ISSN 2250-3153

Keyframe Based Video Summarization Using Automatic Threshold & Edge Matching Rate

Mr. Sandip T. Dhagdi, Dr. P.R. Deshmukh

Sipna COET, Amravati, India

Abstract- Shot boundary detection and Keyframe Extraction is a fundamental step for organization of large video data. Key frame extraction has been recognized as one of the important research issues in video information retrieval. Video shot boundary detection, which segments a video by detecting boundaries between camera shots, is usually the first and important step for content-based video retrieval and video summarization. This paper discusses the importance of key frame extraction; briefly review and evaluate the existing approaches, to overcome the shortcomings of the existing approaches.

This paper also proposes a new approach for key frame extraction based on the block based Histogram difference and edge matching rate. Firstly, the Histogram difference of every frame is calculated, and then the edges of the candidate key frames are extracted by Prewitt operator. At last, the edges of adjacent frames are matched. If the edge matching rate is above average edge matching rate, the current frame is deemed to the redundant key frame and should be discarded. Histogram-based algorithms are very applicable to SBD; They provide global information about the video content and are faster without any performance degradations.

Index Terms- key frame extraction; Histogram Difference; Prewitt operator; edge matching rate

I. INTRODUCTION

The availability of cheap digital storage has led to a rapid expansion of digital video archives, and with such enormous video data resources, sophisticated video database systems are highly demanded to enable efficient browsing, searching and retrieval. Initial attempts to meet this need involved the use of textual data obtained from subtitles and speech recognition transcripts. However, while these methods are effective for some needs, such as broadcast news retrieval, they take no account of the visual content of the video.

The traditional video indexing method, which uses human beings to manually annotate or tag videos with text keywords, is time-consuming, lacks the speed of automation and is hindered by too much human subjectivity. Therefore, more advanced approaches such as content-based video retrieval are needed to support automatic indexing and retrieval directly based on videos content, which provide efficient search with satisfactory responses to the scenes and objects that the user seeks.

Video shot boundary detection, which segments a video by detecting boundaries between camera shots, is usually the first and important step for content-based video retrieval [1]. A video consists of a sequence of images (often being called frames), which can be played consecutively at the speed of around 20 to 30 frames per second in order to view smooth motion. To index and retrieval a video, shot boundary detection is usually conducted to segments the video into shots by detecting boundaries between camera shots.

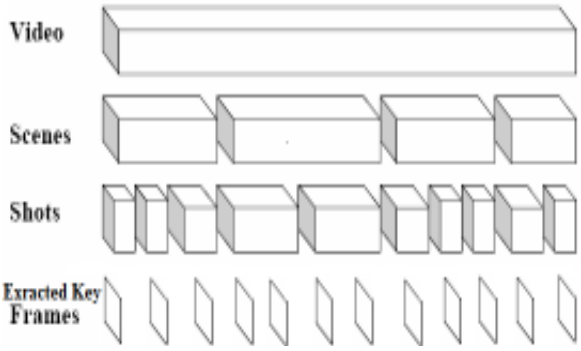

Figure.1.1: Overview of Shot boundary detection.

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 [1]. The cut boundaries show an abrupt change in image intensity or colour, 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 over two frames.
- 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.

1.1 Motivation

Recent developments in video compression technology, the widespread use of digital cameras, high capacity digital systems, coupled with the significant increase in computer performance and the growth of Internet and broadband communication, have increased the usage and availability of digital video. Applications such as multimedia information systems, distance learning, video on-demand produce and use huge amount of video data. This

situation created a need for tools that can effectively categorize, search and retrieve the relevant video material.

In general, management of such activities over large collections of video requires knowledge of the "content" of the video. In particular, digital video data can be processed with the objective of extracting the information about the content conveyed with this data. The algorithms developed for this purpose, referred as "video content analysis" algorithms and serve as the basis for developing tools that would enable us to understand the events and objects within the scene of a video, or generate summary of large video material or even to derive semantically meaningful information from the video [2].

The definition of "content" is highly application dependent but there are a number of commonalities in the applications of content analysis. Among others, *shot boundary detection* (SBD), also known as *temporal video segmentation* is one of the important aspects.

Parsing a video into its basic temporal units -shots- is considered as the initial step in the process of video content analysis. A shot is a series of video frames taken by a single camera, such as, for instance, by zooming into a person or an object, or simply by panning along a landscape [2]. The content is similar in shot regions. The regions where the significant content change occurs are, therefore, called shot boundaries. Since the SBD is a prerequisite step for most of the video applications involving the understanding, parsing, indexing, characterization, or categorization of video, temporal video segmentation has been an active topic of research in the area of content based video analysis.

1.2 THRESHOLDING

Shot boundaries are identified based on the visual content change. Therefore, the most critical activity in the SBD process is the selection of the thresholds in any shot boundary detection step. The performance of the algorithm mainly remains in the thresholding phase. However, using a single threshold cannot perform equally well for all video sequences. Using a dynamic global threshold by extracting the overall sequence characteristic cannot solve this problem. Dynamic local thresholds are considered as a better alternative but thresholding still remains as a major problem in this area.

II. FUNDAMENTAL PROBLEMS OF SBD

Shot boundary detection (SBD) is not a new problem anymore. It has been studied more than a decade and resulting algorithms have reached some maturity. However, challenges still exist and are summarized in the upcoming sections:

2.1 DETECTION OF GRADUAL TRANSITION

During the video production process, first step is capturing the shots by using a single camera. Two consecutive shots are then attached together by a shot boundary that can either be abrupt or gradual. Abrupt shot boundaries are created by simply attaching a shot to another. While there is no modification in the consequent shots in an abrupt shot boundary, gradual transitions result from editing effects applied to the shots during attachment operation. According to the editing effect gradual transitions can be further divided into different subtypes. The number of possible transitions due to editing effect is quite high but most of the transitions fall into the three main categories: dissolve, fades (fade in, fade out), and wipes. Different types of transitions are demonstrated in the following figures:

Figure 2.1: Dissolve Effect I

Figure 2.2: Dissolve Effect II

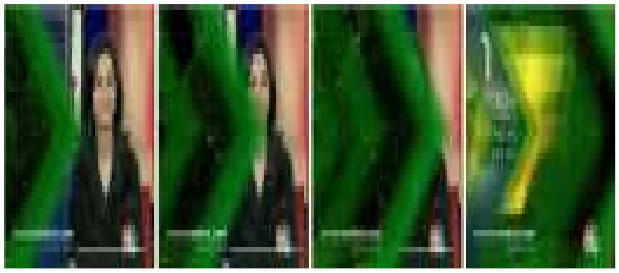

Figure 2.3: Wipe Effect

Figure 2.4: Fade In/Fade Out Effect

Detection of abrupt changes has been studied for a long time. On the other hand, gradual transitions pose a much more difficult problem. This situation is mainly due to the amount of available video editing effects. The problem gets harder when multiple effects are composed in the case of a lot of object or camera motion. Another reason is that the gradual transitions spread over time. Each editing effect has a different temporal

pattern than the others and the temporal duration changes from three frames to hundred frames. Finally, the temporal patterns, as a result of editing effects to create a gradual transition, are very similar to the patterns due to camera/object motion. Therefore, gradual transitions remain to be one of the most challenging problems in SBD.

2.2 FLASHLIGHTS

Color is the primary element of video content. Most of the video content representations employ color as a feature. Continuity signals based on color feature exhibit significant changes under abrupt illumination changes, such as flashlights. Such a significant change might be identified as a content change (i.e. a shot boundary) by most of the shot boundary detection tools. Several algorithms propose using illumination invariant features, but these algorithms always face with a tradeoff between using an illumination invariant feature and loosing the most significant feature in characterizing the variation of the visual content. Therefore, flashlight detection is one of the major challenges in SBD algorithms.

2.3 OBJECT/CAMERA MOTION

Visual content of the video changes significantly with the extreme object/camera motion and screenplay effects (e.g. one turns on the light in a dark room) very similar to the typical shot changes. Sometimes, slow motion cause content change similar to gradual transitions, whereas extremely fast camera/object movements cause content change similar to cuts. Therefore, it is difficult to differentiate shot changes from the object/camera motion based on the visual content change. Therefore, the most critical activity in the SBD process is the selection of the thresholds in any shot boundary detection step. The performance of the algorithm mainly remains in the thresholding phase. However, using a single threshold cannot perform equally well for all video sequences. Using a dynamic global threshold by extracting the overall sequence characteristic cannot solve this problem. Dynamic local thresholds are considered as a better alternative but thresholding still remains as a major problem in this area.

III. LITERATURE REVIEW & RELATED WORK

SBD is a popular area in the video content analysis and has been studied for a long time. Research has resulted in a variety of algorithms. In this section, briefly the SBD work in the literature is reviewed.

Many approaches used different kinds of features to detect shot boundary, including histogram, shape information, motion activity. Among these approaches, histogram is the popular approach. However, in these histogram-based approaches, pixels' space distribution was neglected. Different frames may have the same histogram.

In view of this, Cheng *et al* [13] divided each frame into *r* blocks, and the difference of the corresponding blocks of consecutive frames was computed by color histogram; the difference $D(i,i+1)$ of the two frames was obtained by adding up all the blocks' difference; in the meanwhile, the difference $V(i,i+1)$ between two frames i and $i+1$ was measured again without using blocks. Based on $D(i,i+1)$ and $V(i,i+1)$, shot boundary was determined[1].

Zhuang *et al.* [14] proposed an unsupervised clustering method. A video sequence is segmented into video shots by clustering based on color histogram features in the HSV color space. For each video shot, the frame closest to the cluster centroid is chosen as the key frame for the video shot. Notice that only one frame per shot is selected into the video summary, regardless of the duration or activity of the video shot.

Zuzana Cernekova [15] proposed a new approach for shot boundary detection in the uncompressed image domain based on the MI and the joint entropy (JE) between consecutive video frames. Mutual information is a measure of information transported from one frame to another one. It is used for detecting abrupt cuts, where the image intensity or color is abruptly changed. A large video content difference between two frames, showing weak inter-frame dependency leads to a low MI. The entropy measure provides with better results, because it exploits the inter-frame information flow in a more compact way than a frame subtraction.

Ali Amiri [16] proposed a novel video summarization algorithm which is based on QR-decomposition. Some efficient measures to compute the dynamicity of video shots using QR-decomposition was utilize in detecting the number of key frames selected for each shot. Also, a corollary that illustrates a new property of QR-decomposition. This property was utilized in order to summarize video shots with low redundancy.

Hanjalic *et al.* [17] developed a similar approach by dividing the sequence into a number of clusters, and finding the optimal clustering by cluster-validity analysis. Each cluster is then represented in the video summary by a key frame. The main idea in this paper is to remove the visual redundancy among frames.

DeMenthon *et al.* [18] proposed an interesting alternative based on curve simplification. A video sequence is viewed as a curve in a high dimensional space, and a video summary is represented by the set of control points on that curve that meets certain constraints and best represent the curve.

Doulamis *et al.* [19] also developed a two step approach according to which the sequence is first segmented into shots, or scenes, and within each shot, frames are selected to minimize the cross correlation among frames' features.

3.1 STATE OF ART SHOT BOUNDARY DETECTION

SBD algorithms from the literature are studied and analyzed. A brief summary of the algorithms is presented in the remainder of this chapter. Following are the algorithms.

3.1.1 Pixel-wise Difference with Adaptive Thresholding

In this algorithm individual pixels from frames are compared to find out rame dierence. Pair-wise comparison evaluates the differences in intensity or color values of corresponding pixels in two successive frames. In this algorithm the pixel-wise difference algorithm gives quite acceptable results with adaptive thresholding. By considering difference between the difference signal values of adjacent frames is a worthwhile approach. In practice, it is observed that it is useful to reduce the effects of scenes containing a lot of movement by comparing the

difference signal with a threshold derived from the maximum and minimum difference signals over a small aperture.

Even with the adaptive thresholding, the algorithm produces false alarms, if the shot before/after the shot boundary includes high motion activity. The reason can be explained as follows: The weakness of the pixel based features is the high sensitivity to the video content. It is difficult for this algorithm to understand whether the change in the continuity signal is due to shot boundary or due to disturbances/motion. In order to enhance the algorithm, adaptive thresholding can be used. However, the high level of activity in the images around shot boundary produces a larger difference signal than expected. As a result adaptively obtained threshold is larger. A threshold that is larger than expected results in missed shot boundary.

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. It is observed that cuts are falsely detected when a small part of the frame undergoes a large, rapid change. For the same reason, the algorithm is not able to detect most of the flashlights.

3.1.2 Histogram Difference with Adaptive Thresholding

While the pixel-wise approach focuses on local intensity (color) comparison between individual pixels, this method is interested with the global percentage of colors that an image contains. The method works by calculating percentages from the bin totals and comparing them with those of the adjacent frame giving a difference value. A difference above the threshold value will be classed as a shot change [2].

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, It is 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 observation of the algorithm is that it 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.

3.1.3 Edge Change Ratio

During a cut or a dissolve, new edges appear far from the locations of old edges. In addition, old edges disappear far from the location of new edges. This observation was applied to digital video segmentation by Zabih et. al. [4] who identified two new types of edge pixels:

- Entering pixel: One that appears far from an existing edge pixel.
- Exiting pixel: One that disappears far from an existing edge pixel.

According to Zabih et. al. it is possible to detect CUTs and GTs by counting the entering and exiting pixels.

The main drawback for this algorithm is its execution time. Calculating the edges and the dilation operation consumes significant computing power and in parallel takes more time to process the frames. Cut detection results can be considered acceptable, if the algorithm could be faster. However, gradual transition results are not promising.

3.1.4 Petersohn's Algorithm with 2-Means Clustering

This system which uses pixel, edge and histogram difference statistics for detecting CUTs and GTs. The system uses the luminance information only and down-samples all the frames by a factor of 8 in *x* and *y* directions before processing.

This algorithm is quite fast. Main reason behind this improvement in the speed of the shot boundary detection is that the system uses the luminance information only and down samples all the frames by a factor of 8 in *x* and *y* directions.
Algorithm senses only significant changes in the video content. In addition to using DC images, the algorithm employs pixel and histogram features together.

3.1.5 Segmentation Method

Although many methods have been proposed for this task, finding a general and robust shot boundary method that is able to handle the various transition types caused by photo flashes, rapid camera movement and object movement is still challenging. In this method the problem of shot boundary detection is casted into the problem of text segmentation in natural language processing. This is possible by assuming that each frame is a word and then the shot boundaries are treated as text segment boundaries (e.g. topics).

The text segmentation based approaches in natural language processing can be used. The shot boundary detection process for a given video is carried out through two main stages. In the first stage, frames are extracted and labeled with predefined labels. In the second stage, the shot boundaries are identified by grouping the labeled frames into segments. Following six labels to label frames in a video can be used: NORM FRM (frame of a normal shot), PRE CUT (pre-frame of a CUT transition), POST CUT (post-frame of a CUT transition), PRE GRAD (pre-frame of a GRADUAL transition), IN GRAD (frame inside a GRADUAL transition), and POST GRAD (post-frame of a GRADUAL transition).

Given a sequence of labeled frames, the shot boundaries and transition types are identified by looking up and processing the frames marked with a non NORM FRM label. It uses a support vector machine (SVM) for the purpose of shot boundary detection.

3.1.6 Motion-Based Algorithm

Similar to Petersohn's Algorithm frames are first down-sampled by a factor of 2 in both *x*- and *y*-directions. The algorithm further performs a preprocessing step for filtering out the obvious non-boundary frames. With this achievement, it is allowed to spend more processing power on the areas that are likely to be shot boundaries.

In this early processing step, sum of absolute differences between the R, G and B values are calculated for candidate shot

boundary frames. Generally, motion-based algorithms are not preferred in the uncompressed domain, since estimation of motion vectors consumes significant computational power and time. In contrary, Algorithm is faster mainly due to the preprocessing step for skipping the frames which have very low probability of being a shot boundary. Secondly, utilizing down-sampled images together with the fastest block matching algorithm (i.e. Dual Cross Search) increases the speed of the algorithm significantly.

3.1.7 Motion Activity Descriptor Based Algorithm

The motion activity is one of the motion features included in the visual part of the MPEG-7 standard. It also used to describe the level or intensity of activity, action, or motion in that video sequence. The main idea underlying the methods of segmentation schemes is that images in the vicinity of a transition are highly dissimilar. It then seeks to identify discontinuities in the video stream. The general principle is to extract a comment on each image, and then define a distance [5] (or similarity measure) between observations.

The application of the distance between two successive images, the entire video stream, reduces a one-dimensional signal, in which we seek then the peaks (resp. hollow if similarity measure), which correspond to moments of high dissimilarity. In this work [19], the extraction of key frames method based on detecting a significant change in the activity of motion is used. To jump 2 images which do not distort the calculations but we can minimize the execution time. First the motion vectors between image i and image i+2 is extracted then calculates the intensity of motion, we repeat this process until reaching the last frame of the video and comparing the difference between the intensities of successive motion to a specified threshold. The idea can be visualized in figure below.

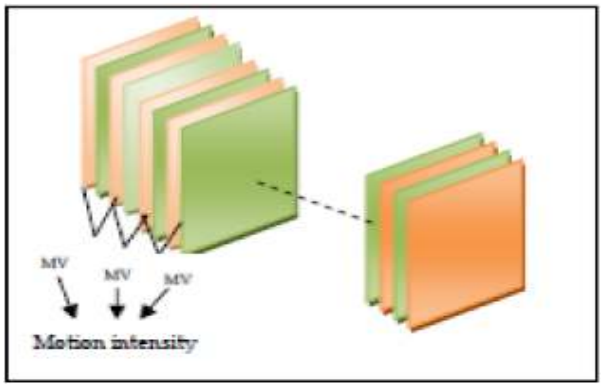

Figure 3.1: the Idea of Video Segmentation Using Motion Intensity

IV. IMPLEMENTATION STRATEGY

The method for key frame extraction consists of three steps: Input a video and calculate the block based histogram difference of each consecutive frame. Choose the current frame as a candidate key frame whose histogram difference is above the threshold point. Extract the edges of the candidate key frames and calculate the edge matching rate of adjacent frames. If the edge matching rate is above average edge matching rate, the current frame is considered as a redundant frame and should be eliminated from the candidate key frames.

Project will consist of following three modules.

- Shot boundary detection
- Key frame extraction
- Eliminate Redundant Frames

Following is the explanation of each module with their algorithm.

4.1 SHOT BOUNDARY DETECTION:

Let F(k) be the k^{th} frame in video sequence, k = 1,2,…., Fv (Fv denotes the total number of video). The algorithm of shot boundary detection is described as follows.

Algorithm : Shot boundary detection

Step 1: Partitioning a frame into blocks with m rows and n columns, and B(i, j, k) stands for the block at (i, j) in the k^{th} frame;

Step 2: Computing x^2 histogram [8] matching difference between the corresponding blocks between consecutive frames in video sequence. 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) th frame respectively. Block's difference is measured by the following equation:

$$D_B(k,k+1,i,j)=\sum_{l=0}^{L-1}\frac{[H(i,j,k)-H(i,j,k+1)]^2}{H(i,j,k)} \quad (4.1)$$

Where L is the number of gray in an image;

Step 3: Computing x^2 histogram difference between two consecutive frames:

$$D(k,k+1)=\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} D_B(k,K+1,i,j) \quad (4.2)$$

where wij stands for the weight of block at (i, j) ;

Step 4: Computing threshold automatically:
Computing the mean and standard variance of x^2 histogram difference over the whole video sequence [2]. Mean and standard variance are defined as follows :

$$MD = \frac{\sum_{k=1}^{Fv-1} D(k,k+1)}{Fv-1} \quad (4.3)$$

$$STD = \sqrt{\frac{\sum_{k=1}^{Fv-1}(D(k,k+1)-MD)^2}{Fv-1}} \quad (4.4)$$

Step 5: Shot boundary detection

Let threshold $T = MD + a \times STD$. Shot candidate detection: if $D(i, i+1) \geq T$, the i^{th} frame is the end frame of previous shot, and the $(i+1)^{th}$ frame is the end frame of next shot.

Final shot detection: shots may be very long but not much short, because those shots with only several frames cannot be captured by people and they cannot convey a whole message. Usually, a shortest shot should last for 1 to 2.5 s. For the reason

of fluency, frame rate is at least 25 fps, (it is 30 fps in most cases), or flash will appear. So, a shot contains at least a minimum number of 30 to 45 frames. In our experiment, video sequences are down sampled at 10 fps to improve simulation speed. On this condition, the shortest shot should contain 10 to 15 frames. 13 is selected for our experiment. We formulate a "shots merging principle": if a detected shot contain fewer frames than 13 frames, it will be merged into previous shot, or it will be thought as an independent one.

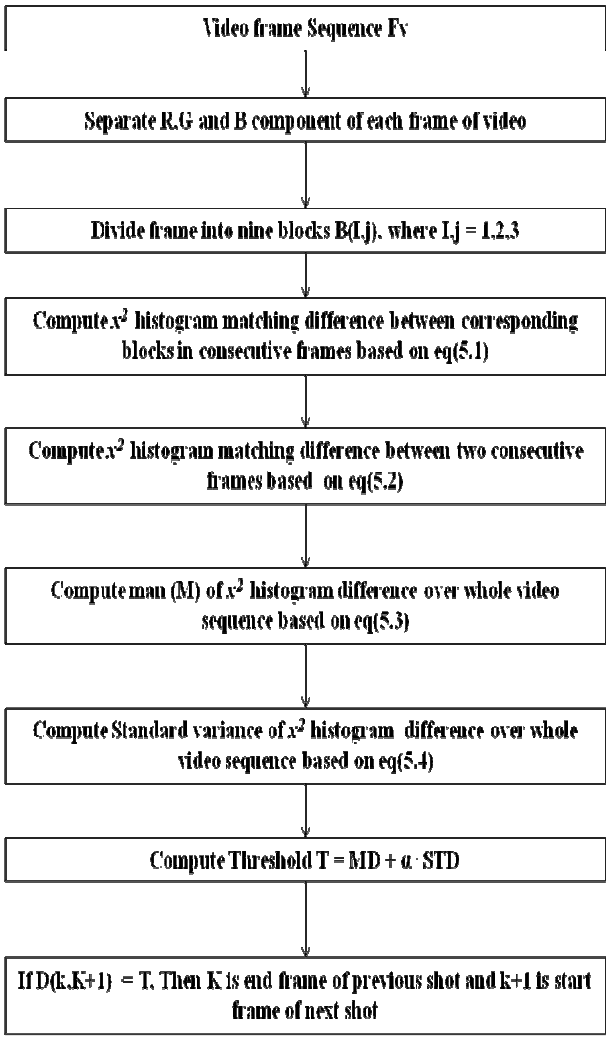

Figure 4.1 Flowchart of shot boundary detection algorithm.

Definition 1: Reference Frame: it is the first frame of each shot; General Frames: all the frames except for reference frame; "Shot Dynamic Factor" max(i): the maximum x^2 histogram within shot i;

Definition 2: Dynamic Shot and Static Shot: a shot will be declared as dynamic shot, if its max(i) is bigger than MD; otherwise it is static shot; $F_C(K)$: the k^{th} frame within the current shot, k=1,2,3… F_{CN} (K) (F_{CN} (k) is the total number of the current shot).

4.2 KEYFRAME EXTRACTION

The algorithm of key frame extraction is described as follows.

Algorithm: Key frame extraction

Step 1: Computing the difference between all the general frames and reference frame with the above algorithm:

$$D_C(1,k)=\sum_{i=1}^{m}\sum_{j=1}^{n}w_{ij}D_{CB}(1,k,i,j),k=2,3,4,.....F_{CN} \quad (4.5)$$

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

$$\max(i)=\{D_C(1,k)\}_{\max},k=2,3,4,...F_{CN} \quad (4.6)$$

Step 3: Determining "ShotType" according to the relationship between max(i) and MD: StaticShot(0) or DynamicShot:

$$ShotType_C = \begin{cases} 1 \; if \;\; \max(i) \geq MD \\ 0 \; Others \end{cases} \quad (4.7)$$

Step 4: Determining the position of key frame: if $ShotType_C$ = 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 choose as key frame. If $ShotType_C$ = 1, the frame with the maximum difference is declared as key frame.

4.3 ELIMINATE REDUNDANT FRAMES

4.3.1 Extract Edges of the Candidate Key Frames

The candidate key frames obtained from the above treatment do well in reflecting the main content of the given video, but exist a small amount of redundancy, which need further processing to eliminate redundancy.

As the candidate key frames are mainly based on the Histogram difference which depends on the distribution of the pixel gray value in the image space, there may cause redundancy in the event that two images whose content are the same exist great difference from the distribution of the pixel gray value. For example, the substance content of images A and B in Fig.5.2. don't change, but the two images are both identified as key frames as a result of the different gray value distribution, resulting in redundancy.

Figure 4.2: Redundant Frames A and B

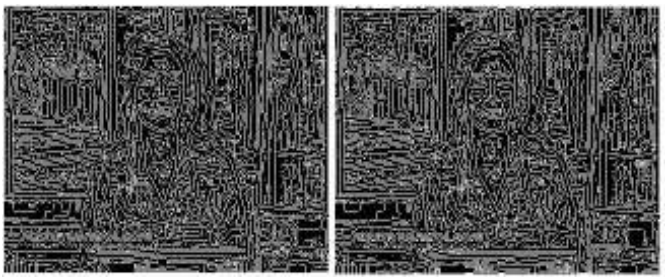

Figure 4.3: Edge Images of A and B.

As edge detection can remove the irrelevant information and retain important structural properties of the image, we can extract the edges of objects in the image to eliminate redundancy. At present, there are many edge detection algorithms, which are mainly based on the differentiation and combined with the template to extract edges of images.

Edge detection operators that are commonly used are: Roberts [20] operator, Sobel operator, Prewitt operator and the Laplace operator etc. Here we extract edges of frames by Prewitt operator.

4.3.2Eliminate Redundant Frames Based on the Edge Matching Rate

The edge images have no difference in the distribution of the gray value. For example, the images shown in Fig.4.3 are the edge images of the images shown in Fig.4.2. with Prewitt operator and both of them are remarkably similar. So we use the edge matching rate to match the edges of adjacent frames to eliminate redundant frames. The formula for calculating the edge matching rate is as follows:

$$p(f_i, f_{i+1}) = s / n \tag{4.8}$$

In the formula,

$$n = \max(n_{fi}, n_{fi+1}) \tag{4.9}$$

$$s = \sum_{i}^{m} \sum_{j}^{n} h(i, j) \tag{4.10}$$

$$h(i,j) = \begin{cases} 1, v_{fk}(i,j) = v_{fk+1}(i,j) \\ 0, Otherwise \end{cases} \tag{4.11}$$

Where $v_{fk}(i,j)$ and $v_{fk+1}(i,j)$ are the pixel values of the position (i, j) in the frame *fk* and the frame *fk+1*, respectively. *m* and *n* indicate the height and the width of the image, *nfi* and *nfi+1* represent the number of the pixels on the edge of the frame *fi* and the frame *f* i+1 respectively. Assume the keyframe sequence as {f1, f2, f3,…….fk} (the total number of the candidate key frames is k), we make use of the following steps to eliminate redundant frames:

a) Use the Prewitt operator to extract edges of the candidate key frames and obtain their corresponding edge images.
b) Set j=2.
c) Calculate the edge matching rate *p(fj-1 , fj)* between the current frame *fj* and the previous frame *fj-1* with the formula (5.8). If *p(fj-1 , fj)* is above average edge matching rate, the current frame *fi* will be marked as a redundant frame.
d) j =j+1, if j>k, go to (e). Otherwise, return to (c) and continue processing the remaining frames.

e) Remove the frames which have been marked as
f) Redundant frames from the candidate key frames.
g) The remaining candidate key frames are the ultimate key frames. With the edge detection and edge matching, we eliminate redundant key frames, improve the accuracy rate of the key frame extraction and reduce the redundancy.

V. EXPERIMAENTAL RESULTS AND DISCUSSION

Proposed Keyframe Extraction Algorithm is implemented with GUI based approach which provides easy user interaction. The GUI of the proposed system is shown in Figure 5.1.

Algorithm work on Avi Video files. Implemented project first loads Avi video file after loading video file it initializes ShotBoundary Algorithm to extract key frames depending input parameters set by user. These Frames will be called as Candidate Keyframes and will be extracted in “Candidate frames” folder. Shot Boundaries will be stored in “SBD” folder. Project then removes redundant Candidate frames by using edge detection Algorithm which will be called as actual keyframes, and will be stored in “Keyframes” folder. Project also provides facility to view extracted Keyframes on GUI itself.

The algorithm Shotboundary used in this project is evaluated and tested by using three different videos CNBC_News.avi, Football_Match.avi, Shrek_3.avi. Video clips are obtained from Internet and various television channels. les are selected since every video provides different pattern to extract Keyframes. Videos are of AVI (Audio Video Interleaved) format.

Figure 5.1: Keyframe Extraction GUI

Following is the example of Keyframes extracted by our proposed algorithm.

Figure 5.2 Example of Keyframes extracted

5.1 CONSTANT PARAMETERS SELECTED FOR EVALUATION

Evaluation is done by keeping total number of frames constant from each video. Threshold Factor is selected as 1 Figure 5.3 shows relation in between Number of frames extracted and Threshold Factor selected. Max Columns and Max Rows is selected as constant as 20.Weight Matrix for assigning weight to different blocks in frame is calculated using rand() function. "prewitt" operator[20] is selected when removing redundant Candidate keyframes in Edgedetection algorithm.

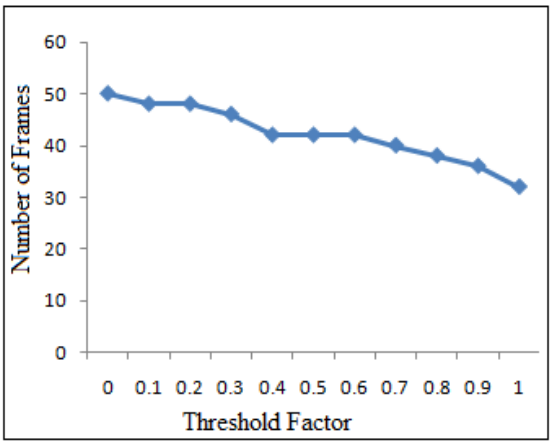

Figure 5.3: Number of Frames vs. Threshold Factor

5.2 SHOT BOUNDARY DETECTION FROM UNCOMPRESSED VIDEO

Following simulation results are plot of frame number versus x^2 histogram matching difference between consecutive frames. Plots represent variation in x^2 histogram matching difference for various effects like cut, fade, dissolve etc.

Case 1: Video:: Shrek_3.avi

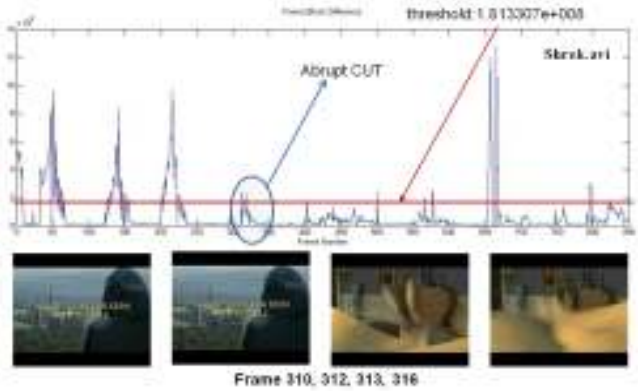

Figure 5.4: Abrupt Cut detection

Case 2: Video:: Shrek_3.avi

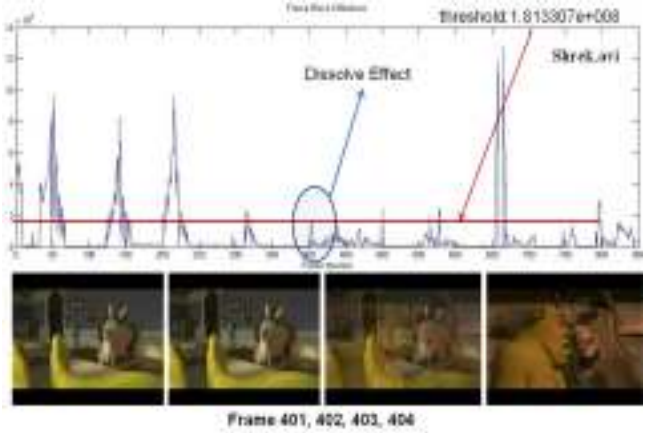

Figure 5.5: Dissolve Effect Detection

Case 3: Video:: Shrek_3.avi

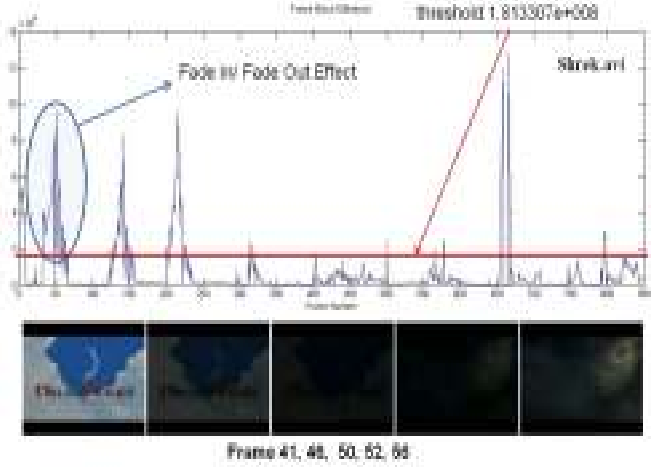

Figure 5.6: Fade In /Fade Out Effect Detection

Case 4: Vi6deo:: Shrek_3.avi

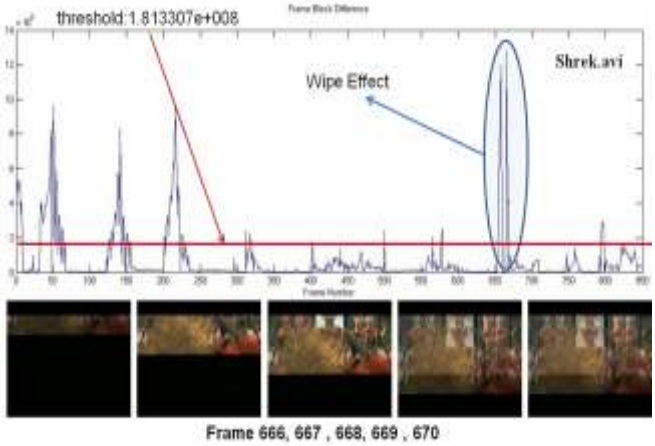

Figure 5.7: Wipe Effect Detection

From the above four cases we can see how $\chi 2$ histogram can be used to detect shot boundaries, It also shows Automatic Threshold calculated by Shotboundary algorithm. Frames above Threshold are extracted as Candidate keyframes.

5.3 **PERFORMANCE EVALUATION**

In order to evaluate the performance of the shot cut detection method presented in Chapter 5, following measures were used.

Definition 1: Recall: the *Recall* measure, also known as the true positive function or sensitivity that corresponds to the ratio of correct experimental detections over the number of all true detections:

Precision Rate = Correctly Detected/(Correctly Detected+ Missed Detected) (5.1)

Definition 2: Precision: the *Precision* measure defined as the ratio of correct experimental detections over the number of all experimental detections:

Recall Rate = Correctly Detected/ (Correctly Detected+ Errorly Detected) (5.2)

In order compare the overall performance of the algorithms, *F* measure, which combines recall and precision results with equal weight.

F(recall, precision) = (2×recall×precision)/(recall + precision) (5.3) Where,

Correctly Detected are the Keyframes correctly detected by Shotboundary algorithm.

Missed Detected are the Keyframes which are missed in while detecting Shot Boundary detection.

Errorly Detected are the Keyframes detected as Boundaries but actually they are not the boundaries of shots.

In further tables 2000 frames of six different video are analyzed and Recall & Precision are calculated and compared.

Following are the notations used in further table.
CT : CUT
DS : Dissolve
FD: Fade
WP: WIPE
AP: Actual present effects in video
DT: Effects detected by proposed algorithm

MS: Effects missed in detection by proposed algorithm
ED: Errorly detected effects by proposed algorithm
RR: Recall Rate
PR: Precision Rate

Results for : Shrek 3_AVI_1_35

	AP	DT	MS	ED	RR	PR
CT	12	9	3	1	90	75
DS	8	6	2	2	75	75
FD	4	4	0	0	100	100
WP	2	1	1	0	100	50

Table 5.1: Recall Rate and Precision Rate for Shrek 3 Video

Following is the example of cut in Shrek3 movie detected by proposed algorithm.

Frame 351, 352

Figure 5.8: Example of cut in Shrek3 Movie

Following is the example of fade in Shrek3 movie detected by proposed algorithm.


Frame 705, 706

Figure 5.9: Example of Wipe in Shrek3 Movie

Results for :Football Match_AVI_1_35

	AP	DT	MS	ED	RR	PR
CT	10	10	0	3	77	100
DS	9	7	1	1	88	88
FD	NIL	NIL	NIL	NIL	NIL	NIL
WP	NIL	NIL	NIL	NIL	NIL	NIL

Table 5.2: Recall Rate and Precision Rate for Football Match Video

Following is the example of Cut detected in Football Match


Frame 714, 715

Figure 5.10: Example of Cut in football Match

Following is the Example of Dissolve in Football Match


Frame 413

Figure 5.11: Example of Dissolve in Football Match

Results for :CNBC News_AVI_51

	AP	DT	MS	ED	RR	PR
CT	1	1	0	0	100	100
DS	NIL	NIL	NIL	NIL	NIL	NIL
FD	NIL	NIL	NIL	NIL	NIL	NIL
WP	3	3	0	1	75	100

Table 5.3: Recall Rate and Precision Rate for CNBC News Video

Following is the Cut detected in CNBC News by proposed algorithm.

Frame 296, 297

Figure 5.12: Example of Detected Cut in CNBC News

Following is the Errorly detected Wipe in CNBC News by proposed algorithm

Figure 5.13: Example of Errorly detected wipe in CNBC News

Video Name	Shrek 3 Trailer		Football Match		CNBC News		Average	
	RR	PR	RR	PR	RR	PR	RR	PR
CT	90	75	77	100	100	100	92	90
DS	75	75	88	88	NIL	NIL	85	88
FD	100	100	NIL	NIL	NIL	NIL	100	100
WP	100	50	NIL	NIL	75	100	94	88
TOTAL AVERAGE							**93**	**92**

Table 5.4: Summary of Experimental Results

Above table shows some of the video which do not have either of the effect like cut, Dissolve Fade or Wipe there NIL is entered. The values of precision Rate and Recall Rate may vary depending on sample video selected for experiments. From above tables we can see that proposed algorithm is better in detecting cuts and wipe effect, it also detect Dissolve effect and Fade effect.

	Our Algorithm	Approch in [1]	Approch in [2]
Recall Rate in %	93	90	88
Precision Rate in %	92	92	75

Table 5.5: Comparison of Recall Rate and Precision Rate

The video sequences are of various shots. Generally speaking, there are two kinds of shots: abrupt shot and gradual shot. The temporal and spatial down sample is performed to 25 fps and 320×240 pixels to improve simulation speed. Table 5.5 shows that our algorithm can detect all the abrupt shots and most gradual shots, and, recall and precision are 93% and 92%, respectively. The performance is better than many current algorithms.

VI. CONCLUSION

In this project, we proposed Block based $\chi 2$ Histogram algorithm for shot boundary detection. Shot boundary detection and key frame extraction system using image segmentation is a novel approach for video summarization. First video is segmented in frame, then employed different weights to compute the matching difference and threshold. By using the automatic threshold, boundaries are detected. We detect various shot boundaries like Cut, Fade and Dissolve with the help of $\chi 2$ histogram matching difference between consecutive frames and automatic threshold. Experimental results show that the proposed algorithm gives satisfactory performance for shot boundary detection. The contributions and characteristics of the proposed approach are summarized below:

- Efficiency: Easy to implement and fast to compute. Only the x^2 Histogram is o be found out in order to extract keyframes.
- No Redundant Keyframes: As Redundant Key Frames are removed use Edgedetection algorithm Information per Keyframe has improved and time to summarize video will also increase.
- Detection of Zoom: Proposed algorithm is also efficient in detecting Keyframes with Zoom and Object in front of camera effect.
- Higher Recall & Precision: Algorithm provides higher Recall Rate and Precision Rate as compared to the algorithm proposed in [1,2].

In order to further improve accuracy, Graph Partition Model with Support Vector Machine can be used. Graph Partition Model will be the further direction.

VII. FUTURE WORK

x^2 Histogram and Automatic Threshold approach is very efficient for detecting. We can further improve the performance of algorithm by using Graph Partition Model.

Graph Partition Model with Support Vector Machine is new research area in Shot boundary detection. Graph theoretic segmentation algorithms are widely used in the fields of computer vision and pattern recognition. Segmentation with graph partition model is one of the graph theoretic segmentation algorithms, which offers data clustering by using a graph model. Pair-wise similarities between all data objects are used to construct a weighted graph as an *adjacency matrix* (*weight matrix* or *similarity matrix*) that contains all necessary information for clustering. Representing the data set in the form of an edge-weighted graph converts the data clustering problem into a graph partitioning problem.

REFERENCES

[1] ZHAO Guang-sheng "A Novel Approach for Shot Boundary Detection and Key Frames Extraction", 2008 International Conference on Multimedia and Information Technology.

[2] Naimish.Thakar " Analysis and Verification of Shot Boundary Detection in Video using Block Based $\chi 2$ Histogram Method" International Journal of Advances in Electronics Engineering.

[3] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 1280–1289,Dec. 1999.

[4] R. Zabih, J. Miller, and K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," Proc. ACM Multimedia 95, pp. 189-200, 1995.

[5] A. Hanjalic, Content-based Analysis of Digital Video, Boston: Kluwer Academic Publishers, 2004.

[6] B. T. Truong, "Shot Transition Detection and Genre Identification for Video Indexing and Retrieval," Honours, School of Computing, Curtin University of Technology, 1999.

[7] B. C. Song, and J. B. Ra, "Automatic Shot Change Detection Algorithm Using Multi-stage Clustering for MPEG-Compressed Videos," Journal of Visual Communication and Image Representation, vol. 12, no. 3, pp. 364-385, 2002.

[8] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "A Unified Approach To Scene Change Detection In Uncompressed And Compressed Video," IEEE Transactions on Consumer Electronics, vol. 46, no. 3, pp. 769-779, 2000.

[9] F. Arman, A. Hsu, and M. Y. Chiu, "Image processing on compressed data for large video databases," Proc. ACM Multimedia, pp. 267-272, 1993.

[10] E. Deardorff, T. Little, J. Marshall et al., "Video scene decomposition with the motion picture parser," IS&T/SPIE, vol. 2187, pp. 44-45, 1994.

[11] N. Vasconcelos, and A. Lippman, "Statistical models of video structure for content analysis and characterization," IEEE Transactions on Image Processing, vol. 9, no. 1, pp. 3-19, 2000.

[12] B. Kayaalp, "Video Segmentation Using Partially Decoded MPEG Bitstream," METU, Ankara, 2003.

[13] 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.

[14] Y. Zhuang, Y. Rui, T. S. Huan, and S. Mehrotra, "Adaptive key frame extracting using unsupervised clustering," in Proc. Int. Conf. Image Processing, Chicago, IL, 1998, pp. 866–870.

[15] Zuzana Cerneková, Ioannis Pitas "Information Theory-Based Shot Cut/Fade Detection and Video Summarization" in IEEE proc. in circuits and systems for video technology, VOL. 16, NO. 1, JANUARY 2006.

[16] Ali Amiri and Mahmood Fathy "Hierarchical Keyframe-based Video Summarization Using QR-Decomposition and Modified k-Means Clustering" in Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing, Volume 2010.

[17] A. Hanjalic, "Shot Boundary Detection: Unraveled and Resolved?," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no.2, pp. 90-105, February 2002.

[18] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in Proc. 6th Int. ACM Multimedia Conf., Bristol, U.K., 1998, pp. 211–218.

[19] N. Doulamis, A. Doulamis, Y. Avrithis, and S. Kollias, "Video content representation using optimal extraction of frames and scenes," in Proc. IEEE Int. Conf. Image Processing, Chicago, IL, 1998, pp. 875–878.

[20] Kintu Patel, "Key Frame Extraction Based on Block based Histogram Difference and Edge Matching Rate", International Journal of Scientific Engineering and Technology, Volume No.1, Issue No.1 pg:23-30

[21] A. Hanjalic, Content-based Analysis of Digital Video, Boston: Kluwer Academic Publishers, 2004.

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

AUTHORS

First Author – Mr. Sandip T. Dhagdi, M.E. (C.E. Second Year), Sipna COET, Amravati, Email: sandip.yml@gmail.com
Second Author – Dr. P.R. Deshmukh, Professor Computer science & IT Department, Sipna COET, Amravati, Email: pr_deshmukh@yahoo.com