

Key Frame Extraction using Shot Boundary Detection Technique of Edge Contrast Approach from Uncompressed Video Stream

Upesh Patel

Charotar University of Science & Technology,
Email - patelupesh69@gmail.com

Abstract: Key frame extraction has been recognized as one of the important research issues in video information retrieval. The amount of data in video processing is significantly reduced by using video segmentation and key-frame extraction. This paper presents algorithm for shot boundary detection and key frames extraction. The algorithm differs from conventional methods in terms of image segmentation and attention model. Matching difference between two consecutive frames is computed with different weights. Shot boundaries are detected with automatic threshold. Key frame is extracted by using reference frame-based approach. Edge contrast technique is also useful for same purpose.

Key Words: automatic threshold, image segmentation, shot boundary detection, key frame extraction, Edge contrast.

1. INTRODUCTION:

Recent years have seen a rapid increase in the usage of multimedia information. In all the media types (text, image, graphic, audio and video), video is the most challenging one, as it combines all the other media information into a single data stream. Owing to the decreasing cost of storage devices, higher transmission rates, and improved compression techniques, digital video is becoming available at an ever increasing rate.

However, efficient access to video is not an easy task due to video's length and unstructured format. Video abstraction and summarization techniques are needed to solve this difficulty. Shot boundary detection and key frame extraction are two bases for abstraction and summarization techniques. Researchers have actively developed different approaches for intelligent video management, including shot Transition detection, key frame extraction, video retrieval, etc. Among these approaches, shot transition detection is the first step of content-based video analysis and key frame is a simple yet efficient form of video abstract. It can help users to understand the content at a glance and is of practical value.

Many approaches used different kinds of features to detect shot boundary which includes histogram, shape information, motion activity. Among these approaches, histogram is the popular approach. But, in these histogram-based approaches, pixels' space distribution was neglected. Different frames may have the same histogram, In view of this, Cheng *et al* where each frame divided into r blocks, and the difference of the corresponding blocks of consecutive frames was computed by color histogram.

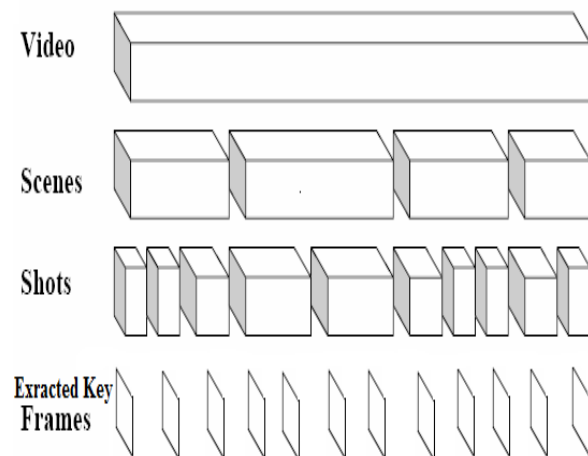


Fig.1. Overview of key frame Extraction

A shot is defined as the consecutive frames from the start to the end of recording in a camera. It shows a continuous action in an image sequence. There are two different types of transitions that can occur between shots, abrupt (discontinuous) also referred as cut, or gradual (continuous) such as fades, dissolves and wipes. The cut boundaries show an abrupt change in image intensity or color, while those of fades or dissolves show gradual changes between frames.

- A cut is an instantaneous transition from one scene to the next and it occurs between two frames.
- A fade is a gradual transition between a scene and a constant image (fade out) or between a constant image and a scene (fade in).
- A dissolve is a gradual transition from one scene to another, in which the first scene fades out and the second fades in.
- A wipe occurs as a line moves across the screen, with the new scene appearing behind the line.

2. SHOT BOUNDARY DETECTION:

2.1 Image Segmentation

First, each frame is divided into nine blocks, B(1,1), B(1,2), B(1,3), B(2,1), B(2,2), B(2,3), B(3,1), B(3,2), B(3,3). Then the difference of the corresponding blocks between two consecutive frames is computed. Finally, the final difference of two frames is obtained by adding up all the differences through different weights. Different position's pixels have different contribution to shot boundary detection: pixels on the edge are more important than others. Thus, different weights are given to blocks of different position. Here more weights are assigned to corner blocks compared to other blocks.

2.2 Matching Difference

There are six kinds of histogram match [08]. Color histogram was used in computing the matching difference in most literatures. However, through comparing several kinds of histogram matching methods, Nagasaka reached on conclusion that x^2 histogram outperformed others in shot boundary Recognition. Hence, x^2 histogram matching method is referred in this paper.

3. ALGORITHMS DESCRIPTION:

3.1 Shot boundary detection [3]

Let $F(k)$ be the k^{th} frame in video sequence, $k \in \{1, 2, \dots, F_v\}$ (F_v denotes the total number of frames in video).

$$D_B(k, k+1, i, j) = \sum_{l=0}^{L-1} \frac{[H(i, j, k) - H(i, j, k+1)]^2}{H(i, j, k)} \quad (1)$$

Where, $H(i, j, k)$ and $H(i, j, k+1)$ stand for the histogram of blocks at (i, j) in the k^{th} and $(k+1)^{\text{th}}$ frame respectively and

L is the number of gray in an image.

$$D(k, k+1) = \sum_{i=1}^m \sum_{j=1}^n w_{ij} D_B(k, k+1, i, j) \quad (2)$$

Where $m=n=3$, $w_{11}=2$, $w_{12}=1$, $w_{13}=2$, $w_{21}=1$, $w_{22}=1$, $w_{23}=1$, $w_{31}=2$, $w_{32}=1$, $w_{33}=2$.

$$MD = \sum_{k=1}^{F_v-1} \frac{D(k, k+1)}{F_v - 1} \quad (3)$$

$$STD = \sqrt{\frac{\sum_{k=1}^{F_v-1} (D(k, k+1) - MD)^2}{F_v - 1}} \quad (4)$$

$$\text{Threshold, } T = MD + a \times STD \quad (5)$$

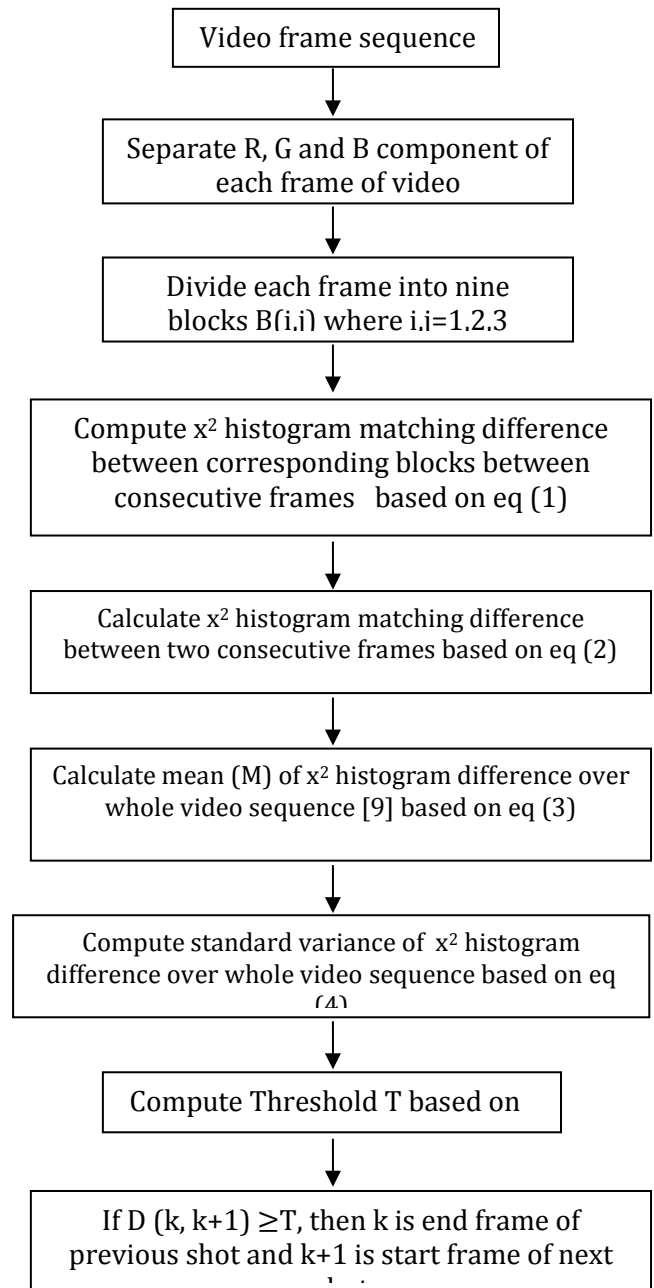


Fig.2. Flowchart of shot boundary detection algorithm.

3.2 Key frame extraction [3]

Step 1: First frame of each shot is reference frame and all other frames within shots are general frames. Computing the difference between all the general frames and reference frame in each shot with the above algorithm:

$$D_C(1, K) = \sum_{i=1}^m \sum_{j=1}^n w_{ij} D_{CB}(1, k, i, j) \quad (6)$$

Where $k=1, 2, 3, \dots, F_{CN}(k)$ (where $F_{CN}(k)$ is total number of frame in current shot).

Step 2: Searching for the maximum difference within a shot:

$$\text{Max}(i) = \{D_c(1, k)\}, k=2, 3 \dots F_{CN}(k) \quad (7)$$

Step 3: Determining shot type according to the relationship between Max (i) and MD: static shot (0), dynamic shot (1).

$$\text{Shot type} = 1 \text{ if } \text{Max}(i) \geq \text{MD} \\ = 0 \text{ otherwise} \quad (8)$$

Step 4: Determining the position of key frame: if Shot Type=0, with respect to the odd number of a shot's frames, the frame in the middle of shot is chose as key frame; in the case of the even number, any one frame between the two frames in the middle of shot can be chose as key frame. If Shot Type=1, the frame with the maximum difference is declared as key frame.

4. ADAPTIVE THRESHOLD FOR CUT DETECTION:

Choosing the right threshold is an important problem in the histogram comparison method. Global thresholds, i.e., constant thresholds are chosen in the above methods. A common problem of global thresholding is that in practice it is impossible to find a single global threshold that works with all kinds of video materials. Therefore, global threshold should be avoided. Instead, an adaptive threshold can be a better alternative to enhance the detection precision. It uses the local thresholds of the feature or similarity function to be compared, which in the above case are histogram similarity.

The adaptive thresholding algorithm can be achieved as following.

Let $S(f_n, f_{n+1})$ be the similarity function between two consecutive frames f_n and f_{n+1} . In addition, I use a sliding window of size $2w+1$ along the frame number, which covers the frames $f_{n-w}, \dots, f_n, \dots, f_{n+w}$.

This method is based on the fact that the value of the function $S(f_n, f_{n+1})$ is α times greater than the mean value of all $S(f_n, f_{n+1})$ values within the window excluding $S(f_n, f_{n+1})$ itself. It is achieved by the following steps.

- Find the maximum value of the similarity function $S(f_n, f_{n+1})$ within each window.
- To detect the hard cuts, the adaptive threshold within the window satisfies

$$S(f_n, f_{n+1}) + c \geq \frac{\sum_{i=n-w, i \neq n}^{n+w} (S(f_i, f_{i+1}) + c)}{2w}$$

Where α is the adaptive threshold, c is a constant.

When the nearby frames of f_n and f_{n+1} have zero values of $S(f_n, f_{n+1})$, it is difficult to find a proper threshold, which is the reason for adding the constant c . Generally, c is 0.8 in our case.

- Therefore, the adaptive threshold within the window is local constant, which can be expressed as

$$\alpha = \frac{S(f_n, f_{n+1}) + c}{\left(\frac{\sum_{i=n-w, i \neq n}^{n+w} (S(f_i, f_{i+1}) + c)}{2w} \right)}$$

In my calculation, the values are chosen as $w=1$, ALPHA=1

5. SIMULATION RESULTS:

We tested this algorithm for different type of videos that has cut or abrupt change between shots, smooth or gradual transition between shots, fade effect between shots and wipe effect between shots. For each video we extract key frames.

Case 1: Video A

In this case input video having cut or abrupt change between shots. Video A (250 frames of size 240×352), out of these frames we extracted 6 key frames successfully which preserves the content of video as shown below.

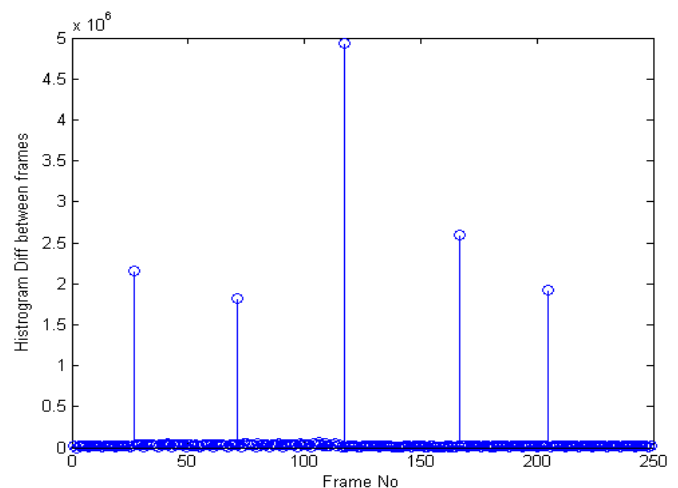




Fig. 3.Extracted key frames from video A (Courtesy: CHARUSAT)

Case 2: Video B

In this case input video having gradual transition between shots. Video B (90 frames of size 240×240), out of these 90 frames we extracted 14 key frames successfully which preserves the content of video as shown below.

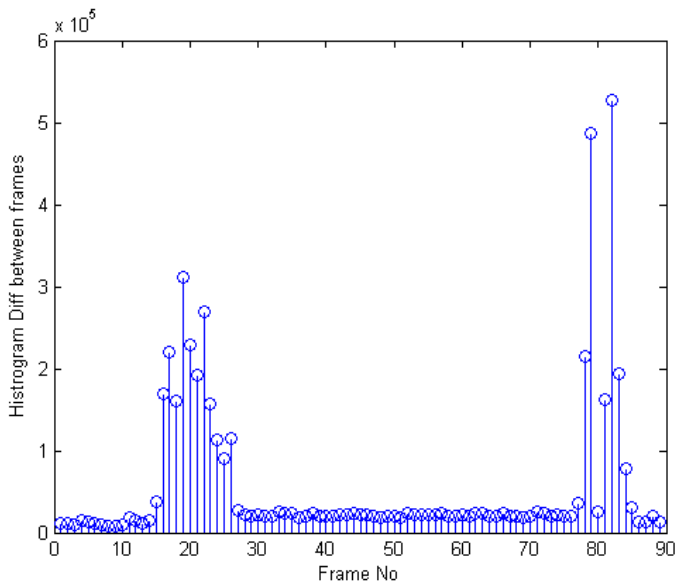


Fig.4. Extracted key frames from video B.

Case 3: Video C

In this case input video having Fade out effect

between shots. Video C (250 frames of size 240×352), out of these 250 frames we extracted 25 key frames successfully which preserves the content of video as shown below.

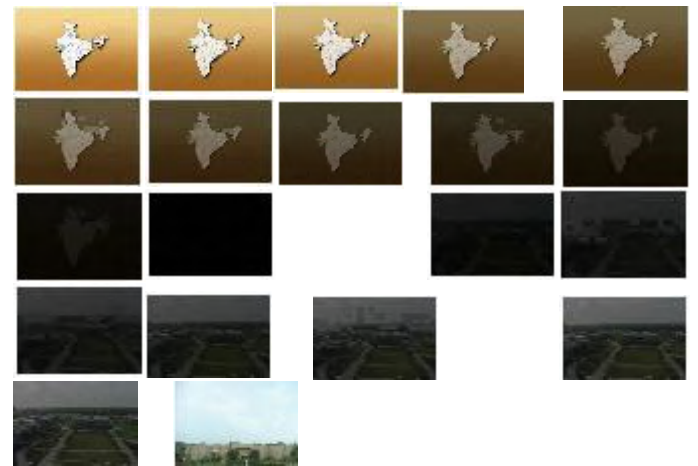
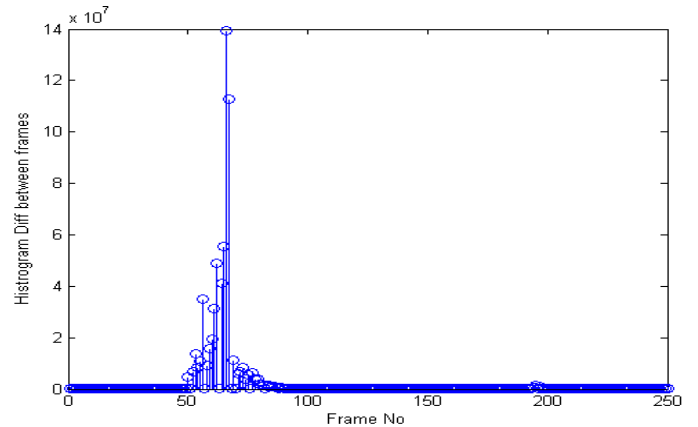
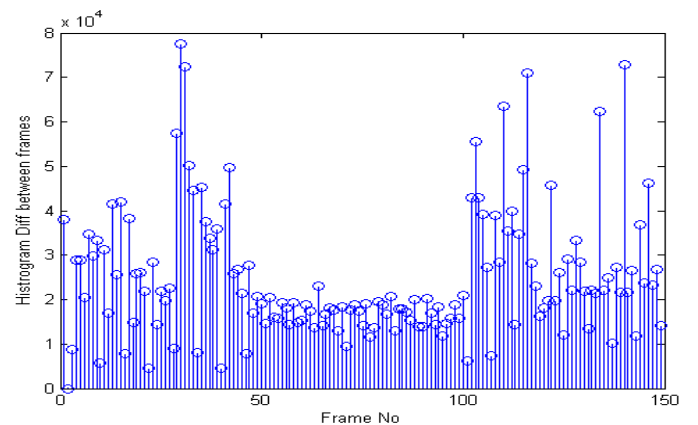


Fig. 5.Extracted key frames from video C

Case 3: Video D

In this case input video having wipe effect between shots. Video D (150 frames of size 240×352), out of these 150 frames we extracted 22 key frames successfully which preserves the content of video as shown below.



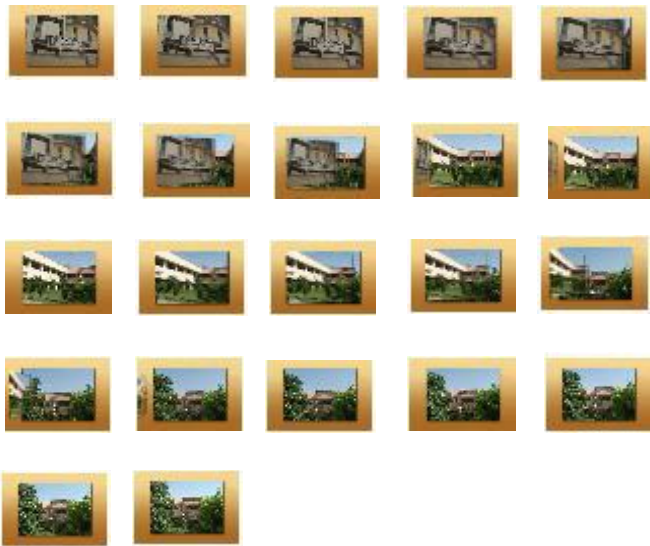


Fig. 6.Extracted key frames from video D

Following simulation results indicate both global thresholds as well as local adaptive threshold based on that cut boundaries are detected.

Case 1: In this case input video Charusat.avi was tested and cut boundaries were detected with respect to adaptive threshold.

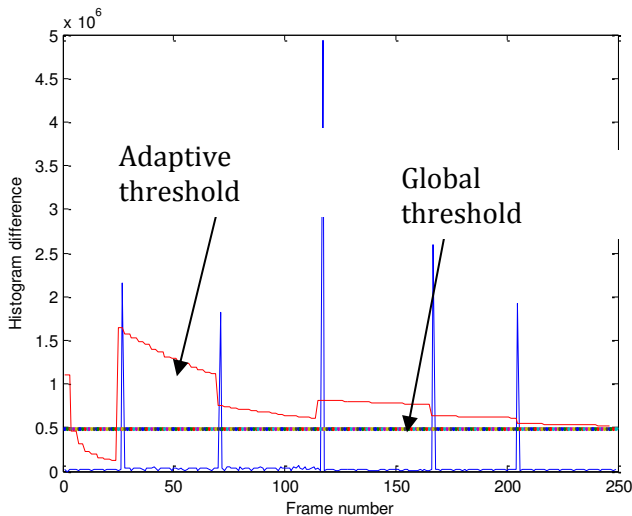


Fig.7. cut detection by Adaptive threshold from video charusat.avi

Case 2: In this case input video Black dawn.avi was tested and cut boundaries were detected with respect to adaptive threshold.

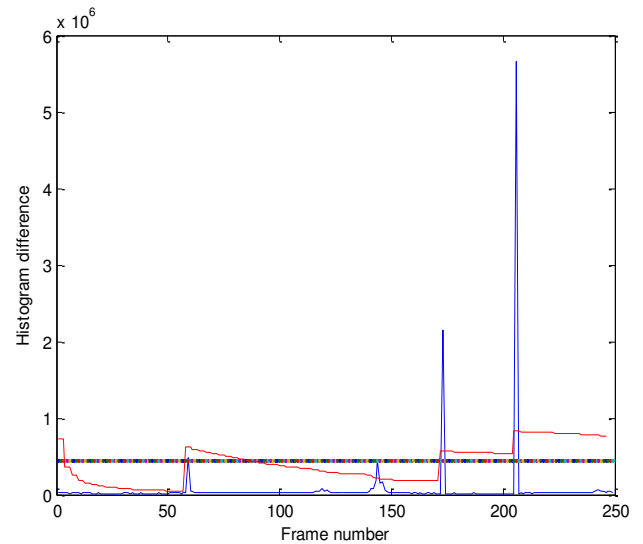


Fig 8. cut detection by Adaptive threshold from video Black dawn.avi

6. EDGE CONTRAST APPROACH FOR FADE, DISSOLVE DETECTION:

In previous topic Block based χ^2 histogram algorithm was explained. This algorithm detects Cut, Fade as well as Dissolve effect as shown in simulation results but in certain cases Fades and Dissolves are missed and not detected accurately. So in this chapter a new approach known as Edge based contrast (EC) is used for detection of only Fades and Dissolves.

During a fade in, object edges or contours gradually show up, while during a fade out object edges gradually disappear. During a dissolve, object edges gradually disappear and new object edges gradually show up. As a consequence, the perceived contrast decreases toward the center of a dissolve. Therefore I employ a so called edge-based contrast (EC) feature to detect the fades and dissolves in the video.

The basic idea of the edge-based contrast (EC) approach is to capture and emphasize the loss in contrast and/or sharpness, which is realized by applying stronger and weaker edges.

- Detect the edge map $Edge_n(x, y)$ of the frame f_n (I use the Canny edge detector^[105]).
- Define the strong and weak edges by setting a lower and higher threshold value as

$$S_n(x, y) = \begin{cases} Edge_n(x, y), & \text{If } Edge_n(x,y) \geq Ths \\ 0, & \text{otherwise} \end{cases}$$

And

$$W_n(x, y) = \begin{cases} Edge_n(x, y), & \text{If } Thw \leq Edge_n(x, y) \leq Ths \\ 0, & \text{otherwise} \end{cases}$$

Where Th_s and Th_w are the thresholds for the strong and weak edges.

- The strengths of strong and weak edge points are summed up by

$$SE_n = \sum_{x,y} S_n(x,y)$$

And

$$WE_n = \sum_{x,y} W_n(x,y)$$

The edge-based contrast (EC) is then defined as

$$EC_n = \frac{SE_n - WE_{n-1}}{SE_n + WE_{n+1}} + 1,$$

The Edge Contrast possesses the following features:

- The Edge Contrast is 0, if there are no strong edges in the frame.
- The EC lies between 0 and 1, if there are much more weak edges than strong edges in the frame
- The EC is about 1, if there are same amount of weak and strong edges in the Frame.
- The EC lies between 1 and 2, if there are much more strong edges than Weak edges in the frame.
- The EC is 2, if there are only strong edges in the frame.
- Following figure illustrates an example of how fades and dissolves influence the EC. They can easily be recognized by the peaks. The boundaries of a fade or a dissolve are detected by the abrupt end of the steep parts.

Following simulation result was obtained from video Charusat.avi consisting Fade and Dissolve effect.

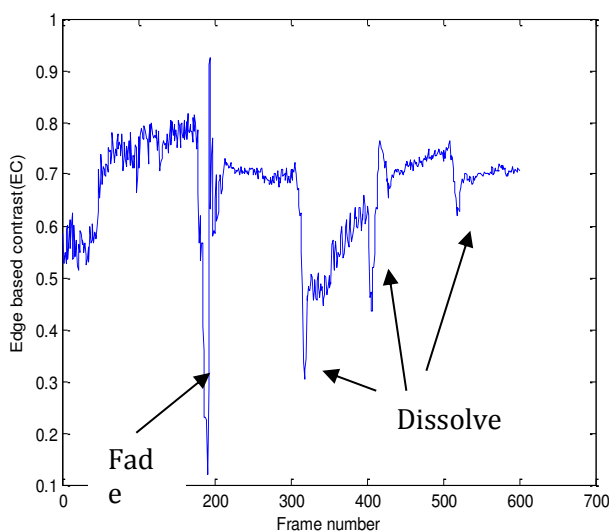


Fig 9. Fade and Dissolve detection based on Edge Contrast.

7. CONCLUSION:

We proposed algorithm of shot boundary detection and key frame extraction for different type of videos. We detect shot boundary with the help of χ^2 test histogram matching difference between consecutive frames with automatic threshold. After that key frames are extracted based on reference frame approach. We successfully detected shot boundaries and key frames. In different part of video having shot types like cut change between shots, Gradual transition with fade out or fade in effect between shots and wipe effect between shots. This work can implement for particular video also with all the effect with dissolve shot detection. The concept of shot boundary which separate redundant frames from video which indirectly saves memory and role of key frame which represents the salient content of video.

8. ACKNOWLEDGMENT:

We acknowledge the involvement of Dr. Trushit Upadhyaya, Head of Electronics & Communication Department of CHARUSAT, in the initial phase of the work reported here. We also extended our acknowledgement to Dr. A. D Patel, Principal, CSPIT, CHARUSAT for their motivation.

REFERENCES:

- P. Aigrain, H. Zhang, and D. Petkovic, "Content based representation and retrieval of visual media: A state-of-the-art review," *Multimedia Tools and Applications*, vol. 3, Nov 1996.
- Seung-Hoon Han, Kuk-Jin Yoon, and In So Kweon, 2000. "A new technique for shot detection and key frames selection in histogram space." *12th Workshop on Image Processing and Image Understanding*, pp 475-479.
- ZHAO Guang-sheng, "A Novel Approach for Shot Boundary Detection and Key Frames Extraction", 2008 *International Conference on Multimedia and Information Technology, IEEE-20083*
- Hanjalic, "Shot-boundary detection: Unraveled and resolved?", *IEEE Transaction on Circuits and System for Video Technology.*, Vol.12, No.2, February, 2002, pp. 90-105.
- Z. Cernekova, I. Pitas, and C Nikou, "Information Theory-based shot cut/fade detection and Video Summarization", *IEEE Transaction on Circuits and System for Video*

Technology, Vol.16, No.1, January 2006, pp. 82-91.

6. N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi, "Personalized abstraction of broadcasted American football video by highlight selection", *IEEE Transaction On Multimedia*, Vol.6, No.4, August 2004, pp. 575-586.
7. Hanjalic, "Multimodal approach to measuring excitement in video", *Proceedings of International Conference on Multimedia and Expo ICME 03[C]*, Vol.2, July 2003, pp. 289-292.
8. Y. Cheng, X. Yang, and D. Xu, "A method for shot boundary detection with automatic threshold", *TENCON'02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communication, Control and Power Engineering[C]*, Vol.1, October 2002: 582-585.
9. Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering", *Proceeding.ICIP'98[C]*, Chicago, IL, 1998, Vol.1, pp. 866-870.
10. D.J. He, N. Geng, "Digital image processing", Press of Xidia University. July, 2003, First Edition, pp. 104-106.