Temporal Video Scene Segmentation

A

Dissertation

Submitted in partial fulfilment of the requirements
for the award of degree

Of

Master of Technology
in
Computer Science and Engineering

Submitted by

Manoj Kumar
Enrolment No. – 14535025

Under the guidance of

Dr. R. Balasubramanian

Department of Computer Science and Engineering,

Indian Institute of Technology, Roorkee

Roorkee – 247667, India

May-2016

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled **"Temporal Video Scene Segmentation"** towards the partial fulfilment of the requirements for the award of the degree of Master of Technology with specialisation in Computer Science Engineering submitted in the department of Computer Science and Engineering, Indian Institute of Technology, Roorkee (India) , is an authentic record of my own work carried out during the period of August 2015 to May 2016 under the guidance of **Dr. R. Balasubramanian**, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee.

I have not submitted the matter embodied in this dissertation for the award of any other degree or Diploma.

Date :

Place : MANOJ KUMAR

CERTIFICATE

This is to certify that the above statements made by the candidate is correct to the best of my knowledge and belief.

Date :

Place :

Dr. R. Balasubramanian,
Associate Professor,
Department of Computer Science and Engineering,
IIT Roorkee

ACKNOWLEDGEMENT

It gives me pleasure to thank all those people who have, at various stages and in various ways have played a key role in successful completion of this work. I would take this opportunity to extend my heartfelt gratitude to my guide and mentor Dr. R. Balasubramanian, Associate Professor, Indian Institute of Technology, Roorkee for his invaluable advice, guidance, encouragement and for sharing his knowledge. His wisdom and commitment to the highest standards motivated me throughout. He has been very generous in providing the necessary resources to carry out the research. He an inspiring teacher, a great advisor and most importantly a person.

I am also highly indebted to all my friends, and my family who gave me the moral support and valuable suggestions. On a personal note, I owe everything to the almighty.

MANOJ KUMAR

ABSTRACT

This Dissertation report discusses one of the computer vision problem, which is implemented using various video and image processing techniques from last two decades that is “TEMPORAL VIDEO SCENE SEGMENTATION”. From last decade a lot of work has been done on automatically video scene segmentation. In field of computer vision this problem has received most popularity because it is the first and most important part of the other problems like video summarization, video indexing and browsing etc. This dissertation includes a new approaches to solve this problem which consists of different feature detector and descriptors, clustering algorithms, window based scene boundary defining etc. As we know scenes are the grouping of semantically similar shots which are temporarily close. So our first task is divide the video into shots, which includes cluster of similar frames. This task is done using HSV color histogram. Now we extract some visual, motion and SIFT features of the shots and calculated inter shot pairs similarity, which is interpreted with shot similarity graph (SSG). At the final stage using sliding window method we have grouped similar shots which are higher similarity then some decided threshold. This grouping has considered inverse time proximity. This dissertation has also include some of implemented papers with their different approaches and achieved results are also discussed.

TABLE OF CONTENTS

CANDIDATE’S DECLARATION i

ABSTRACT iii

LIST OF FIGURES vi

LIST OF TABLES vii

1 INTRODUCTION 1

1.1 Video scene segmentation an important problem for several reasons[6] 2

1.2 Thesis statement 2

1.3 Thesis motivation 2

1.4 Purposed Method 3

2 LITERATURE SURVEY 5

2.1 Rule based video scene segmentation 5

2.2 Graph based video scene segmentation 5

3 PROPOSED WORK 8

3.1 Terminologies used 8

3.1.1 Video 8

3.1.2 Shots 8

3.1.3 Key Frames 9

3.1.4 Scene 9

3.1.5 HSV Histograms 9

3.1.6 Scale Invariant Features Transform (SIFT FEATURES)[8][15] 11

3.2 Flow Chart For Proposed work 15

3.2.1 Defining shot boundary 15

3.2.2 Defining Scene boundary 16

3.3 Defining Shot Boundary 16

3.3.1 Key Frame Extraction 18

3.3.2 Defining shot similarity graph for temporary shots 20

3.3.3 Grouping of temporary shots to make permanent shot using sliding window method 21

3.4 Scene detection using Sliding window based method 23

3.4.1 Visual Similarity between shot pairs 23

3.4.2 Motion similarity between shot pairs: 23

3.4.3 SIFT Similarity between shot pairs: 24

3.4.4 Inverse time proximity between shot pairs: 25
3.4.5 Construction of Shot Similarity Graph (SSG): 25
3.4.6 Defining scene boundaries using sliding window method: 26
3.5 Selection of sliding window 27
4 EXPERIMENTS AND RESULTS 31
4.1 Experiments and Results 32
4.2 Setting of threshold values 37
4.3 Comparison of F-score values for various previous approaches 39
5 APPLICATIONS AND ISSUES IN VIDEO SCENE SEGMENTATION 40
5.1 Applications: 40
5.2 Some major challenges in video scene segmentation 41
6 CONCLUSION AND FUTURE WORK 42
REFERENCES 43

LIST OF FIGURES

Figure 1.1 Video modules.. 1

Figure 1.2 Scene segmentation of video overview 4

Figure 2.1 Block diagram for GSTG[2]... 6

Figure 3.1 HSV Histogram measures 10

Figure 3.2 Cylendrical meausre of HSV histogram color model... 10

Figure 3.3 SIFT keypoints are circular image regions with an orientation 12

Figure 3.4 The SIFT descriptor is a spatial histogram of the image gradient.............................. 13

Figure 3.5 VLFeat conventions[8] 14

Figure 3.6 Defining Shot Boundaries 15

Figure 3.7 Defining Scene Boundaries 16

Figure 3.8 One Frame selected as key frame for a shot... 19

Figure 3.9 Multiple key frames selected in a shot due to motion or some chasing actions.......... 19

Figure 3.10 (Sliding window): This figure shows that shot5 is similar to any of the shot2 or shot3 or shot4, So shot5 is considered as part of the previous permanent shot. Window is slides to check for next shot6 22

Figure 3.11 (sliding window): This figure shows that shot5 is less similar than threshold value so from shot5 a new permanent shot is begins 22

Figure 3.12 Sliding window... 27

Figure 3.13 Shot 4 is the part of scene m.. 28

Figure 3.14 Shot 4 is not part of scene m, so new sliding window created and shot 4 is in scene (m+1) , now checking for shot5 29

Figure 4.1 Result Comparison between without sift features and with sift features 35

Figure 4.2 Showing results for a movie(union is strength)... 36

Figure 4.3 Tuning of threshold value(δ) for scene boundaries 38

Figure 4.4 Comparing various approaches results 39

LIST OF TABLES

TABLE	TABLE CAPTION	PAGE NO.
1	Ground truth for the dataset (Video Details)	31
2	Results using Visual and Motion Similarity	32
3	Results using Visual, Motion and SIFT Similarity	33
4	Result Comparision between without sift features and with sift features	34

1 INTRODUCTION

Video Scene segmentation and video shot segmentation on the basis of their semantic information is the important problem in the field of computer vision and video processing. More than last 20 years, this problem is being solved by different researchers and scientists of this field, they have done a great work using different techniques of video and image processing and machine learning[1].

All types of video are constructed in a hierarchical modules these modules are:

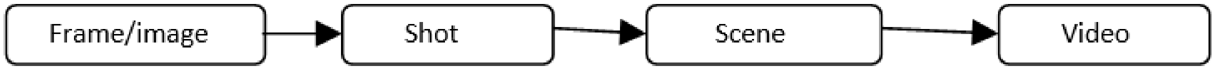


Figure 1.1 Video modules

Frame: frames or images are the elementary module of the video. Video is nothing but storing the sequence of images and then displaying them at a suitable rate which is adjustable to human eyes. Mostly commercial and educational video are plays at 25 to 32 frames per second.

Shot: Shots are collection of continues captured images or frames without any interruption of camera and background. Shot represents the contiguous action in the space and time. The shot has very less information about the video theme[1][2][4].

Scene: Scene are the collection or grouping of similar shots. Which has semantic meaning of the video parts which are coherent to some object or theme of the video[1][2][3].

Video: Video is the collection of scenes in temporal fashion. Video has full information about the theme on which it is recorded or captured on.

During recent years, we are producing a tremendous amount of data on the daily basis. For efficient management of this data, researchers are doing a very good work in saving, retrieval and browsing of the different multimedia and other files like images, videos and documents etc., on the basis of their semantic means. If we talk about only videos on the internet we have a huge amount videos producing by different application like satellite, surveillance cameras, movie industries, home videos, educational videos and many other type video. These videos are not in saved in some structured libraries. For future video libraries, the scene segmentation and shot segmentation provides the semantic means of the video, which can use for efficient content based management of the digital video.

1.1 Video scene segmentation an important problem for several reasons[6]

1. Video scene segmentation is the first step toward automatic video a notation of video sequences.
2. The scene is the first step towards semantic understanding of the entire video.
3. Video summarization, based on the scenes.
4. Video indexing and making highlights of sports or any videos.
5. Browsing or reusability of indexed video segments.
6. Breaking up a long video into scenes will allow for non-linear navigation of video data. For example, scene segmentation will be useful for browsing news programs.

1.2 Thesis statement

My Thesis work has entitled as “Temporal Video Scene Segmentation”, so my focus of work was on the segmenting the video into the meaningful parts segments of the video which are named scene. Generally scene can be defined as the grouping of related or similar action part of the video, these parts may be taken anywhere from the video along time line. Means if any character is acting at the staring of video and then same character acting at the ending part of video at same background then we may group these two related action as scene. But my work is to grouping the similar action shots into a single scene which are temporally closed to each other. Means if two shots of same action are recorded one after another in the video only then I will group them into a single scene. If as earlier stated shots are far to each other according to time line then those shots are labeled as two different scene of two different context or dialog.

1.3 Thesis motivation

As we know computer vision community has done a lot of work in the field of image and video processing. For higher level of task of videos like summarization of the video, annotation and tagging of the video, searching, retrieval and browsing of the video according to their semantic meanings, making of highlights of the sports video and many more are required preprocessing of given video. The end results of these high level tasks are depended on the fundamental units like key frames, shots and scenes. If any preprocessing algorithm provides good fundamental units, then accuracy of results of these tasks are automatically raises. So my theme work is to provide the temporal scenes of video with higher accuracy.

1.4 Purposed Method

All video processing tasks are starts with very basic unit of video that is frame. First of all all the frames of video are extracted and then take some low level and high level features from those frames. On the basis of those features we can proceed with our ultimate task.

Temporal video scene segmentation is also divided into 6 main parts:

1. Features detection and extraction of the video frames.
2. Defining of shot boundaries on the basis of extracted features.
3. Key frames extraction from each shots.
4. Extract some features of those extracted key frames.
5. Find the similarity between all shots pairs.
6. Defining the boundaries of scene on the basis of similarity of shots.

For extraction of features I have used different techniques like:

1. For features extraction I have used HSV Color Histogram, SIFT Features.
2. Visual and motion features of the frames are also extracted for calculating the similarity between shot shots.
3. Sliding Window method is used for defining the scene boundary.

In this thesis work I have used some movies, house video, and randomly recorded videos. Using purposed method I have reached to good results. Results are calculated in the form of precision and recall. I have implemented my proposed framework in MATLAB15b[16].

This is the overall picture presentation of the video scene segmentation:

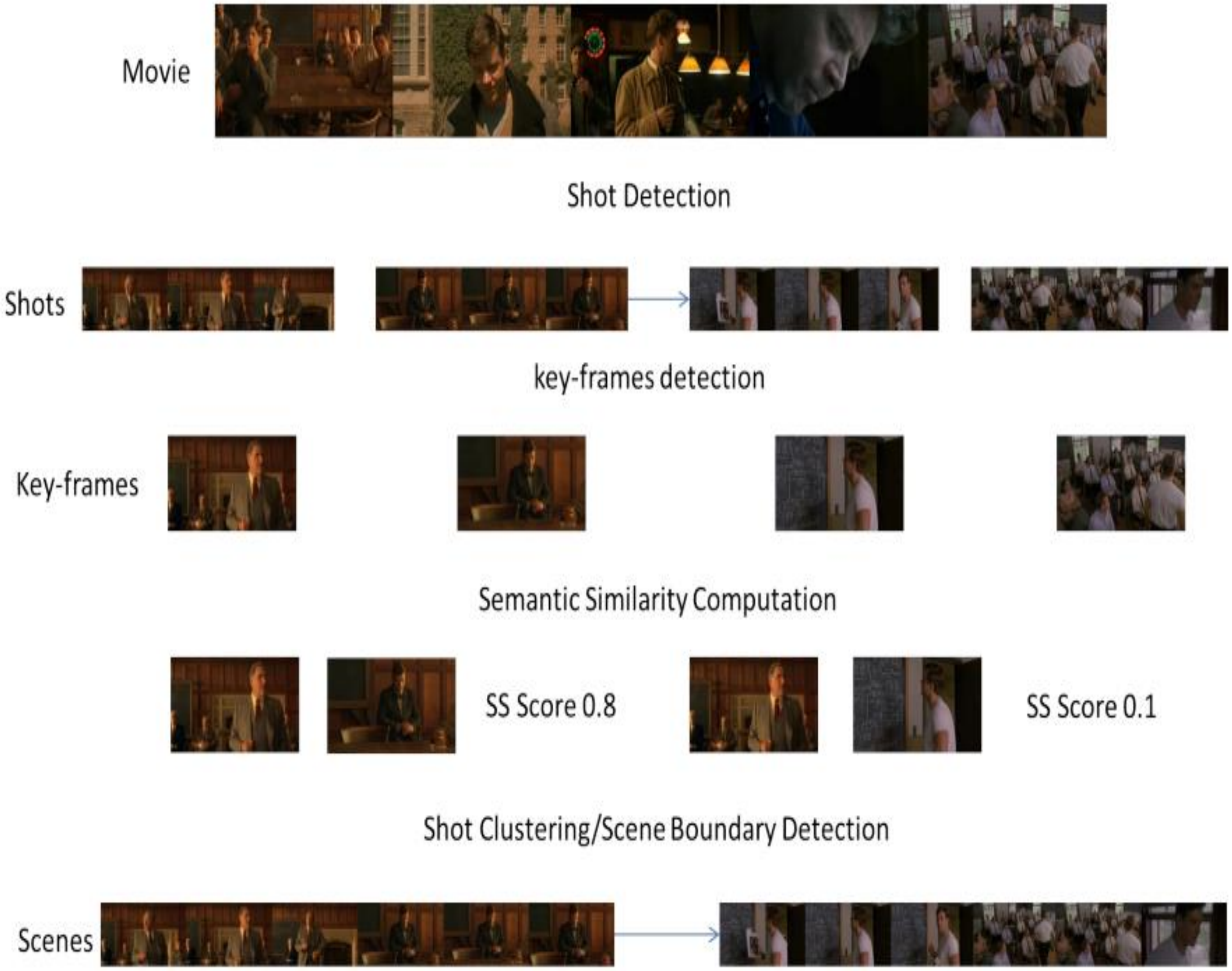


Figure 1.2 Scene segmentation of video overview

This figure depicts the flow of temporal scene segmentation. First shot boundary are detected and movie divided into shots, and then extract the key frames of shots that represents the shot. Using some semantic similarity score (SS score) find the similarity between consecutive shots and then finally using some techniques of graph cuts find the scene boundary, and got segmented scene[6].

2 LITERATURE SURVEY

Computer vision community has done a great work in video processing and image processing. Video Scene Segmentation is very well known problem of the video processing because of some higher level tasks require video scenes as the fundamental unit. Previously shots and frames were used as fundamental unit for video processing but they did not come up with significant accurate results. Shots and frames are not contains much of information of the video. Whereas scene has a good semantic meaning of the video parts, so scenes as a fundamental unit for higher video processing task provides good results[4][6].

Mostly there are two types of approach for video scene segmentation [6]

1. Rule based and
2. Graph based.

These approaches are rely on visual, textual, audio, color features.

2.1 Rule based video scene segmentation

In rule based approach the different authors has first select the specific videos and video format which has the predefined structure of the scene in the video making. In these videos scene is structured in professional movie production. Zhu and Liu et al. [3], for example, propose a visual based probabilistic framework that works on MPEG format videos, and present an approach for the segmentation of continuously recorded TV broadcasts videos. In this paper they have considered temporal constraints and color texture features. This type of approach have limitations that if any movie maker intentionally change the structure of scene in the video then this approach give bad results.

2.2 Graph based video scene segmentation

In Graph based approach first video is divided into key frames or shots using some key features like color histograms. Shots are arranged in a graph representation and then clustered by partitioning the graph. The Shot Transition Graph (STG), used in [1] [2], is one of the most used models in this category: here each node represents a shot and the edges between the shots are weighted by shot similarity. Then split this STG into sub graphs by applying the normalized cuts for graph partitioning.

In [2] Panagiotis Sidiropoulos, Vasileios Mezaris et al., used both high level and low level audiovisual features. In this framework authors used Global Scene Transition Graph (GSTG) of both feature of video that is audio part and visual part. For video part they used color HSV color histograms to detect the boundaries of initial shots. Then using primary linking, secondary linking and trivial link, trivial double link techniques redefine the shot boundaries based on the similarities. After defining inter shot similarity they created multiple STG graphs of low level and visual concept features. Simultaneously on audio part collect some audio features and event detectors, and using these parameters creates multiple STGs of audio part too. Finally all using STGs graphs cut algorithms scene boundaries are defined with two separate audio and video part. At last use some probabilistic approach to combine their results into actual scene segmented output.

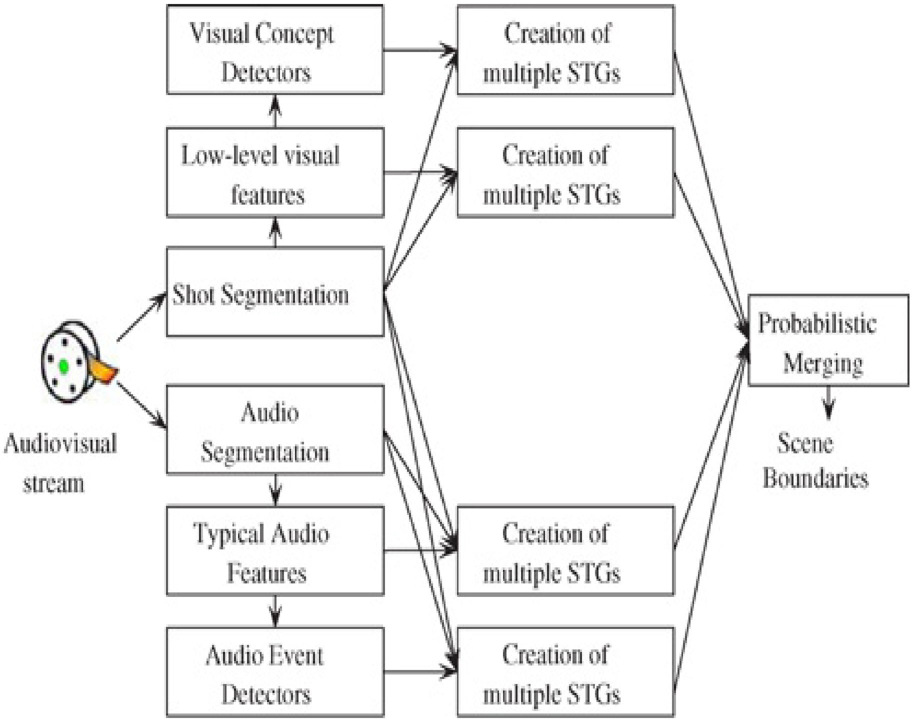


Figure 2.1 Block diagram for GSTG[2]

In [6] this, author extracted the key frames using color histograms comparison of each successive frames and then using k-means clustering he found the shot representative key frames. For further steps he used BOW (Bag of Words) model, in the context of video features we can say

BOF (Bag of Features). The set of all the visual words for a movie is the visual vocabulary for that movie. Visual words are computed by taking all the key frames from all the shots and applying clustering on the SIFT features extracted from those key frames. Now a shot S_i can be represented as a K dimension vector (histogram) where S_{ij} gives the count of j^{th} visual word in i^{th} shot and K is the total number of visual words or vocabulary.
Now every shot is processed in temporal fashion for their inter similarity, so that author can group these shots and finalize the scene. For further good result he has also used Bipartite Graph Model (BGM) to makes scene of shots.

Zeeshan Rasheed and Mubarak Shah [1] in 2005 they have used 16 bin Color HSV histogram to extract the visual or color information of the frames and then according to some threshold value between 80% – 90%, they defines the shot boundaries. They collects more than 1 key frame of each shot. To find the inter shot similarity they have used 2 features (Visual similarity and motion similarity) which are depended on the color histogram. At last for defining the scene boundaries according to temporal fashion, they uses inverse time proximity as another features. It means as time between two shots are increases similarity between those shots are decreases. Combining of the features they creates are SSG (Shot Similarity Graph), which show all pair shots similarity. Using iterative method of normalized graph cut technique they partition the graph into the sub graphs. All shots within a single graph has more intra similarity so they grouped together and makes a scene.

In Video Scene segmentation using markov chain Monte Carlo (MCMC) technique [3], Yun Zhai and Mubarak shah uses different techniques to segments the scene. All other techniques uses static threshold values. But in this paper authors uses two parameters to define the scene boundary those are: jumping and fusion. Authors tested this technique on two type of videos: home video and feature films. Initially video is divided into 2 scenes and then repetitively use MCMC to on each part to define weak and strong scene boundaries. In this framework author makes three types of update in each iteration using two parameters

1 shuffling of boundaries of video shots

2 merging of two adjacent scenes.

3 splitting of one scene into two scenes.

At last all local strong scene boundaries are detected and make them permanent scenes.

3 PROPOSED WORK

After understanding of problem statement and going through previous work done of this field, I came with novel method which is combination of two techniques first define the shot boundary using some low level visual features and then grouping of the similar shots. The proposed method entitled as two pass sliding window method for video scene segmentation[1][2].

Proposed framework is done in following three parts:

1. Define the shot boundaries for a video using HSV color Histogram and Structure similarity score.
2. Find the similarity between all pair shot using SIFT Features, visible similarity, motion similarity and inverse time proximity and create a SSG (Shot Similarity Graph).
3. Define the scene boundaries using sliding window method.

3.1 Terminologies used

3.1.1 Video

A video is nothing but collection of continuous images. A video may or may not have audio data. Now a days videos is available in different formats. Format of any video is basically a container or data structure which stores the video frames, audio signals and metadata in compressed form. The metadata is the structural information of the compressed form of audio-video part of the video. Metadata contains information like title of the video file, total number of frames in video, Frame rate of the video, pixel size of the frames, format techniques used for the video etc. Here are some video file format like: AVI (Audio Video Interleave), MPEG (Moving Picture Experts Groups), FLV, WMV, MPEG-4 etc[8].

Most of the formats are supported by Matlab, for experimental purposes I have used .avi and .mp4 file format.

3.1.2 Shots

After frames of the video shots are next fundamental unit of the video. A shot is collection of similar frames, means collection of temporarily sequence of frames taken without any interruption by single camera. Movie character and background are same during that particular shot or action recoding[6].

$$S_i = [f1, f2, f3, f4 \dots \dots \dots fn] \text{ where } i = \{1,2,3,4,5\dots\dots n\} \qquad 3.1$$

a shot has multiple similar frames, it means intra similarity of the shot is maximize and the inter shot similarity is minimized that's why movies can be divided into shots.

3.1.3 Key Frames

Each shot has similar type of frames in it which has very high intra similarity. If we consider all frames for higher level of video processing then it will takes a huge amount of time, space and computation power. [1]To reduce these computation time we can extract some of the frames as key frames from each shot. We select only those frames which has maximum information about that particular shot and redundancy of similar frame is minimized up to some decided threshold limit. Technically and view point of video processing shots are represented by those extracted key frames.

3.1.4 Scene

A scene is the next important fundamental unit of the video, which has more semantic information of the video than a shot has. So technically a scene is the grouping of similar shots. So a scene can be defined as the collection of temporally contiguous and related shot which are overlapped by some similar contents like character, background or same action. Technically we have to define some threshold value if up to that extend of threshold temporally closed shots are similar to each other then those shots can be grouped and called them as a scene[1][2][6].

$$Scene_i = [S1, S2, S3, f4 \dots \dots \dots Sn] \qquad 3.2$$

where $Scene_i$ is the i^{th} scene and $i = \{1,2,3,4,5\dots\dots n\}$ *and*

S_j *are the Shots and* $j=\{1,2,3,\dots\dots m\}$, *where* $m>=n$

3.1.5 HSV Histograms

HSV stands for Hue, Saturation, and Value, and is also often called HSB (B for brightness). HSV color model is considered better then RGB color model because in RGB model give the detail of only red, green and blue colors present in the image. While HSV describes the additive blend of colors and other simplistic characteristics like tint, shades and tones. HSV Histogram will gives the pixel intensity of H,S and V and plot the graph between H,S,and V values and number of cell or pixel that are of same value[11][14].

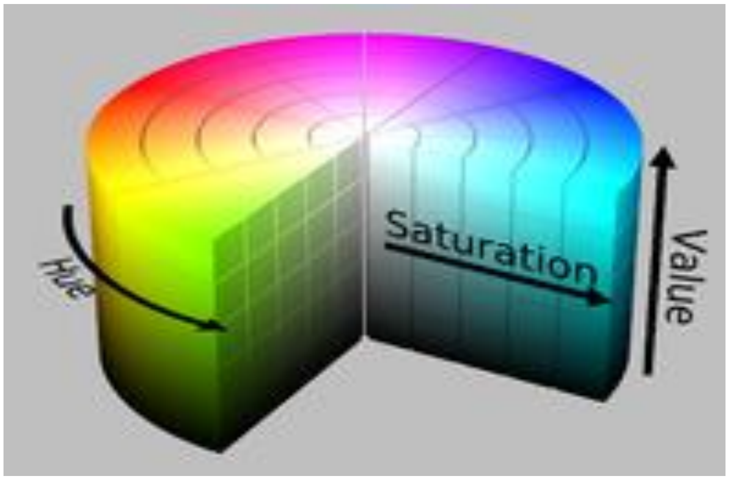


Figure 3.1 HSV Histogram measures

It has cylendrical measure of the HSV values :

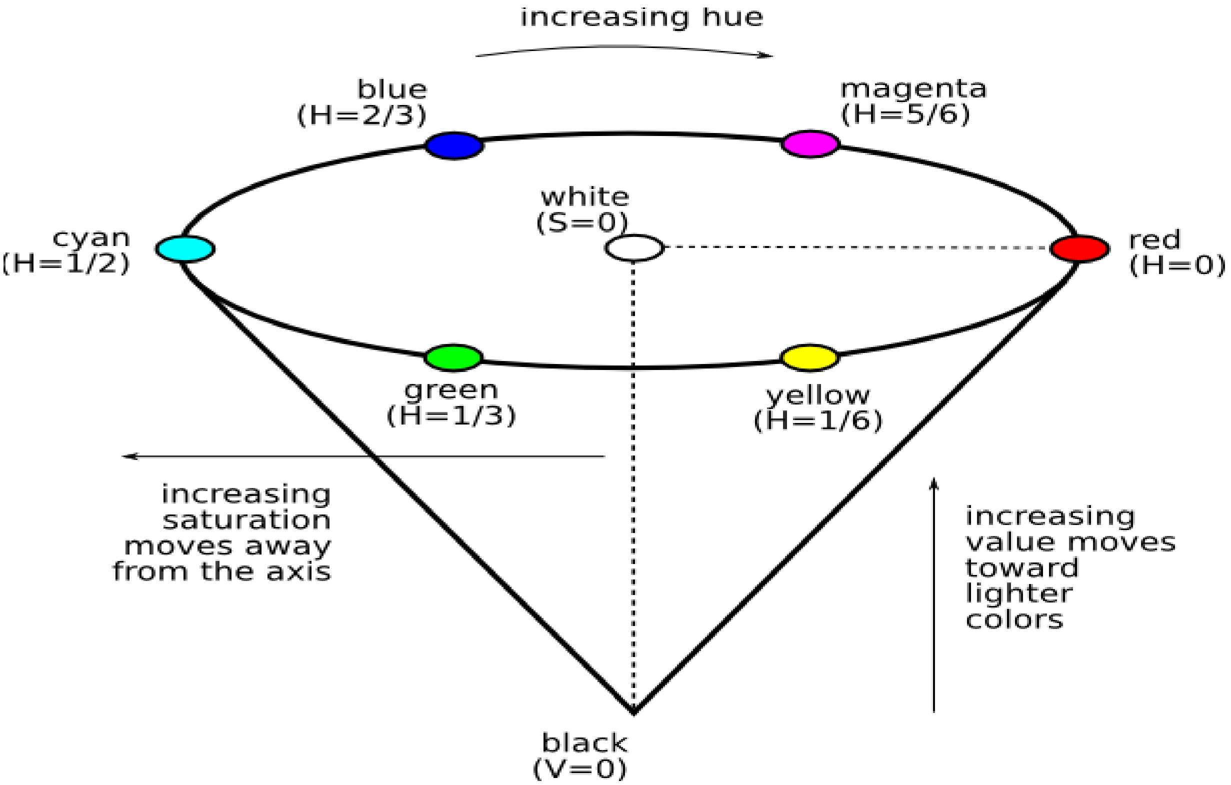


Figure 3.2 Cylendrical meausre of HSV histogram color model

Hue : Hue describe the pureness of color, it has range from 0 to 1. All pure colors are represents by 1 and the fadness aproches to 0. All the shades and tint of red color has same hue value. On the color wheel of HSV model there are total 360^o angles, so at different range of angles represents the different color like red starts at 0^o , yellow starts at 60^o, green starts at 120^o,cyan starts at 180^o, blue starts at 240^o, and magenta starts at 300^o.

Saturation : saturation describes the whiteness of the color or how much amount of gray (0% to 100%) in the color. The saturation value of white color is 0 and saturation value of Red color is 1, and all the tint and shades of red color has satureation values less then 1.

Value : Value discrive the lightness of the color or how dark the color is. The value of black color is 0. And the value of white is 1. It means if the lighness of the color is increases then its value will dicreases towards 0.

3.1.6 Scale Invariant Features Transform (SIFT FEATURES)[8][15]

A SIFT feature is a selected image region (also called keypoint) with an associated descriptor. SIFT feature of an image is calculate and generated with following two methods.

1. **SIFT Detector (Keypoints detection):** A SIFT *keypoint* is a circular image region with an orientation. It is described by a geometric*frame* of four parameters: the keypoint center coordinates *x* and *y*, its *scale* (the radius of the region), and its *orientation* (an angle expressed in radians). The SIFT detector uses as keypoints image structures which resemble “blobs”. By searching for blobs at multiple scales and positions, the SIFT detector is invariant (or, more accurately, covariant) to translation, rotations, and re scaling of the image.

   The keypoint orientation is also determined from the local image appearance and is covariant to image rotations. Depending on the symmetry of the keypoint appearance, determining the orientation can be ambiguous. In this case, the SIFT detectors returns a list of up to four possible orientations, constructing up to four frames (differing only by their orientation) for each detected image blob[8][15].

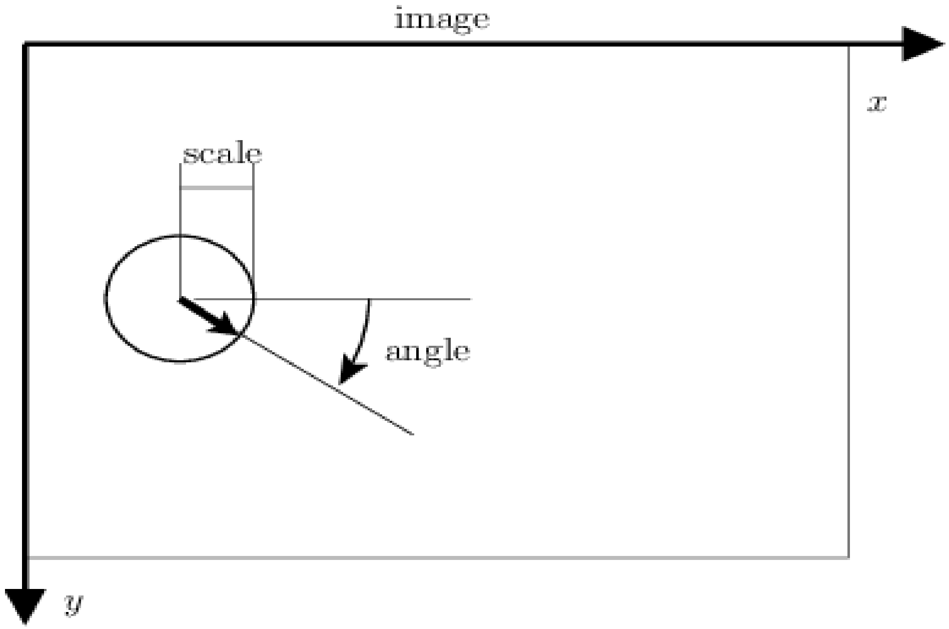


Figure 3.3 SIFT keypoints are circular image regions with an orientation

There are several parameters that influence the detection of SIFT keypoints. First, searching keypoints at multiple scales is obtained by constructing a so-called “Gaussian scale space”. The scale space is just a collection of images obtained by progressively smoothing the input image, which is analogous to gradually reducing the image resolution. Conventionally, the smoothing level is called *scale* of the image. The construction of the scale space is influenced by the following parameters[8][15]:

- **Number of octaves**. Increasing the scale by an octave means doubling the size of the smoothing kernel, whose effect is roughly equivalent to halving the image resolution. By default, the scale space spans as many octaves as possible (i.e. roughly log2(min(width,height)), which has the effect of searching keypoints of all possible sizes.
- **First octave index**. By convention, the octave of index 0 starts with the image full resolution. Specifying an index greater than 0 starts the scale space at a lower resolution (e.g. 1 halves the resolution). Similarly, specifying a negative index starts the scale space at an higher resolution image, and can be useful to extract very small features (since this is obtained by interpolating the input image, it does not make much sense to go past -1).

- **Number of levels per octave**. Each octave is sampled at this given number of intermediate scales (by default 3). Increasing this number might in principle return more refined keypoints, but in practice can make their selection unstable due to noise.

- Keypoints are further refined by eliminating those that are likely to be unstable, either because they are selected nearby an image edge, rather than an image blob, or are found on image structures with low contrast. Filtering is controlled by the follow:

- **Peak threshold.** This is the minimum amount of contrast to accept a keypoint.
- **Edge threshold.** This is the edge rejection threshold.

2. **SIFT Descriptor:** A SIFT descriptor is a 3-D spatial histogram of the image gradients in characterizing the appearance of a keypoint. The gradient at each pixel is regarded as a sample of a three-dimensional elementary feature vector, formed by the pixel location and the gradient orientation. Samples are weighed by the gradient norm and accumulated in a 3-D histogram *h*, which (up to normalization and clamping) forms the SIFT descriptor of the region. An additional Gaussian weighting function is applied to give less importance to gradients farther away from the keypoint center. Orientations are quantized into eight bins and the spatial coordinates into four each, as follows[16]:

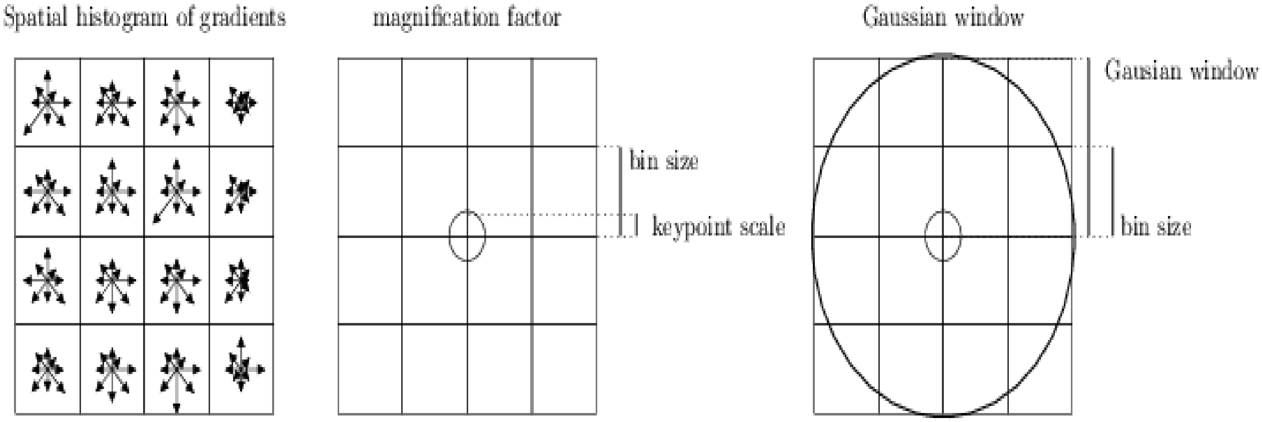


Figure 3.4 The SIFT descriptor is a spatial histogram of the image gradient

VLFeat SIFT descriptor uses the following convention. The *y* axis points downwards and angles are measured clockwise (to be consistent with the standard image convention). The 3-D histogram (consisting of 8×4×4=128 bins) is stacked as a single 128-dimensional vector, where the fastest varying dimension is the orientation and the slowest the *y* spatial coordinate. This is illustrated by the following figure[13][16].

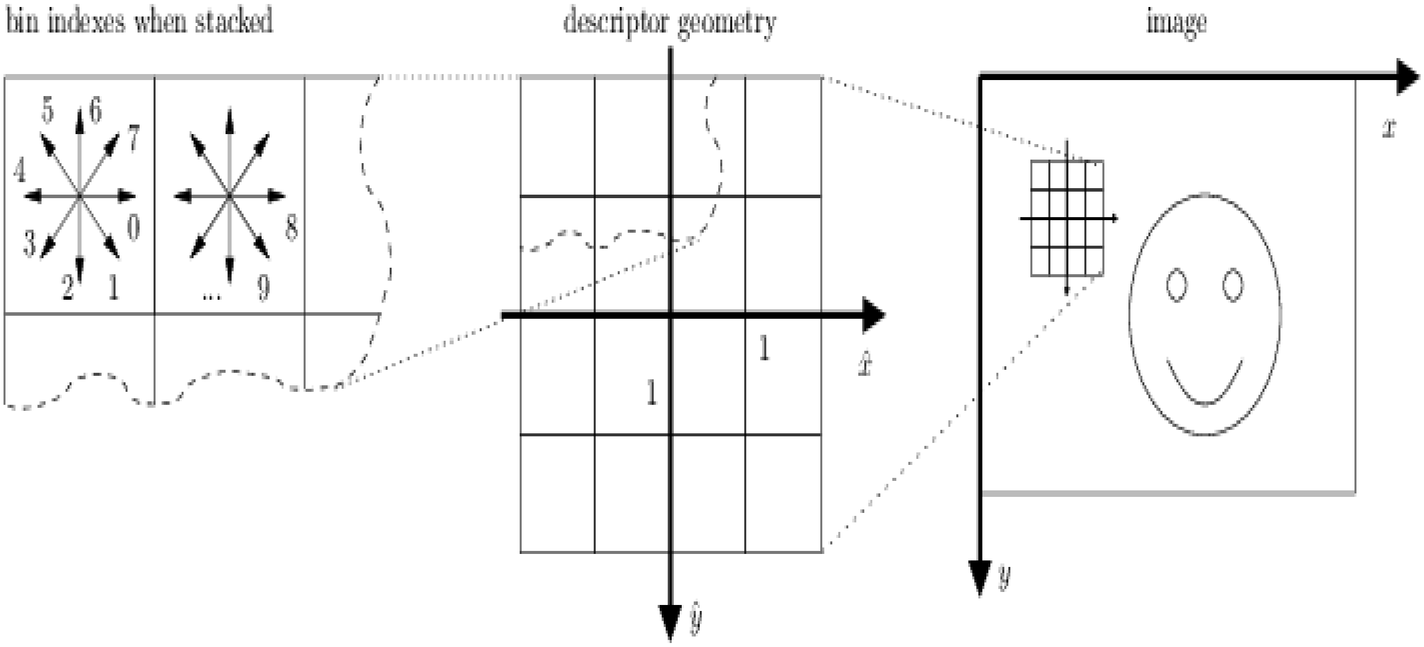


Figure 3.5 VLFeat conventions[8]

3.2 Flow Chart For Proposed work

3.2.1 Defining shot boundary

The following procedure divides the given video into shots (grouping of similar frames).

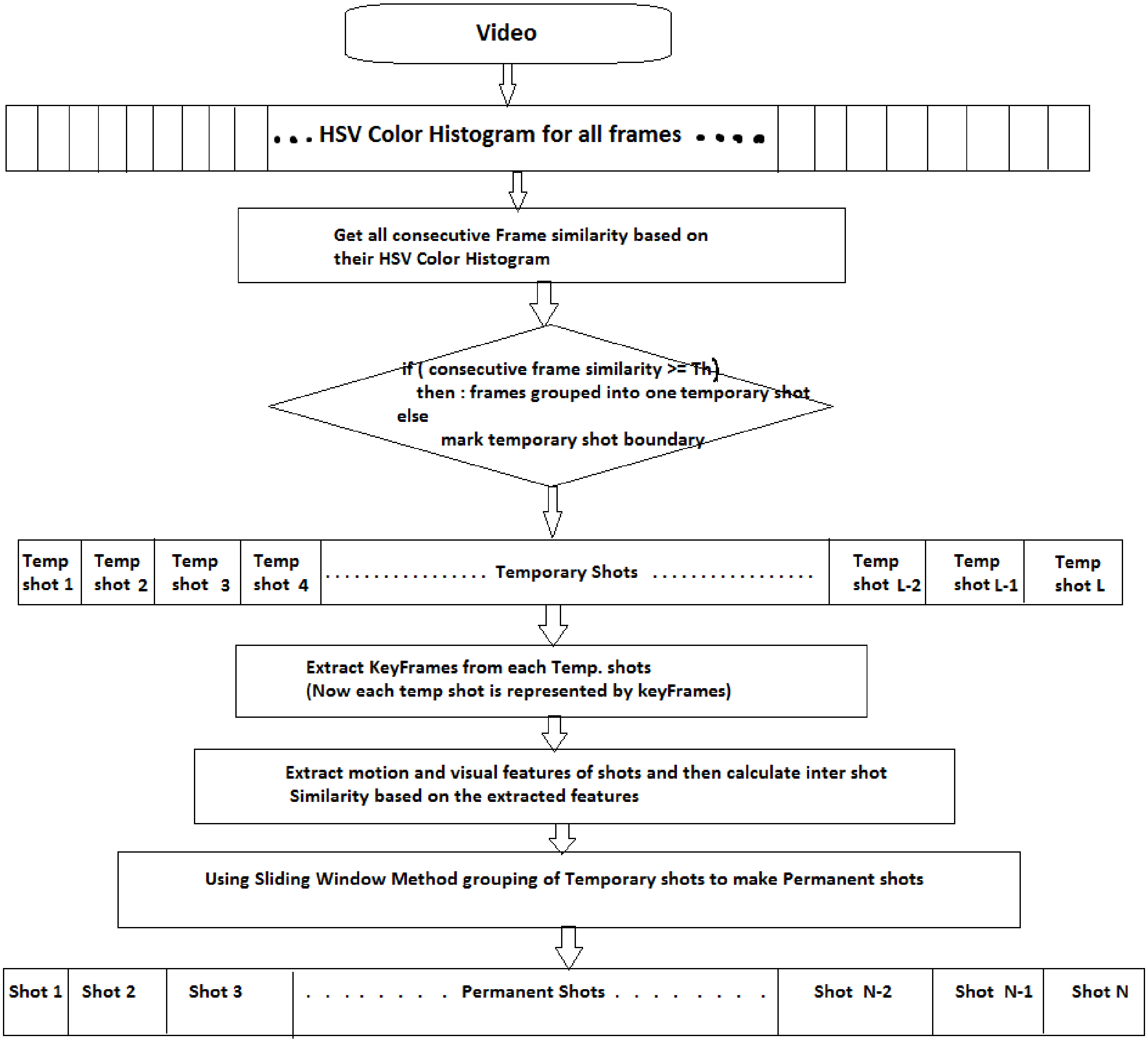

Video
. . . HSV Color Histogram for all frames . . . .
Get all consecutive Frame similarity based on their HSV Color Histogram
if (consecutive frame similarity >= Th)
then : frames grouped into one temporary shot
else
mark temporary shot boundary
Temp shot 1
Temp shot 2
Temp shot 3
Temp shot 4
. . . . Temporary Shots . . . .
Temp shot L-2
Temp shot L-1
Temp shot L
Extract KeyFrames from each Temp. shots
(Now each temp shot is represented by keyFrames)
Extract motion and visual features of shots and then calculate inter shot Similarity based on the extracted features
Using Sliding Window Method grouping of Temporary shots to make Permanent shots
Shot 1
Shot 2
Shot 3
. . . . Permanent Shots . . . .
Shot N-2
Shot N-1
Shot N

Figure 3.6 Defining Shot Boundaries

3.2.2 Defining Scene boundary

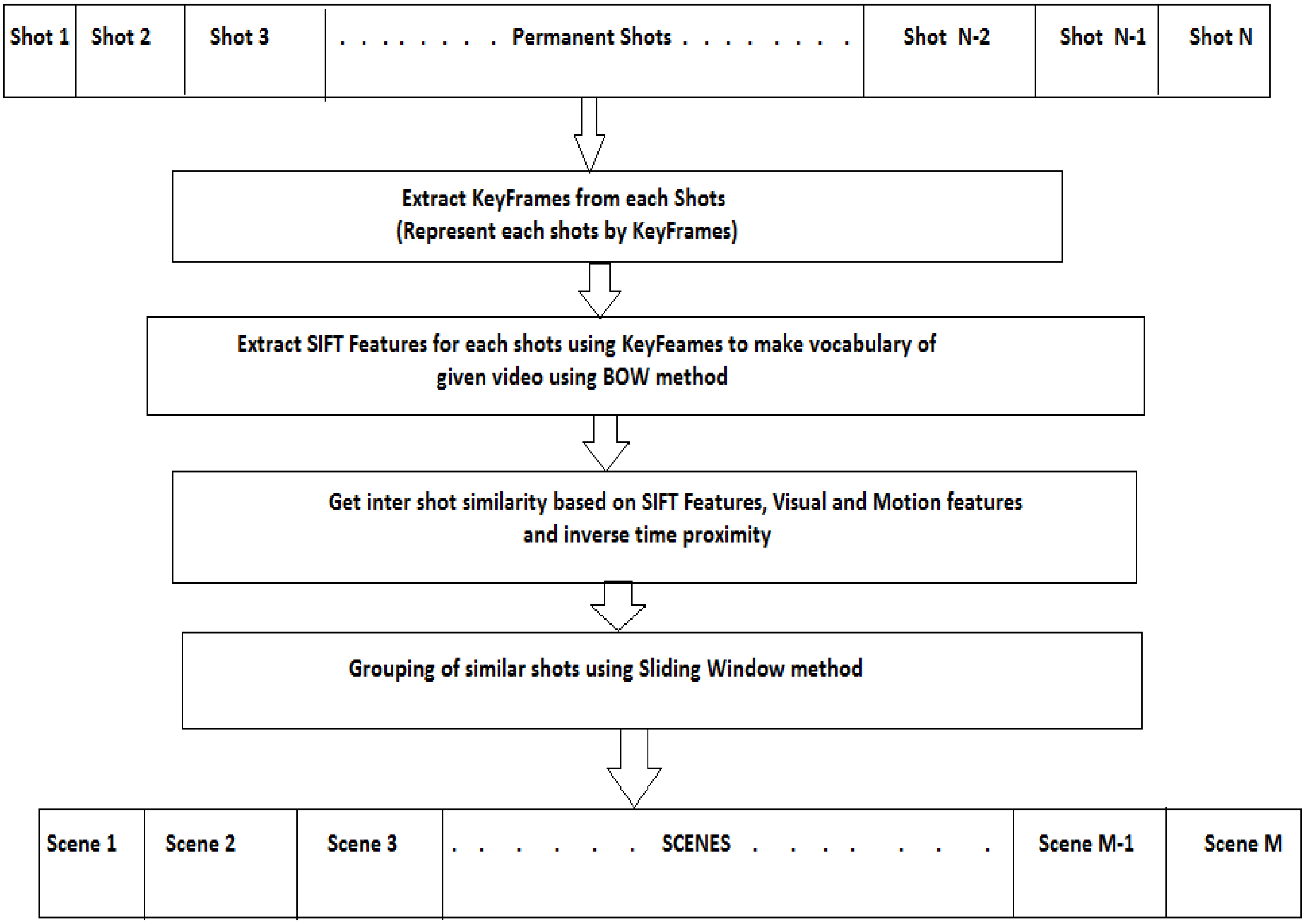

Shot 1
Shot 2
Shot 3
. . . . Permanent Shots . . . .
Shot N-2
Shot N-1
Shot N
Extract KeyFrames from each Shots
(Represent each shots by KeyFrames)
Extract SIFT Features for each shots using KeyFeames to make vocabulary of
given video using BOW method
Get inter shot similarity based on SIFT Features, Visual and Motion features
and inverse time proximity
Grouping of similar shots using Sliding Window method
Scene 1
Scene 2
Scene 3
. . . . SCENES . . . .
Scene M-1
Scene M

Figure 3.7 Defining Scene Boundaries

3.3 Defining Shot Boundary

First of all the video is loaded into the memory and then extract all frames of video. These frames are in RGB format. As I worked with HSV color map, so I converted each RGB frame into the HSV image and then taken 256 (H=8, S=8, V=4) bins Histogram of HSV image.

Now the histograms define the frames and further computation can be done using these histogram. I have taken similarity between consecutive frames as[1][4][7]:

$$ColSim(i,j) = \sum_{b\in bins}^{n} \min(H_i(b), H_j(b)) \tag{3.3}$$

Where H_i and H_j are the HSV color histogram with 256 bins each and jth frame is the next frame of ith, means j=i+1. Color similarity is define in the range of 0 and 1, $ColSim(i, j) \in [0, 1]$.

A threshold value is considered and then traverse the similarity table which has all consecutive frame similarity measure. If the similarity between two consecutive frame is above and equal to that threshold then that particular pair of frame is share the previous shot and if this measure is less than threshold then a new shot is starts[1][4].

$$If \quad ColSim(i, i+1) \geq Threshold$$

$$then \ \ (i+1) \in mth\ shot$$

$$Else \quad (i+1) \in (m+1)th\ shot$$

It means if i^{th} frame is in m^{th} shot, and the similarity measure between i^{th} and $i{+}1^{th}$ frames is greater than or equal to decided threshold value then $i{+}1^{th}$ frame is also belongs to m^{th} shot. And if the similarity measure between i^{th} and $i{+}1^{th}$ frames is less than decided threshold value then $i{+}1^{th}$ frame is the first frame of next $m{+}1^{th}$ shot. I have variable Threshold value for experimental view that is in range of [0.8, 0.9]. this process cuts the video into temporary shots.

This above process virtually cuts whole video into some finite number of shots[6].

$$movie = [S1, S2, S3, S4 \ldots \ldots \ldots Sm] \text{ where } i = \{1,2,3,4,5.....m\} \quad and$$

$$S_i = [f1, f2, f3, f4 \ldots \ldots \ldots fn] \text{ where } = \{1,2,3,4,5.....n\}$$

The above process define boundaries for the temporary shots. but if there are multiple consecutive shots which have very less frames, means the video is splits into shots which are have a few frames or of very short duration then we can merge that shot to previous or next shot based on the HSV Color Histogram simalarity index.

```
if   ( shot(end_frame – start_frame) < frame_rate_of_the_video )
     then     SSIM1 = SSIM( mid_previous_shot , mid_current_shot)
              SSIM2 = SSIM( mid_next_shot , mid_current_shot)
              if   (SSIM1 > SSIM2 )
```

```
            then  MERGE(previous_shot , current_shot )
        else MERGE(next_shot , current_shot )
        end
else   Continue
end
```

This process reduces the number of shots and gives the best shot with high information about the particular shot. Now finally we get the temporary shots.

3.3.1 Key Frame Extraction

As earlier state that if we consider all frames of the video for video processing then system use more computation power, primary memory and it will take huge time to process all of frame. So to reduce the computation time and memory we reduce some redundancy of the similar frame and select only those frame with are less similar. So by doing this only key frames can represents the whole shot[12][13].

The algorithm for selection of key frames from each shot is[1]:

Step 1: select middle frame of a shot is as first key frame

$$K_z \longleftarrow \left\{ f^{\left\lfloor \frac{(firstframe+lastframe)}{2} \right\rfloor} \right\}$$

```
Step 2:    for      i=first frame :  last frame
                If     $\max(ColSim(i,k)) < Threshold$   ,        $\forall f^k \in K_z$
                     then   $K_z \longleftarrow K_z \cup \{f^i\}$
        end
```

here in above algorithm all most similar caries redundancy for the shot information so we selects only those shots which are less similar then some threshold. In my experiments I have used [0.85-0.95] as the threshold value. Using this method we can easily represents the shot by these key frames only. In above algorithm for each shot initially I selected mid frame as first key frame for the particular shot. After that for loop considers all frames from first frame of shot to last frame of the shot for color similarity comparison with all selected key frames. If the chosen frame is less similar then decided threshold then that frame will merge into the set of key frame.

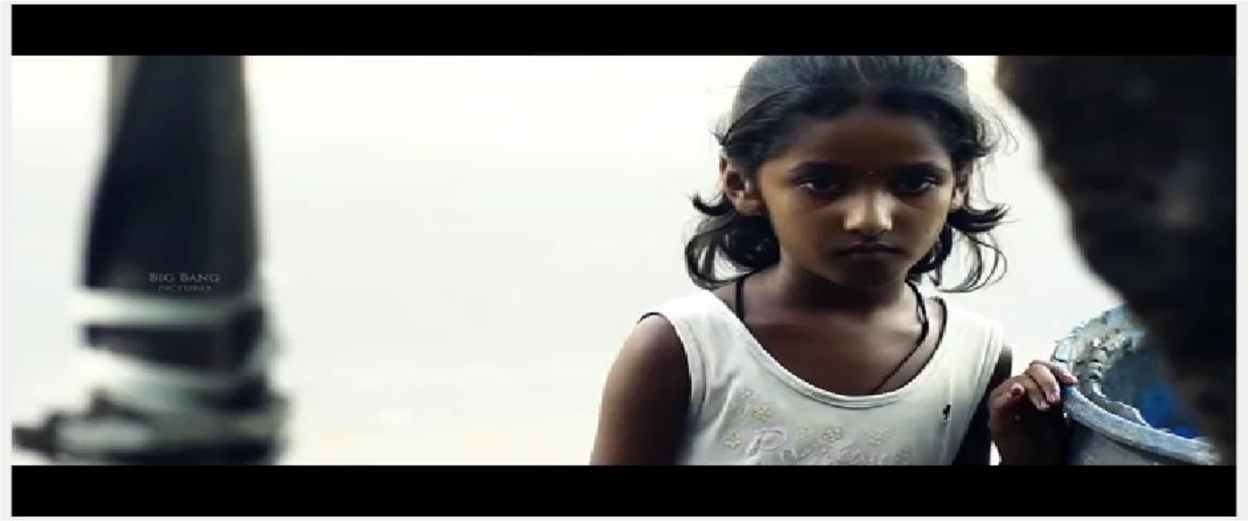


Figure 3.8 One Frame selected as key frame for a shot

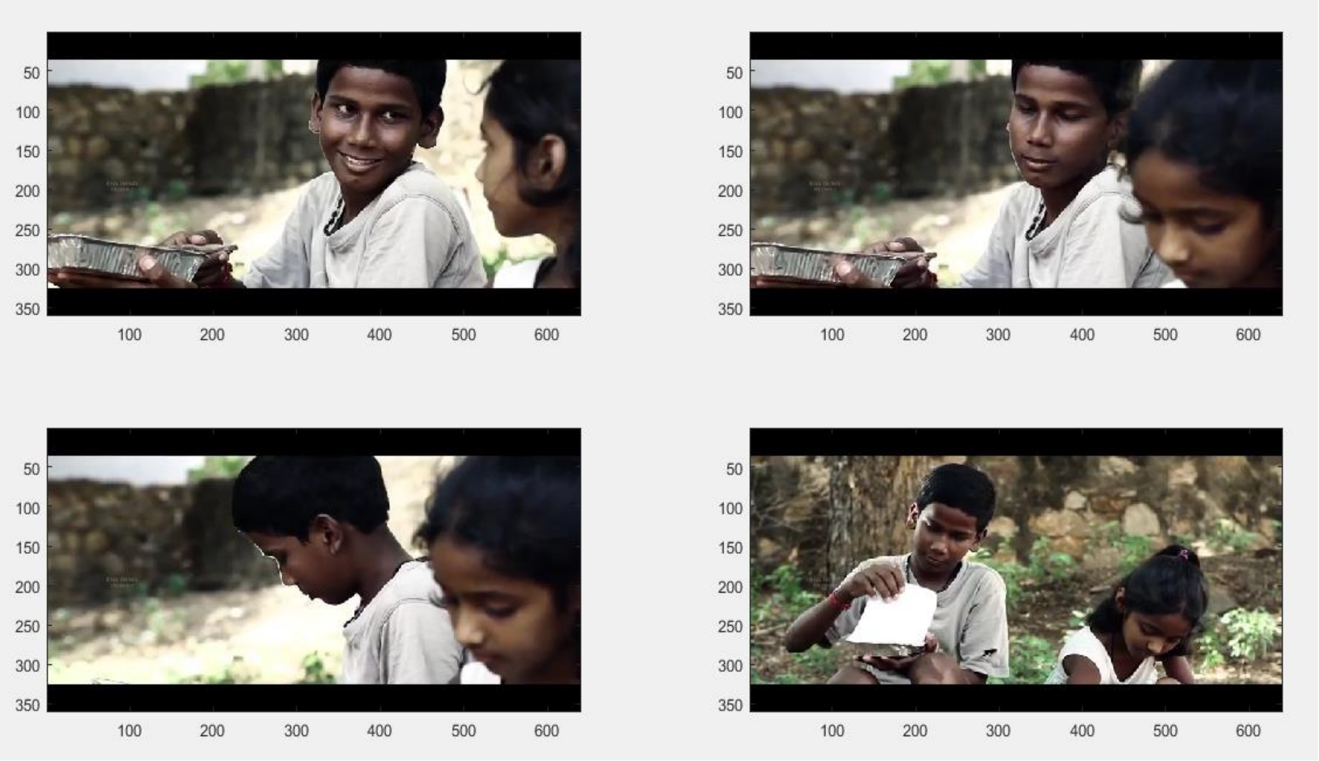


Figure 3.9 Multiple key frames selected in a shot due to motion or some chasing actions

3.3.2 Defining shot similarity graph for temporary shots

As earlier stated that these shots are temporary shots which are defined based on the color similarity only. There may be chances of some disturbed frame in the video that cause to split the one single shot into two shots. This disturbance occurs by some blurred frame or opaque (black) frame. Due to these abnormal frames initially we may get bad shots that can cause to produce different end results. So to make them correct we have to merge the shots which are temporarily closed and have higher similarity in between.

So now we find the similarity between all shot pairs based on 2 or more features vectors. I have used Visual similarity and Motion features of the shots. Visual similarity is based on the color similarity of the frames. Below equation shows the similarity between two shots *i* and *j* [1][5].

$$ShotSim(i,j) = \alpha . VisSim(i,j) + \beta . MotSim(i,j) \qquad 3.4$$

In above equation α and β are the weights are given to each shot features, in our method I have taken the value of α and β as α + β = 1, (α = 0.75 and β=0.25). I my experiments these values provides satisfactory results.

3.3.2.1 Visual Similarity between temporary shot pairs

Visual similarity between the shots are defined by the HSV color histogram similarity, key frames of any two shots are taken and then find the HSV color histogram similarity using below equation[1][5] :

$$ColSim(i,j) = \sum_{b \in bins}^{n} \min(H_i(b), H_j(b)) \qquad 3.5$$

Now check the best possible pair of key frames of the two shots which has maximum color similarity and that similarity defines as shot similarity between those two shots[1][5][7].

$$VisSim(i,j) = max_{a \in K_i \ and \ b \in K_j}(ColSim(a,b)) \qquad 3.6$$

Where *i and j* are two shots and *a and b* are key frames form shot *i and j* respectively. The visual similarity index range in [0 - 1].

3.3.2.2 Motion similarity between temporary shot pairs

Motion is another good feature of shots which shows the motion of the character in the video. In action and chasing scenes visual content of the shot is changing very frequently, so if we consider the visual similarity only then the action and chasing scenes are divided into very large number of the shots with a few number of frames. For action and chasing videos the motion content of features of the shot is more important than the steady dialogue of the shot for the scene segmentation.

Motion content of a shot is calculated using below equation, the motion content or feature is also estimate using color similarities of the Key frames of the shot[1][5].

$$avg = \frac{1}{b-a} \sum_{f=a}^{b-1} (1 - ColSim(f, (f+1))) \qquad 3.7$$

$$Mot_i = \sqrt{\frac{1}{b-a} \sum_{f=a}^{b-1} \left(1 - ColSim\left(f, (f+1)\right) - avg\right).\hat{}2} \qquad 3.8$$

Where *Mot*$_i$ ∈ [0, 1] *a* and *b* are first key frame and last key frame of the shot respectively, *Mot*$_i$ is motion content of the i^{th} shot. Motion content of each shot is normalized with their total number of the frames. This Motion feature is nothing but deviation of the color combination within shot. Sometimes motion due to camera like pan or tilt in a dialogue scene, misleads the motion content in the scene, but for steady camera motion is very less so overall estimation is not affected much by the motion outliers.

Motion similarity between any 2 shots is given below[1][5]:

$$MotSim(i, j) = \frac{2 \times \min(Mot_i,\ Mot_j)}{Mot_i + Mot_j} \qquad 3.9$$

Where *MotSim(i, j)* is the similarity between ith and jth shot. If the motion features of two shots then Motion Similarity of those two shots are higher.

3.3.3 Grouping of temporary shots to make permanent shot using sliding window method

In sliding window method a fixed number of slides or shots are compared with each other for deciding that the particular shot is the part of previous permanent shot or that shot is start of the next permanent shot. Within that decided window shots, if a shot has similarity index greater than or equal to some decided threshold then that shot is considered the part of the previous permanent shot. Opposite to this if any particular shot is not similar up to threshold value to any of the shots presents in that window, then that shot starts new permanent shot.

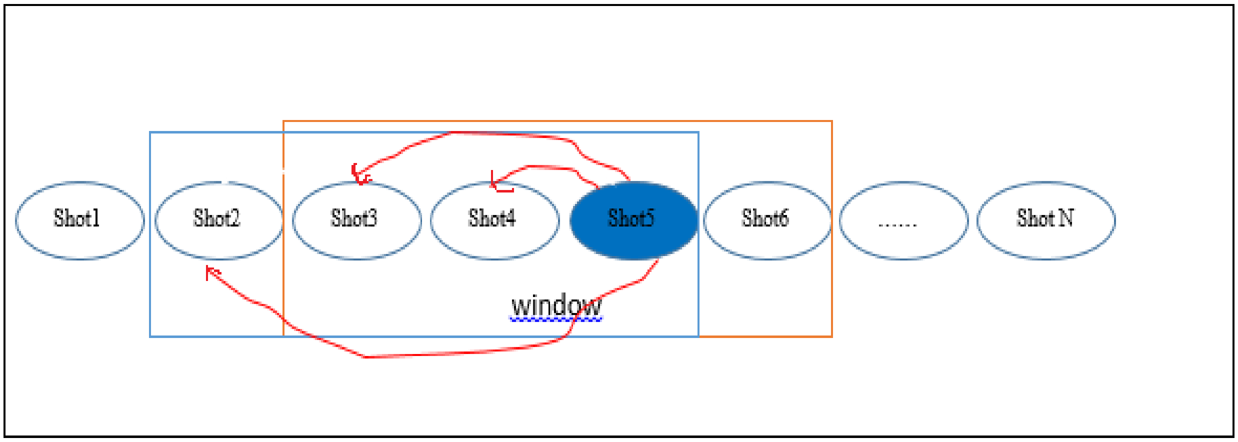


Figure 3.10 (Sliding window): This figure shows that shot5 is similar to any of the shot2 or shot3 or shot4, So shot5 is considered as part of the previous permanent shot. Window is slides to check for next shot6

In my experiments I have fixed the window size of four. Last temporary shot of the window compared with rest of the window shots, if it found any of the shot which is higher similarity than threshold then the last shot is merge to the previous permanent shot and the window is move to next shot[5].

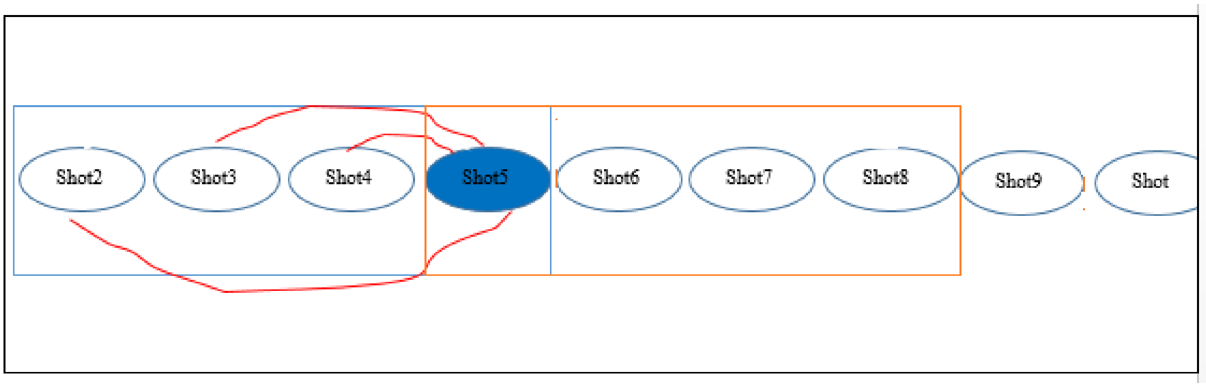


Figure 3.11 (sliding window): This figure shows that shot5 is less similar than threshold value so from shot5 a new permanent shot is begins

If the any temporary shot is not similar to the rest of the window shot then window slides and the staring shot of the window is previously compared shot and window length is to next 3 shots.

3.4 Scene detection using Sliding window based method

Shots have limited information about the action playing in the video. So to find the some semantic meaning of the character who playing the action, things and palace where this action is happening or background we have to group the similar shots to make them a scene. So more specifically a scene is grouping of similar shots which are temporarily closed to each other and share either the same background or character who is acting.

To find the similarity between the shots we are considering 3 features of the video frames. These features are visual similarity, motion features similarity and inverse time proximity. Based on the combination of above three features we define the similarity index of the shots.

As earlier stated that we can't process all the frames of shots so we have to extract some key frames from each shot, so that the whole shot is represented by the key frames and all further processing and computational work is done on those selected key frames only. For key frames extraction we will use the same technique which has earlier described in 3.2.1 section of this chapter.

Now let's say movie is divided into shots and movie, and each shot is represented by the key frames which is shown in equation[1][6]

$$S_i = [kf1,\ kf2,\ kf3,\ kf4 \dots\dots\dots kfn]$$

Where, i = {I,2,3,4,………… m } are total number of shots. and *Kf1, kf2*………… *kfn* are key frames of the *ith* shot.

3.4.1 Visual Similarity between shot pairs

Visual similarity between any pair of shots can be calculated using equations 3.5 and 3.6, which are discussed earlier in section 3.3.2.1 of this chapter.

3.4.2 Motion similarity between shot pairs:

Motion similarity between any pair of shots are calculated using equations 3.7, 3.8 and 3.9, which are discussed earlier in section 3.3.2.2 of this chapter.

3.4.3 SIFT Similarity between shot pairs:

As discussed in section 3.1.7 detector method of the vl_sift function of vl feat library uses gassian space for defining the keypoints on image/frame. These KeyPoints are described by a geometric*frame* of four parameters: the keypoint center coordinates *x* and *y*, its *scale* (the radius of the region), and its *orientation* (an angle expressed in radians). In my experimates I have used default value of these parameters which are defined by vl feat library. Where peak threshold is 3.5 and edge threshold is 0, Number of levels per octave are 3, first octave index is -1. Etc.

For defining the SIFT similarity between shot pairs I have used Bag Of Words (BOW) model[6]. In this technique fist define the vocabulary or dictionary of features of the video and then find the presence of similar features in shot pair for calculating the similarity between shots.

For creating the features dictionary (BOW) of the movie consider all key frames of the all shots and then find the SIFT feature of each key frames of each shot[8][9]. Because of considering all key frames of all shots there may be multiple redudent feature are added to the dictionary that's why we use k means clustering for reducing the redudend features and create unique vocabulary or dictionary table for the video. The cluseter center are known as features or words of the dictionary, lets say there are K total words in the dictionary.

Now each shot is represented in the form of 1 x K histogram. Where shot(i,j) represents that the ith shot has jth word of the dictionary. Using this convention, a shot is represented by[6][7]

$$S_i = n1, n2, n3, \dots \dots \dots \dots nj \dots \dots \dots \dots .nK \qquad 3.10$$

where nj gives the frequency of jth visual word divided by total number of unique visual words present in ith shot, and K is the size of the vocabulary. With this representation of the shot, we use intersection distance to find the similarity between pair of shots intersection distance between ith and jth shot is given by

$$SIFT\_SIM_{i,j} = \sum_{k=1}^{k=K} \min(n_{i\,k}\,, n_{j\,k}) \qquad 3.11$$

Here n_{ik} gives the normalized frequency of kth visual word in ith shot. This above equation gives the SIFT Similarity between any pairs of shots.

3.4.4 Inverse time proximity between shot pairs:

My thesis work is on temporal video scene segmentation, so I have to consider the timing of the shot occurs in the video. If two shots are very similar to each other but the position of those shots are very far to each other in respect of video length then those shots can’t be grouped within a scene. The scene can formed by the similar shots which are closed to each other in respect of the time. For this purpose I have used the inverse time proximity which normalize the similarity of shot pairs according to their position in the video. The following exponential weighting function decreased the similarity index of two shot with increasing time distance between them[1][5][6].

$$W(i,j) = e^{-\frac{1}{d}\cdot\left|\frac{m_i-m_j}{\sigma}\right|^2} \qquad 3.12$$

Where, W(i, j) is the weight for similarity of two shot i and j, and this weight is decays as time between them increases. In above equation σ is the standard deviation of shot durations in the entire video, and m_i, m_j are the middle frame time of the show shots i and j. d is the constant factor which influence the final scenes of the video. the value of d is 20 for my experiments.

3.4.5 Construction of Shot Similarity Graph (SSG):

Shot similarity graph (SSG) is constructed based on the shots and the weighted similarity between them. A graph G(V,E) is consists of two sets one is set of nodes and other is set of edges between the nodes. The edge between any pairs of shot have some weight. In our graph shots are node of the graph and the weighted similarity is edges between these shots[1][2][5][6].

Shot similarity graph is defined as:

$$SSG(i,j) = W(i,j) \times ShotSim(i,j) \qquad 3.13$$

Where in this equation W(i, j) is the inverse time proximity weight which is defined in equation [3.13] and the ShotSim is defined as below [1]

$$ShotSim(i,j) = \alpha\,.VisSim(i,j) + \beta.MotSim(i,j) + \gamma.SIFT\_Sim(i,j) \qquad 3.14$$

Where in above equation α and β are weights given to each similarity features. This weight is given such that ShotSim index is in range of [0, 1]. In our experiment α + β + γ = 1, where α = 0.5, β = 0.25 and $\gamma = 0.25$.

3.4.6 Defining scene boundaries using sliding window method:

Scene are the grouping of the similar shots, in out proposed sliding window based method we will compares multiple shots which are lied in the window. If the shot is similar to any of the window shots then that shot will merge to previous running scene otherwise where the shot is mismatch to other window shots a new scene is begin.

In our method we have chosen a window of 5 shots. let’s say for shot A, B , C, D and E, if shot A,B and C are part of the same scene and now we are checking for shot D. first D is compared with shot A, if similarity between these two shot is greater than decided threshold then Shot D is also part of the previous running scene and window moves to next shot E and checks for their similarity against previously selected shots in that scene. Now suppose we talk about opposite condition if Shot A and Shot D are not similar than that of threshold value, then we check similarity of shot D with shots B and shot C. if any of them satisfy the similarity greater than threshold value then Shot D will also be the part of previous scene. But if any of shot A, B and C not satisfy the threshold value then previous scene boundary is defined at shot C and from shot D a new scene is begins.

```
$Scene_x$ ={shotA,shotB,shotC }
Check for shot D
if (ShotSim(A,D)>= th OR ShotSim(B,D)>= th OR ShotSim(C,D)>= th OR ShotSim(C,E)>= th )
    then $Scene_x \leftarrow Scene_x \cup$ shot D
else
    $Scene_y \leftarrow$ shot D   // start of new scene y.
end
```

For my experiment purposes I have selected the threshold value is [0.09 – 0.13] according to number of scenes. This threshold is varies with different kind of video like for home video this is set to 0.10 and for chasing and action movies this will set to 0.15.

This window based approach solve one of the major issue of the scene segmentation that is in a single scene there may be two character or two background with different color content switching frequently and due to this the whole scene is divided into the multiple shots. Where shots next to adjacent shot is very much similar to each other. So this problem is solved by the window based approach where up to 3 previous shot are compared and if any of them found similar to comparing shot then that shot is also marge into the current running scene. This above process decides the boundaries for the valid scenes.

3.5 Selection of sliding window

In [5] author discuss a sliding window method for defining the scene boundaries. The size of sliding window is 7, in which initially 3 shots are considered as part of a scene lets say scene *m*.

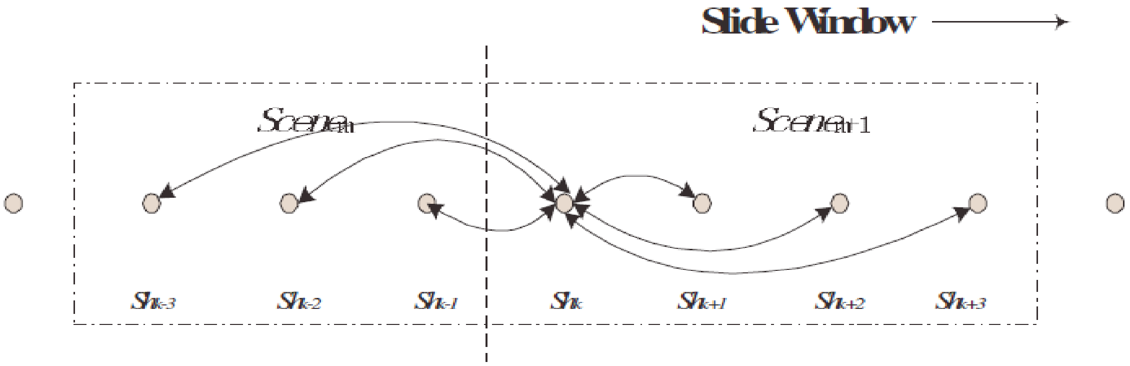
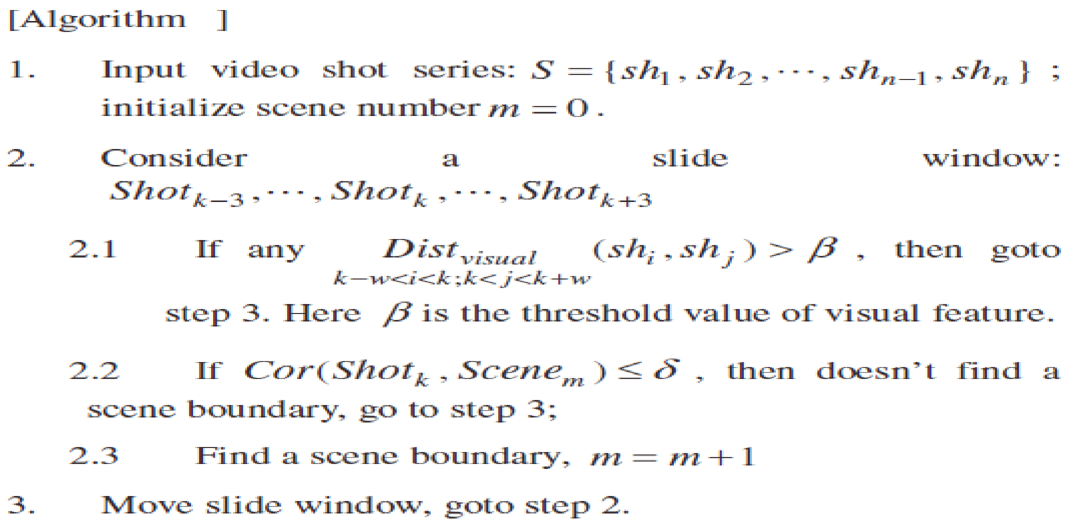


Figure 3.12 Sliding window

now he will check for shot 4 whether that shot is part of the scene m or shot 4 is the starting shot of the next m+1 scene. For this author discussed the following algorithm :



```
[Algorithm ]
1.    Input video shot series: $S = \{sh_1, sh_2, \cdots, sh_{n-1}, sh_n\}$ ;
      initialize scene number $m = 0$ .
2.    Consider a slide window:
      $Shot_{k-3}, \cdots, Shot_k, \cdots, Shot_{k+3}$
    2.1    If any $\underset{k-w<i<k;k<j<k+w}{Dist_{visual}} (sh_i, sh_j) > \beta$ , then goto
           step 3. Here $\beta$ is the threshold value of visual feature.
    2.2    If $Cor(Shot_k, Scene_m) \le \delta$ , then doesn't find a
           scene boundary, go to step 3;
    2.3    Find a scene boundary, $m = m + 1$
3.    Move slide window, goto step 2.
```

In this *Cor*(*Shot k* , *Scenem*) defined as follow :

$$Cor(Shot\_k\ ,Scenem\ ) = \frac{vright}{vleft}$$

Where, $vleft = Dist(shot_{k-3}\ , shot_k) + Dist(shot_{k-2}\ , shot_k) + Dist(shot_{k-1}\ , shot_k)$

$vright = Dist(shot_{k+3}\ , shot_k) + Dist(shot_{k+2}\ , shot_k) + Dist(shot_{k+1}\ , shot_k)$

if $Cor(Shot\ k\ , Scenem\ ) <= \delta$, there δ is a threshold value.

Then: $shot_k$ merge into $scene_m$; else $shot_{k-1}$ is the last shot of $scene_m$ and $shot_k$ is the first shot of $Scene_{m+1}$.

In above alogithm evrytime when scene boundary is defined then next three shots are considered as part of the same scene, this is done without any checking of there visual and any other features of the shot. To correct this gap I have used my own sliding window method[5][10].

My sliding window size dynamically changes and its maximum size is 5. In which 4th shot of window is to be checked for a part of current running scene lets say scene m. for this 4th shot is compared with previous 3 shots, if any of the comparision is greater than defined threshold then 4th shot will be part of the mth scene. If any of amoung three comparision is less then defined threshold then 3rd shot of window will compares with 5th shot of window if similarity between these two shots are higher than defined threshold then also 4th shot will be part of the mth scene.

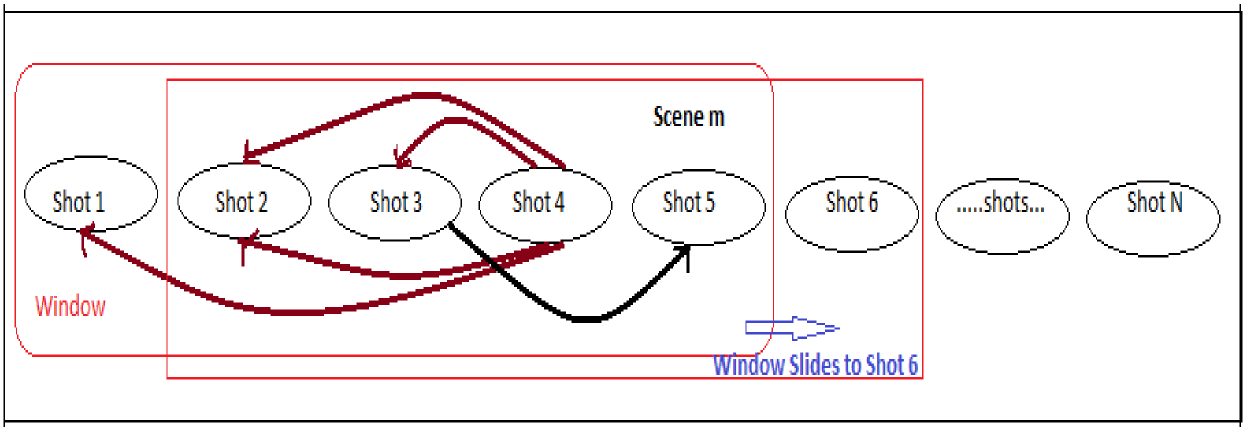


Figure 3.13 Shot 4 is the part of scene m

Now lets suppose 4th shot is not part of the mth scene then boundary for mth scene is defined at 3rd shot of current window.Now window srinks to size 3, where 1st shot of window is the starting shot of new (m+1)th scene. Now checking for 2nd shot of window for a part of the (m+1)th scene. This is done by comparing 1st and 2nd shots of the window and for next next shots window size is increases 1 by 1 and it increaded upto length 5. Then for next shots it will compares only latest 3 shots of current running scene. Algorithm for my sliding window method.

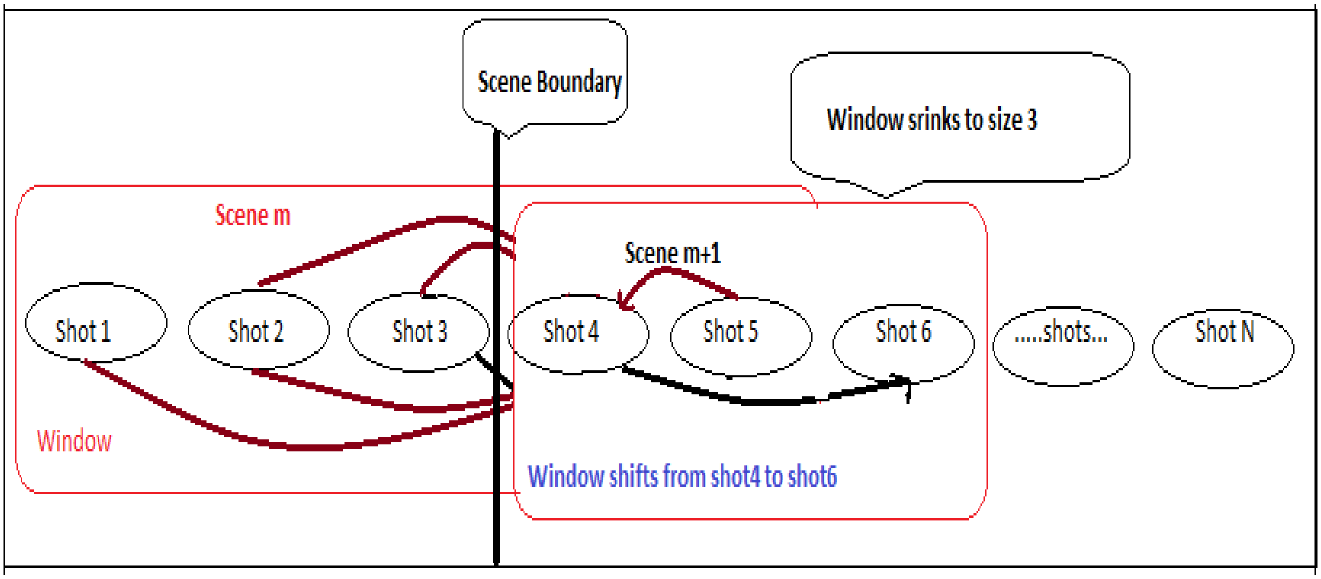


Figure 3.14 Shot 4 is not part of scene m, so new sliding window created and shot 4 is in scene (m+1) , now checking for shot5

Algorithm:

```
Input : shot1, shot2,shot3……………shotN (Array of shots)
Output : scene1,scene2………………sceneM (array of scenes)
K=1, M=1 // Here,  K is for shots and M is for Scenes
While (K <= N-1)
Step 1 :  Scene(M) ⟵ {Scene_M }∪ shot(K)
      K=K+1
Step2 :  len = length( Scene(M) )
       If ( len > 3)
          X=len-3
       Else X=1
      $Sim = \max(\ ShotSim(\ shot(K),\ shot(i)\ )\ )\ where\ ,\ i \in Scene\_M(X, X+1, \dots\dots len)$
    If  ( $Sim \geq Threshold1$  OR ( $ShotSim(\ shot(K-1),\ shot(K+1)\ )\ ) \geq Threshold2$ )
       Then :  Go to step1
     Else
            M=M+1
            Go to step 1.
```

4 EXPERIMENTS AND RESULTS

For the experiment purposes we have taken 6 videos of different kinds. Different videos have different character who is acting the video some of video are of animals and other are of humans. These video are taken from youtube.com, and the following are the details for videos on which we tested my proposed work. Most of the above videos are of action videos. we have done experiments with 2 models

1. Without using SIFT Features.
2. Using SIFT features.

Table 1: Ground truth for the dataset (Video Details)

Sr. no.	Name of video	Total scenes / ground truth	Playing character	Duration (mints)	Kind of video	Total no. of frames
1	Wildlife	07	Animals	0:30	Action and chasing	901
2	Unbelievable lucky people of 2016	34	Humans, animals	2:47	Action and chasing	5032
3	Union is strength (advt)	03	animals	1:19	Action (advertisement)	2375
4	Jana gana mana video	35	humans	11:32	Simple action video	17308
5	Magic short video	14	humans	7:53	Simple action video	14186
6	song	31	humans	7:59	Simple action video	11494
7	Comedy movie	32	human	19:05	Simple action	28629
8	Best Advt	40	Human, dog	3:05	Simple action	4631

Table 1 shows videos with name of video, total scenes, playing character, duration of video, kind of video and total number of frames extracted.

4.1 Experiments and Results

Our proposed work is divided into two major parts:

1. Shot boundary detection of the videos.
2. Scene boundary detection of the videos based on the detected shot.

Following is the experimental status of the shot segmentation and scene segmentation of the following videos based on the 2 strategies

1. Visual features and Motion features based on the HSV color Histogram.
2. SIFT features , Visual features and motion features based on the HSV color Histogram.

Following is the detail table for shot segmentation and scene segmentation for using visual features and motion features.

Table 2: Results using Visual and Motion Similarity

Sr. no	Video name	Detected scene	Correct scene	False negative	False positive	Extracted Shot	Recall (CS/GT)	Precision (CS/DS)
1	Wildlife	06	06	01	0	08	86%	100%
2	Unbelievable lucky people of 2016	24	21	10	03	56	62%	88%
3	Union is strength (advt)	06	02	01	04	19	66%	33%
4	Jana gana mana video	39	29	06	10	89	83%	74%
5	Magic short video	19	13	01	06	42	93%	68%
6	Song	20	18	11	02	64	58%	90%
7	Comedy movie	38	25	07	13	89	78%	66%
8	Best advt	25	21	19	04	68	53%	84%

Table 2 shows Video name, Detected scene(DS), Correct scene (CS), False negative, False positive, Extracted Shot, recall (CS/GT), precision (CS/DS).Where CS – Correct Scenes, GT- Ground truth and DS – Detected Scenes.

The above results can be evaluated with precision and recall methods[1][5][6].

Where,

$$Recall = \frac{total\ correct\ scenes\ detected}{total\ scenes\ in\ the\ video(Groud\ truth)}$$

and

$$Precision\ = \frac{total\ correct\ scenes\ detected}{total\ detected\ scenes}$$

The following table shows the scene detection using SIFT features , Visual features and motion features based on the HSV color Histogram

Table 3: Results using Visual, Motion and SIFT Similarity

Sr. no	**Video name**	**Detected scene**	**Correct scene**	**False negative**	**False positive**	**Extracted Shot**	**Recall (CS/GT)**	**Precision (CS/DS)**
1	Wildlife	07	07	0	0	08	100%	100%
2	Unbelievable lucky people of 2016	29	28	06	01	41	82%	96%
3	Union is strength (advt)	04	03	00	01	16	100%	75%
4	Jana gana mana video	37	33	02	05	73	94%	89%
5	Magic short video	12	11	03	02	14	79%	92%
6	Song	33	29	02	04	31	94%	88%
7	Comedy movie	38	30	02	08	32	94%	79%
8	Best advt	34	28	06	06	40	70%	82%

Where CS – Correct Scenes, GT- Ground truth and DS – Detected Scenes.

Table 2 shows Video name, Detected scene(DS), Correct scene (CS), False negative, False positive, Extracted Shot, recall (CS/GT), precision (CS/DS).Where CS – Correct Scenes, GT-Ground truth and DS – Detected Scenes.

Now we would like to compare F-score for both methods [1][6]

$$F - Score = \frac{2 \, * \, precision \, * \, recall}{precision + recall}$$

Table 4: Result Comparision between without sift features and with sift features

Sr. no	**Video name**	**F-Score for Visual and Motion similarity**	**F-Score for Visual, Motion and SIFT similarity**
1	Wildlife	92.00	100.00
2	Unbelievable lucky people of 2016	72.74	88.45
3	Union is strength (advt)	44.00	85.71
4	Jana gana mana video	78.24	91.43
5	Magic short video	78.55	85.00
6	Song	70.54	90.90
7	Comedy movie	71.50	85.85
8	Best advt	64.99	75.53
Average		71.57	**87.85**

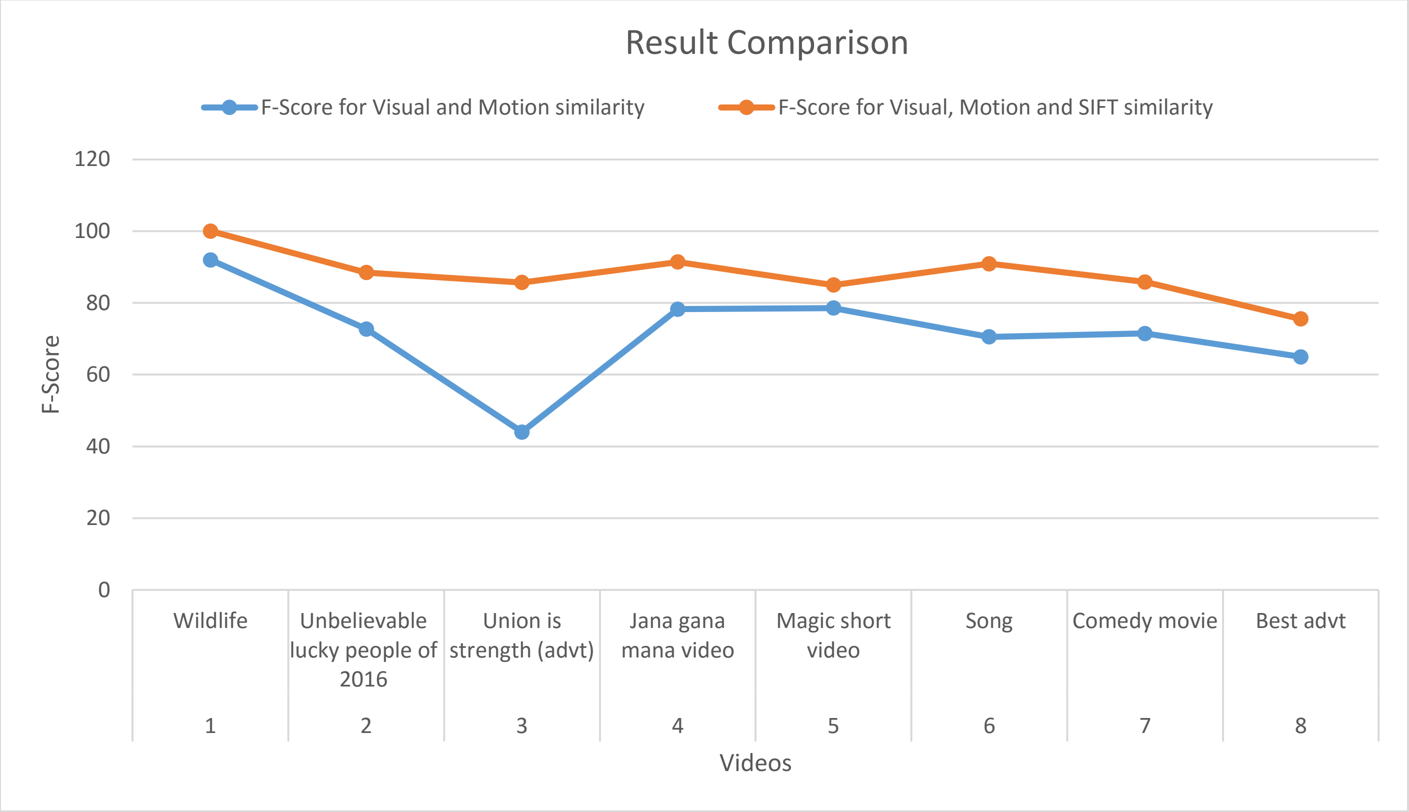


Figure 4.1 Result Comparison between without sift features and with sift features

We performed several experiments on different videos and we calculated f-score values for two different approaches as discussed above

1. F-score value for Visual and Motion similarity.
2. F-Score value for Visual, Motion and SIFT similarity

After analyzing the results from table 4, Average F-score values for visual and motion similarity were 71.57% while average F-Score value for Visual, Motion and SIFT similarity were 87.85%. This shows that F-score values got significantely improved when we considered SIFT features in our approach. So SIFT features are considered essential for video scene segmentation.

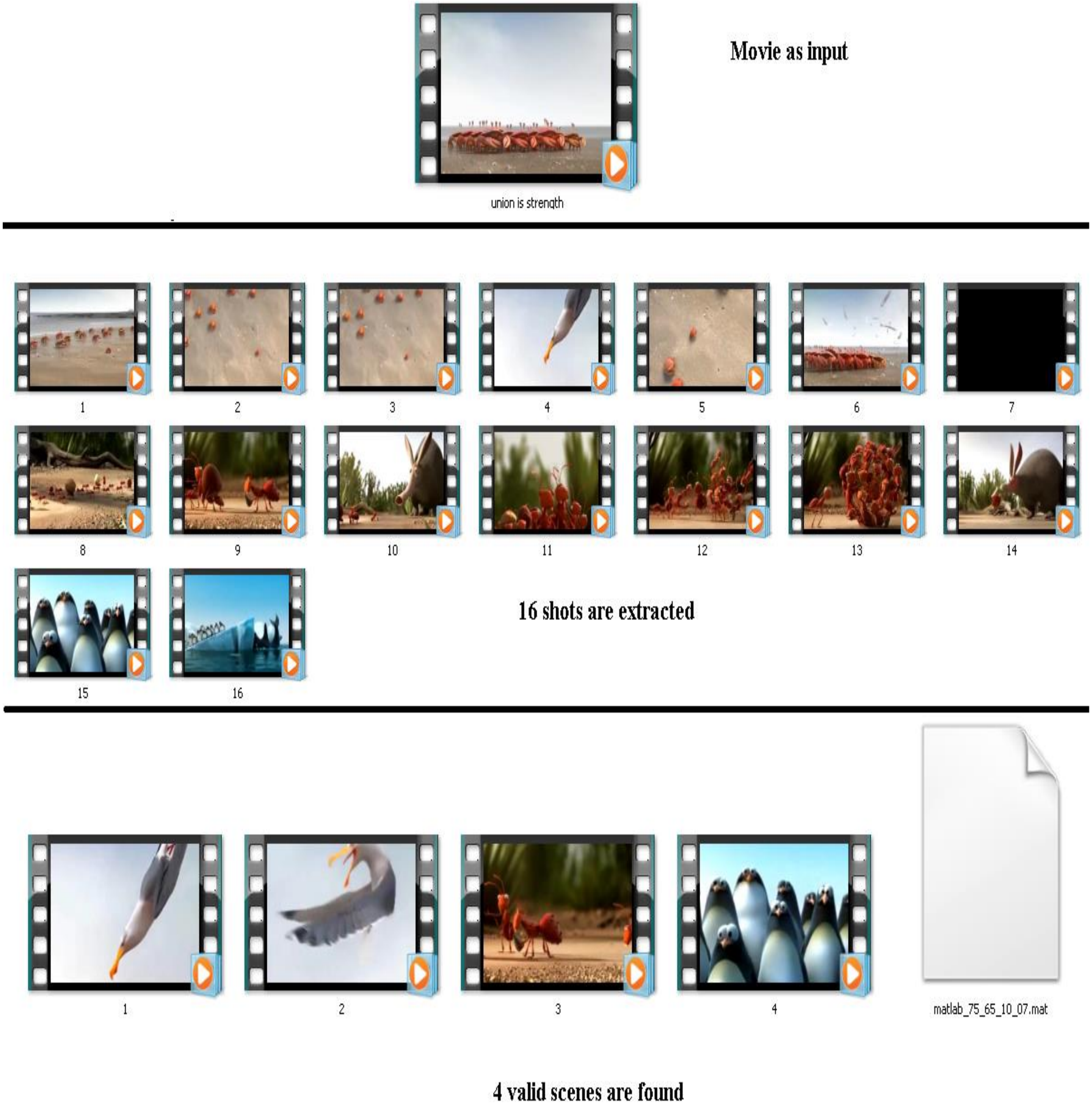


Figure 4.2 Showing results for a movie(union is strength)

4.2 Setting of threshold values

our approach has divided basically in 3 parts, these are:

1. Defining of temporary shots boundary.
2. Defining of permanent shots boundary.
3. Defining of scenes boundary.

Three of above process needs some threshold values to compare and then decides whether the tesing frame is the part of currently running shot or the testing frame is the first frame of next shot. In case of scene boundary defining the threshold value decides for next shot whether that shot is the part of current running scene or first shot of the next scene.

- **Defining of temporary shots boundary:** For defining of temporary shot boundary we have to compare HSV color histogram of consecutive frames. If consecutive frames are similar to some extend or threshold value then we group next frame into the current running temporary shot. For this we have tested the threshold value from 65% to 95% similarity value, but the range 70%-80% similarity value gives good results in our approach. It means if some some consecutive pair of frame have less similarity than 70% then the temporary shot boundary is decided and a new temporary shot is begins. During testing if we set threshold value less than 70%, then temporary shots are less and it combines two scene shots that will produce less valid scenes at the end. But if we set threshold value greater than 85% ,then there are unnecessaty temporary shots are created which takes more computation power to find the further shot similarity and there are chances to produce two parts of a single scene.
- **Defining of permanent shots boundary:** Now we have calculated the similarity between temporary shot pairs based on the visual and motion features. To decide whether two consecutive temporary shots are the part of single shot or not, we compares the similarity of two consecutive temporary shots with some threshold. Using sliding winodow based method transitive shot comparisio is also tested. For this stage we have tested the temporary shot similarity with threshold range from 50% to 80%. But at 60% - 66% threshold value gives good results.

- **Defining of scene boundary:** Now we have calculated the similarity between shots pairs based on visual, motion, SIFT Features and inverse time proximity. To decide wether the next shot is the part of current running scene or first shot of the next scene, we compares the inter shot similarity with some threshold. While using sliding window method for defining the scene boundary some previous shots are also compared with next shot it means transitive shot comparison is also tested. For this purpose we have tested the threshold range from 5% to 30%. The best results are found at 9% - 16% range.

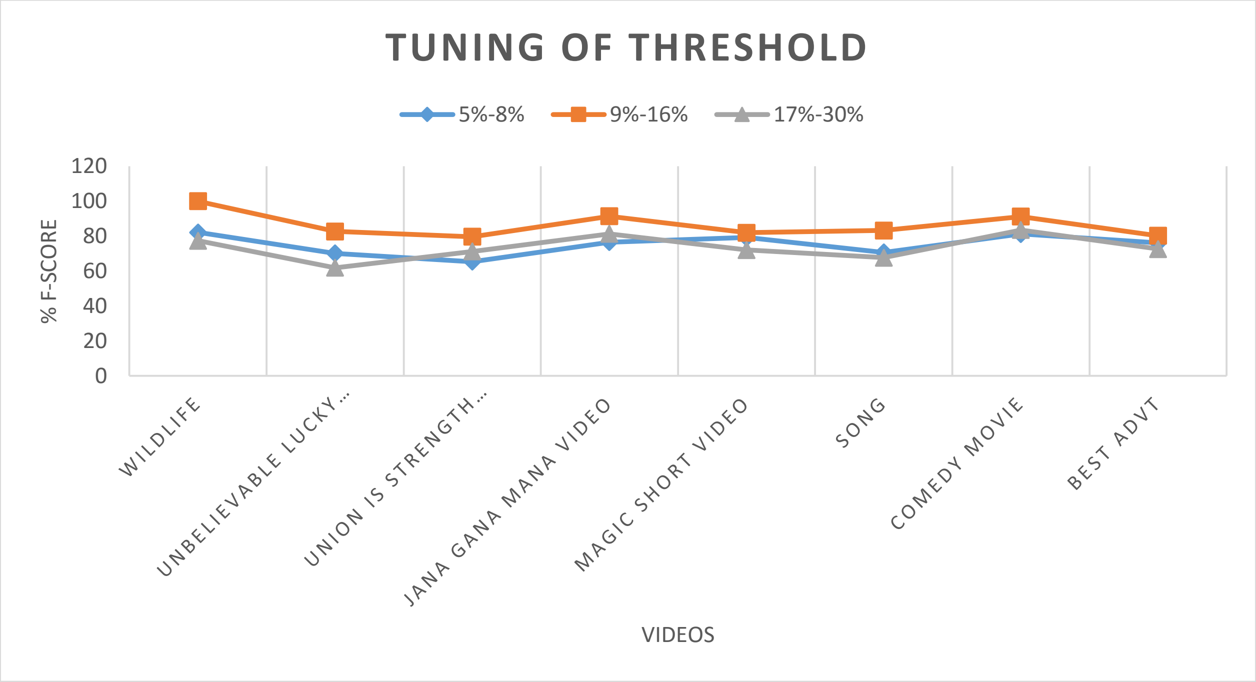


Figure 4.3 Tuning of threshold value(δ) for scene boundaries

In above figure shows that the percentage similarity between 9 to 16 is good for the all videos. this range gives best result. During experiments we have tested for 5% to 30% similarity. In this figure the range is shown by average percentage F-score values. Like fisrt we calculate F-Score value for 5,6,7 and 8 percentage similarity value then average is taken and compared with next ranges average (9% - 16% and 17% - 30%) .

4.3 Comparison of F-score values for various previous approaches

We compared our results with other claimed results from papers[1][2][6], the following figure shows varios average F-score values.

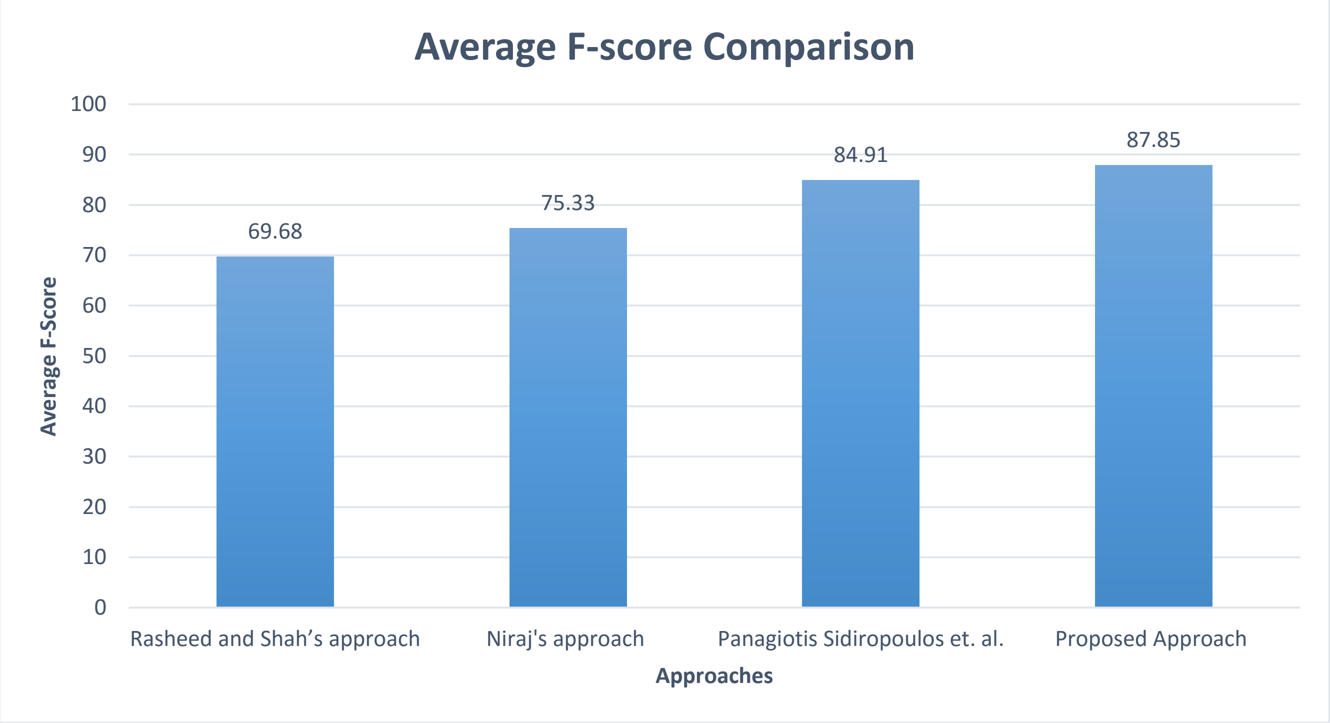


Figure 4.4 Comparing various approaches results

Above comparing figures show that rasheed's and shah's[1] approach gives 69.68%, Niraj's[6] approach gives 75.33%, Panagiotis sidiropoulos et.al. gives 84.91%[2] average F-score values on their own datasets. Our approaches gives better result than others. Our approache gives 87.85% average F-Score for above mensioned video dataset.

5 APPLICATIONS AND ISSUES IN VIDEO SCENE SEGMENTATION

There are numerous field of applications where digital video is acquired, processed and used, such as satellite videos, security surveillance videos, educational videos, civilian or military videos, general identity verification, traffic, criminal justice system, news and sports videos etc. These field uses the videos frequently, so to ease of getting essential information from whole video we have to pre-processes the video which require scene segmentation.

5.1 Applications:

1. **Video scene segmentation is the first step toward automatic video annotation [3]:** In the world of big data there are huge amount of videos are generated on daily basis like security surveillance videos, educational videos, satellite videos, movies, TV shows and many more. For efficient storage and reuse of these videos we have to tag/annotation the video on the basis of semantic meaning and store on the internet, so we can get the right video on demand form the web.
2. **The scene is the first step towards semantic understanding of the entire video:** For video processing for different purposes like enhance the quality, semantic mean of video and analysis of whole video, we have to break the entire video into the suitable scenes and then make some semantic analysis on those scenes.
3. **Video summarization [3]:** video summarization is the open problem in the field of computer vision. In this a video is automatically summarizes based on the story and action happening in the video, to make a good summary of lengthy video, video must divided into different scenes. Each scene is analyzed semantically and with help of sentimental analysis machine generates summary of that scene. At last summary of each scene is combined for whole video.
4. **Video indexing**: video is indexed based on their category like news video, sports video, T.V. shows, documentary video etc. To automatically indexing of video, machine has to analysis the video and then on the basis of their semantics and sentiments meaning videos are indexed. This lead to efficient stores of similar video.

5. **Non- linear navigation of video data:** Breaking up a long video into scenes will allow for non-linear navigation of video data. For example, videos from security cameras and video from satellite are very lengthy videos and if we have to search for something important or odd happened, then this is very difficult task to watch whole video. To make it easy scene segmentation plays very important role to break video into different scenes.
6. **Browsing or reusability of indexed video segments**: if any video is indexed properly then it is very easy to reuse those video. For example scene segmentation will be useful for browsing news programs[10][14].
7. Scene segmentation is also very important in making highlights of sports and trailer of movies.

5.2 Some major challenges in video scene segmentation

1. **Temporal coherence:** temporal scene segmentation of video is more difficult because cost of the videos have similar type of action at different time which are played by same characters. And in between these there may have one or more shots/ scenes. So to make the one scene of similar scene at different time in video is difficult task.
2. **Automatic processing:** As we know all processing is done by machines, it is not so easy to identify and segments the video into perfect scenes, because different videos has some different semantics and characters. Machines has to process on each frames multiple times for desired accuracy.
3. **Scalability issues :** scalability is another major issue, as increasing of size of video it takes more processing time and more resources. Due to scalability issues there possibilities to get false scenes[10].
4. **Interactive segmentation:** for more perfection and desired accuracy, human incorporation is required with automatic interactive segmentation processing tool.
5. High-quality layer separation from a video is a very challenging problem because tightly coupled color, depth, and motion give rise to a large number of variables and significant inexactness in computation.
6. Some videos with special effects of frames like wipe, merging of frame to other frame etc. these type of video are difficult to segment into correct scenes[8].

6 CONCLUSION AND FUTURE WORK

We present a method of partitioning a video, particularly movie, into scenes. We have considered the fact that a scene consists of shots which are semantically related and continuous in time. To achieve that, we have presented a method to compute visual similarity, motion similarity, semantic similarity (SIFT Features) and inverse time proximity between the shots. Once we get these combination of similarities in some suitable ratios, we construct a shot similarity graph (SSG), in which inter shot pair similarities are calculated. Then according to our basic theme "Temporal video scene segmentation" we design a sliding window which is has 6 as maximum length, and merge all similar shtots into a scene. It can be seen from the results that it is important to compute the similarity between the shots not only on the basis of direct similarity but it is important to consider transitive similarity as well. The approach presented here is found to be better than previous approaches. Previous work either includes video segmentation using only color similarity or using only SIFT features, while we have used combination of color and SIFT features so we got better results.

In this thesis we have used only video part of the movie but in future we will consider both the audio and video part of movie. After processing these two parts separately for scene segmentation at the end we will merge both and try to get some better results.

REFERENCES

[1] Rasheed, Zeeshan, and Mubarak Shah. "Detection and representation of scenes in videos." *Multimedia, IEEE Transactions on* 7.6 (2005): 1097-1105.

[2] Sidiropoulos, Panagiotis, et al. "Temporal video segmentation to scenes using high-level audiovisual features." *Circuits and Systems for Video Technology, IEEE Transactions on* 21.8 (2011): 1163-1177.

[3] Zhai, Yun, and Mubarak Shah. "Video scene segmentation using Markov chain Monte Carlo." *Multimedia, IEEE Transactions on* 8.4 (2006): 686-697.

[4] Baraldi, Lorenzo, Costantino Grana, and Rita Cucchiara. "Scene segmentation using temporal clustering for accessing and re-using broadcast video." *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. IEEE, 2015.

[5] Zhao, Li, Shi-Qiang Yang, and Bo Feng. "Video scene detection using slide windows method based on temporal constrain shot similarity." *Proceedings of international conference on Multimedia and Expo*. 2001.

[6] Kumar, Niraj, et al. "Video Scene Segmentation with a Semantic Similarity." *IICAI*. 2011.

[7] Gu, Zhiwei, et al. "EMS: Energy minimization based video scene segmentation." *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007.

[8] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60.2 (2004): 91-110.

[9] David G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision,* Corfu, Greece (September 1999), pp. 1150-1157.

[10] Koprinska, Irena, and Sergio Carrato. "Temporal video segmentation: A survey." *Signal processing: Image communication* 16.5 (2001): 477-500.

[11] Lin, Tong, and Hong-Jiang Zhang. "Automatic video scene extraction by shot grouping." *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. Vol. 4. IEEE, 2000.

[12] Chen, Liang-Hua, Yu-Chun Lai, and Hong-Yuan Mark Liao. "Movie scene segmentation using background information." *Pattern Recognition* 41.3 (2008): 1056-1065.

[13] Li, Jian, Shaogang Gong, and Tao Xiang. "Scene segmentation for behaviour correlation." *Computer Vision–ECCV 2008*. Springer Berlin Heidelberg, 2008. 383-395.

[14] Guan, Genliang, et al. "Keypoint-based keyframe selection." *Circuits and Systems for Video Technology, IEEE Transactions on* 23.4 (2013): 729-734.

[15] Kender, John R., and Boon-Lock Yeo. "Video scene segmentation via continuous video coherence." *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998.

[16] http://www.vlfeat.org

[17] http://www. mathworks.com