is presented to achieve shot boundary. Color histogram difference [7] for abrupt cut boundary and color histogram difference with twin comparison method [8] for dissolve boundary detection are presented. The simplicity of the method relies on the low complexity of the computation of the color histogram difference. For fade (out/in) pixel intensity [11] based method is presented because fade contain at least single monochrome frame and standard deviation of a monochrome frame is near about zero which make easy to identify a monochrome image from others.

1.1 SHOT BOUNDARY DETECTION

1.1.1 Introduction

Shot boundaries can be broadly classified into two types: abrupt transition and gradual transitions [1]. Abrupt transition is instantaneous transition from one shot to the subsequent shot. Gradual transition occurs over multiple frames, which is generated via the application of more elaborated editing effects involving several frames, so that *Fk* frame belongs to one shot, frame *Fk+1* to the second, and the *k-1* frames in between represent a gradual transformation of *Fk* into *Fk+1*.

Gradual transition can be further classified into fade out/in (FOI) transition; dissolve transition, wipe transition, and others transition, according to the characteristics of the different editing effects. As given in [1], they can be defined as follows:

- Cut transition: This is instantaneous transition in which frame *Fk* belongs to one shot and *Fk+1* to the next shot, a clear discontinuity therefore existing.
- Fade transition: This is a shot transition with the first shot gradually disappearing (fade out) before the second shot gradually appears (fade in). During the FOI, two shots are spatially and temporally well separated by some monochrome frames.
- Dissolve transition: This is a shot transition with the first shot gradually disappearing while the second shot gradually appears. In this case, the last few frames of the disappearing shot temporally overlap with the first few frames of the appearing shot.
- Wipe transition: This is actually a set of shot change techniques, where the appearing and disappearing shots coexist in different spatial regions of the intermediate video frames. One scene gradually enters across the view while another gradually leaves.
- Other transition types: There is a multitude of inventive special effects techniques used in motion pictures. They are very rare and difficult to detect.

1.1.2 Algorithms

Two algorithms are developed as follows.

A. Pixel-wise Difference with Adaptive Thresholding

B. Color Histogram Difference

A. Pixel-wise Difference with Adaptive Thresholding

A.1.1. Algorithm

Pair-wise comparison evaluates the differences in intensity or color values of corresponding pixels in two successive frames. The simplest way is to calculate the absolute sum of pixel differences and compare it against a threshold [2]:

$$D(i) = \frac{1}{X * Y}\sum_{x=1}^{X}\sum_{y=1}^{Y}| f_{i+1}(x,y) - f_i(x,y) \quad \text{---(1.1)}$$

where X and Y are the frame width and height respectively, and $f_i(x,y)$ denotes the intensity value of the pixel at (x,y).

Calculate the mean μ and standard deviation σ of the feature vector distances as follows:

$$\mu = \frac{1}{N-1}\sum_{i=1}^{N-1} D(i,i+1) \quad \text{---(1.2)}$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1}[D(i,i+1)-\mu]^2} \quad \text{---(1.3)}$$

$$T_{cut} = \mu + k\sigma \quad \text{---(1.4)}$$

Where, k is pre-specified constant and N is number of frames in video sequence.

If $D(i) > T_{cut}$, declare the frame i as cut boundary.

1.2. Simulation Results

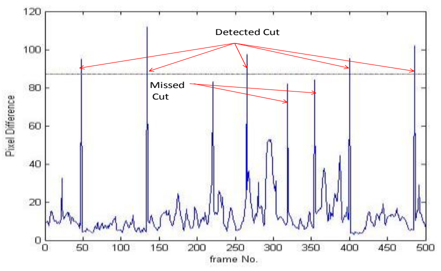

Fig.2 Pixel Difference for Cut detection

The algorithm is tested for detecting only abrupt changes. The simulation results for adaptive threshold are summarized in the following table:

Even with the adaptive thresholding, the algorithm produces false alarms, if the shot before/after the shot boundary includes high motion activity. The results indicate that the pixel-wise difference algorithm gives quite acceptable results with adaptive thresholding. Simulation results indicate that considering the difference between the difference signal values of adjacent frames is a worthwhile approach. The weakness of the pixel based features is the high sensitivity to the video content. The main disadvantage of this method is its inability to distinguish between a large change in a small area and a small change in a large area. We have observed that cuts are falsely detected when a small part of the frame undergoes a large, rapid change.

Table.1.1. Detected cuts using Pixel difference method for uncompressed video stream

Movie	Frame Range	Scaling Factor	Threshold	Desired	Correct	Missed	False Positive	R	P	F_1
Movie:1	1 to 90000	3	77.18	974	842	132	135	0.8644	0.8618	0.8631
Movie:2	1 to 90000	3	57.75	505	408	97	102	0.8079	0.8	0.8039
Movie:3	1 to 84000	3.5	85.6	718	591	127	94	0.8231	0.8627	0.8424

Table.1.2. Detected cuts using Pixel difference method for compressed video stream

Movie	Frame Range	Scaling Factor	Threshold	Desired	Correct	Missed	False Positive	R	P	F_1
Movie:1	1 to 90000	3.4	78.76	792	687	105	123	0.8674	0.8481	0.8576
Movie:2	1 to 90000	3.6	63.97	414	334	94	118	0.7803	0.7389	0.759
Movie:3	1 to 84000	3.8	81.77	548	443	89	115	0.8327	0.7939	0.8128

B. Color Histogram Difference

B.2.1. Overview

The presented method of shot boundary detection is based on the computation of difference of color histogram between frames as a measure of discontinuity. This difference can be the sum of the absolute difference between the bin values [3].

$$d_{RGB}(x,y)=\sum_{i=1}^{M}|h_x(i)-h_y(i)| \quad \text{---(2.1)}$$

Where h_x is the color histogram of image X which contains M different bins.
The shot boundary detection method is based on the difference between color histograms of frames belonging to a video sequence. This difference is computed as

$$HistDiff[i]=\sum_{j=1}^{M}|h_i(j)-h_{i-1}(j)| \quad \text{---(2.2)}$$

Where, h_i the color histogram with M bins of frame i corresponding to the video sequence.

B.2.2. Abrupt cut boundary detection

For abrupt cut detection color histogram is calculated. Then threshold is determined using equations (1.3-1.5), instead of pixel difference here color histogram difference is used. If the difference is greater than a threshold gives cut boundary. This scheme can achieve relatively good performance. But some time it can fail to detect correct boundary and can give false result. So here second order derivative method is used to determine correct cut boundaries.

Second Order Derivative Method

The traditional threshold method, in which feature variation between adjacent frames is directly compared with a global threshold T_c, usually works well.

However, due to illumination change or object/camera motion, feature variations within the same shot frequently exceed T_c. Therefore, it causes many false alarms. To overcome this drawback second order derivative method[4] is adopted and can be determine using (2.3).

$$HD2=d(x_{k+1},x_{k+2})-d(x_k,x_{k+1}) \quad \text{---(2.3)}$$

where *HD2* is 2nd derivative of the color histogram difference.

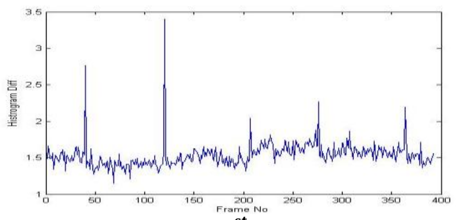

Figure.3. 1st order derivative

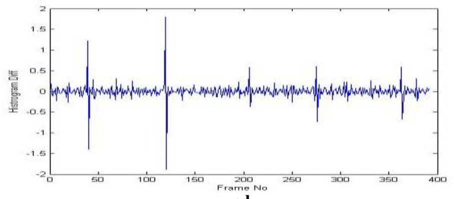

Figure.4. 2nd order derivative

Let f_k is the feature of k^{th} frame and $d(x_k,x_{k+1})$ denote the feature variation between the k^{th} and $(k+1)^{th}$ frame. In traditional method, feature variation $d(x_k,x_{k+1})$ is directly compared with T_c . In second order derivative, instead of $d(x_k,x_{k+1})$, $(d(x_{k+1},x_{k+2}) - d(x_k,x_{k+1}))$ is compared with T_c .

Table 2.1. Detected cuts using Color Histogram difference method for uncompressed video stream

Movie	Frame Range	Desired	Correct	Missed	False Positive	R	P	F_1
Movie:1	1 to 90000	974	938	36	52	0.963	0.9474	0.9551
Movie:2	1 to 90000	505	456	46	35	0.9029	0.9287	0.9156
Movie:3	1 to 84000	717	646	71	89	0.9009	0.8789	0.8898

Table 2.2. Detected cuts using Color Histogram difference method for compressed video stream

Movie	Frame Range	Desired	Correct	Missed	False Positive	R	P	F_1
Movie:1	1 to 90000	792	739	53	135	0.933	0.8455	0.8871
Movie:2	1 to 90000	414	348	66	67	0.8405	0.8385	0.8395
Movie:3	1 to 84000	548	472	76	46	0.8613	0.9111	0.8855

B.2.3. Simulation Results

Typical histogram difference patterns for CUTs are shown in Figure 4.

Despite the fact that during a gradual transition the frame to frame differences are usually higher than those within a shot, they are much smaller than the differences in the case of a cut and cannot be detected with the same threshold.

Table 2.1. and Table 2.2. summarizes the simulation results of the histogram difference method for both compressed and uncompressed video streams. We have also observed that the precision and recall for the uncompressed videos are better than compressed videos.

Simulation results indicate that histogram difference method obviously performs better than the pixel-wise method. Major reason for this is histogram method is not sensitive to local motion and local illumination changes. In the case of slight illumination changes or small camera/object motion, histogram difference method provides robust performance and better results compared to pixel-wise difference algorithm.

On the other hand, we have observed that global changes in the video frames, such as large brightness change, zooming or fading effects (especially fast zooming), result in false alarms. This is an expected result, since histogram feature is sensitive to the overall (or global) content of the video. Therefore, any effect resulting in a global change in the video content (e.g. fast zooming, large object movement) can be erroneously interpreted by the histogram algorithm.

Another conclusion from the simulation results is the algorithm cannot detect shot boundaries if there is a video-in-video effect. Since the outer frame, which remains constant, decreases the change ratio of the histogram bins significantly, amount of histogram difference is small. Consequently the transition is missed.

On the other hand, since the dark frames do not provide enough color information, histogram method cannot produce good results with dark video content as well.

C. Gradual boundary detection

Gradual transitions can be roughly divided into two classes: those that simultaneously but gradually affect every pixel of the image, and those that abruptly affect an evolving subset of the pixels, with the subset changing in each frame. The first class includes dissolve and fade-in/out effects. Dissolves show one image superimposed on the other as the frames of the first shot get dimmer and these of the second one get brighter. Fade out is slow decrease in brightness resulting in black frame; a fade in is gradual increase in intensity starting fromblack image.

One of the most popular methods for detecting shot transitions is comparing the histograms of consecutive frames. The assumption is that two frames which have a common background and unchanging objects will show little difference in their histograms. The histogram (either color or grayscale) is computed for each frame and the difference is calculated as shown below:

$$D(i,i+1)=\sum_{j=1}^{n}|H_i(j)-H_{i+1}(j)| \qquad \text{---(3.1)}$$

Where $H_i(j)$ is the j^{th} element of the histogram of the i^{th} frame, and n is the number of bins in histogram.

C.2.1 Fade Detection

The key problem of FOI detection [6] is the recognition of monochrome frame, since there is at least one monochrome frame within the FOI transition but monochrome frame seldom appears elsewhere. One dominant characteristic of monochrome frame is its low standard deviation of pixel intensities. FOI detection process is described in following flow chart.

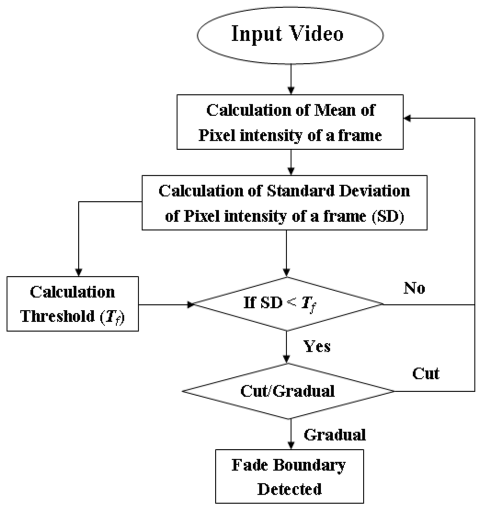
Input Video
Calculation of Mean of Pixel intensity of a frame
Calculation of Standard Deviation of Pixel intensity of a frame (SD)
Calculation Threshold (T_f)
If SD < T_f
No
Yes
Cut/Gradual
Cut
Gradual
Fade Boundary Detected

Figure 5. Flow chart for fade detection

Flow chart is as shown in Figure 5.Threshold is based on standard deviation of all the frames. It can be calculated using equation below:

$$T_f=\frac{\text{average of standard deviation}\times\text{scaling factor}}{100}$$

Scaling factor is variable according to the video. If SD of any frame finds below the threshold T_f, then that frame will be declare as monochrome frame and proceed for fade boundary detection. Fig.6. and Fig.7. shows the results of detected monochrome frames using this method.

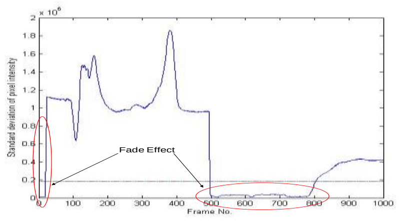

Figure 6. Fade transition detected using SD of pixel intensity for uncompressed video

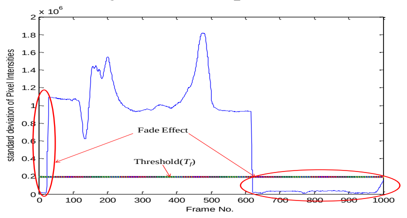

Figure 7. Fade transition detected using SD of pixel intensity for compressed video

C.2.3 Dissolve Detection

Twin-comparison method [5] except interframe differences uses cumulative differences between frames of gradual transition. In first pass cuts are detected using high T_{cut} . In the second pass potential starting frame Fs of the gradual transition is detected using the lower threshold T_l. F_s is then compared to subsequent frames (figure 1).This is called an accumulated comparison because during a gradual transition this difference value increases. The end frame of the transition is detected if two constraints are satisfied: difference between successive frames falls below T_l while the accumulated difference increases over Th. If consecutive difference falls below T_l before cumulative difference reaches T_h potential start frame *Fs* is dropped.

In some gradual transitions consecutive difference falls below T_l. Problem is solved allowing certain number of frames with low difference values before rejecting F_s.

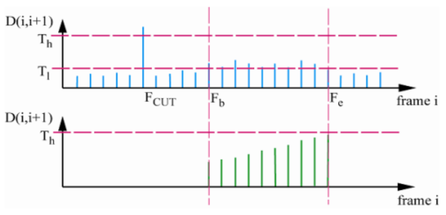

Figure.8. (a) Consecutive and (b) Accumulated histogram differences [5]

For calculating T_h histogram of the difference values of the clip is used. After identifying peak value, we assign T_h index value that corresponds to half of the peak value on right slope of the peak. T_h must be higher than mean value. Based on higher threshold lower threshold T_l will be decided.

C.2.3. Simulation Results

Table 3.1 and Table 3.2. summarizes the simulation results of our algorithms for detecting gradual boundaries for both compressed and uncompressed video streams. From the observation we can see that the precision and recall for the uncompressed videos are better than compressed videos.

In some cases during the fade detection, this method produces false alarm during night scenes. Because of its darkness our algorithm takes it as fade transition. We have observed that some of the dissolves exhibits very similar patterns to the ones that are produced by cuts. As a result the algorithm falsely identifies the dissolve center as cut. Long dissolves are the situations that such false detections mostly occur. In case of fast camera/object movement, it produces false alarm.

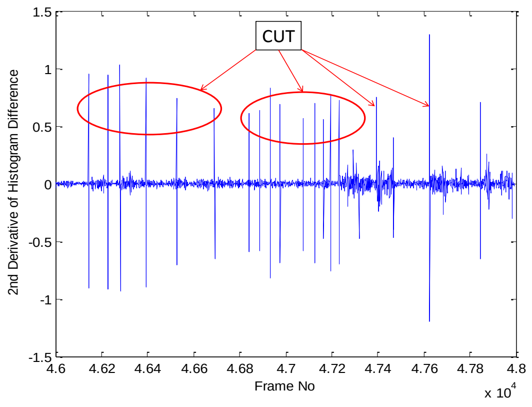

Fig.9. 2nd order color histogram difference of a frame sequence with cut boundaries for uncompressed video

Table 3.1 Detected Gradual Boundaries for Uncompressed video stream

Movie	Desired	Correct	Missed	False Positive	Recall	Precision	F1
Movie 1	26	21	5	4	0.807692	0.84	0.823529
Movie 2	29	23	6	7	0.793103	0.766667	0.779661
Movie 3	5	4	1	2	0.8	0.666667	0.727273

Table 3.2 Detected Gradual Boundaries for Compressed video stream

Movie	Desired	Correct	Missed	False Positive	Recall	Precision	F1
Movie 1	26	18	8	6	0.692308	0.75	0.72
Movie 2	29	17	12	9	0.586207	0.653846	0.618182
Movie 3	5	3	2	4	0.6	0.428571	0.5

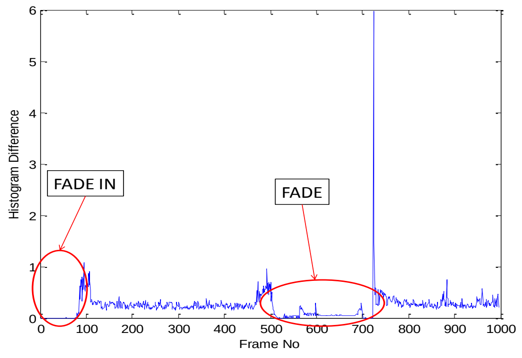

Fig.10. Color histogram difference of a frame sequence with Fade boundaries for uncompressed video

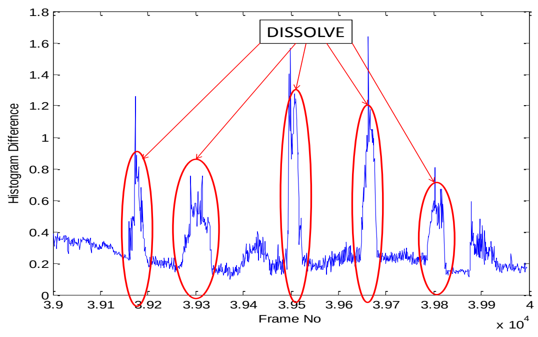

Fig.11. Color histogram difference of a frame sequence with Dissolve boundaries for uncompressed video

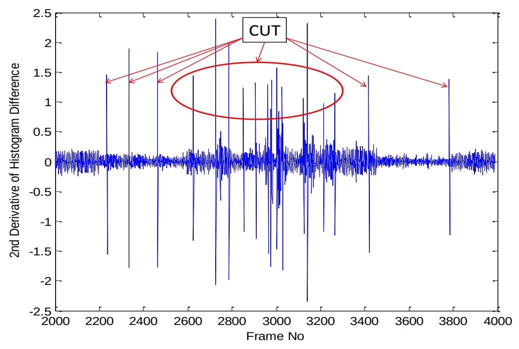

Fig.12. 2nd order color histogram difference of a frame sequence with cut boundaries for compressed video

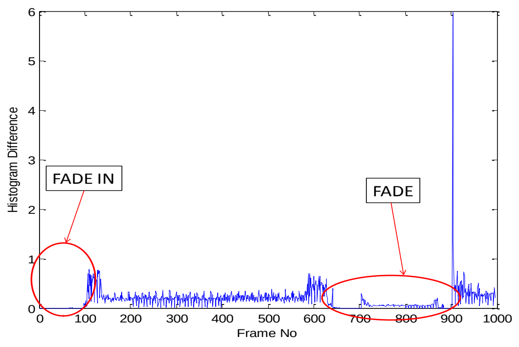

Fig.13. Color histogram difference of a frame sequence with Fade boundaries for compressed video

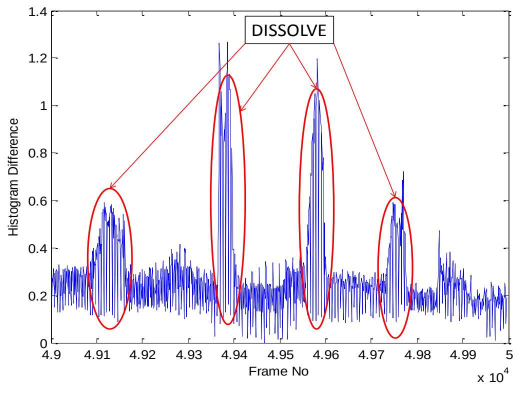

Fig.14. Color histogram difference of a frame sequence with Dissolve boundaries for compressed video

3. CONCLUSIONS

In the color histogram difference based approach for the cut boundary detection is simple and robust in front of camera and moving object for uncompressed videos. As compared to color histogram based approach pixel intensity based approach performed well for fades detection. Performance of twin comparison method for finding dissolve boundaries is also good for uncompressed videos. In future performance of this algorithm can be improved.

4. REFERENCES

[1] Rafael C.Gonzalez, Richard .E.Woods, "Digital Image Processing" Second Edition.

[2] Maheshkumar H. Kolekar and Somnath Sengupta, "video shot boundary detection: a comparative study of three popular approaches", *electronics and Electrical Communication Engineering Department, Indian Institute of Technology*, Kharagpur- 721 302, INDIA.

[3] Jordi Mas and Gabriel Fernandez, “Video shot boundary detection based on color histogram”, *Digital television center*, ramonLlull university Barcelona,Spain.

[4] J. Mas, and G. Fernandez, "Video Shot Boundary Detection based on Color Histogram," *Notebook Papers TRECVID2003*, 2003.

[5] Matko Saric, Hrvoje Dujmic and Domagoj Baricevic,” Shot Boundary Detection in Soccer Video using Twin-comparison Algorithm and Dominant Color Region ”, *FESB, University of Split*, Split.

[6] Jinhui Yuan, Wujie Zheng, Le Chen, Dayong ing, Dong Wang, Zijian Tong, Huiyi Wang, Jun Wu Jianmin Li, Fuzong Lin, Bo Zhang, “Tsinghua University at TRECVID 2004: Shot Boundary Detection and High-level Feature Extraction”, *Department of Computer Science and Technology Tsinghua University,Beijing* 100084, P. R. China.

[7] Jordi Mas and Gabriel Fernandez, *“Video shot boundary detection based on color histogram”*, Digital television center, ramonLlull university Barcelona,Spain, Available:http://www.nlpir.nist.gov/projects/tvpubs /tvpapers 03/ramonlull.paper.pdf.

[8] Matko Saric, Hrvoje Dujmic and Domagoj Baricevic,*”Shot Boundary Detection in Soccer Video using Twincomparison Algorithm and Dominant Color Region”*, FESB, University of Split, Split, Available:http://www.foi.hr/CMS_ home/znan_strucni_rad/konferencije/IIS/2007/papers/T0 9_05. pdf.

[9] Maheshkumar H. Kolekar and Somnath Sengupta*,”video shot boundary detection: a comparative study of three popular approaches”,* electronics and Electrical Communication Engineering Department, Indian Institute of Technology, Kharagpur721302,INDIA,Available:*http://www.ncc.org.in/do wnload.php?f=NCC2004/s17_05.pdf*.

[10] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, "Automatic partitioning of full-motion video", Multimedia Systems 1(1) (1993) 10-28.

[11] Jinhui Yuan, Wujie Zheng, Le Chen, Dayong Ding, Dong Wang, Zijian Tong, Huiyi Wang, Jun Wu Jianmin Li, Fuzong Lin, Bo Zhang*,” Tsinghua University at TRECVID 2004: Shot Boundary Detection and High-level Feature Extraction”,* State Key Laboratory of Intelligent Technology and System Department of Computer Science and Technology Tsinghua University Beijing 100084, P. R. China.