



Analysis and Extraction of Different Dialogue Scenes from Video Patterns

Ravi H. Acharya¹, Prof. Virendra singh², Vipul B. Patel³

Department of ECE, Sagar Institute of Research & Technology, Bhopal, Madhya Pradesh, India^{1,2,3}

raviacharya7@gmail.com¹, virendrasingh1180@gmail.com², vbpatel1678@gmail.com³

Abstract— In this paper we propose a method for the detection of dialogue scenes within movies. This task is of particular interest given the special semantic role played by dialogue based scenes in the most part of movies. A set of methods are prepared for detection, where analysis may be applied in order to detect different types of dialogue scenes. One reference dialogue frames from each shot is extracted from the whole movie and stored in to database and set of methods includes Edge Detection Techniques like sobel operator as well as canny operator for scene extraction, Direct Cosine Transform(DCT), I-DCT and simple correlation are applied to designed FSM model for dialogue scene detection from movie.

Keywords— Correlation, Canny operator, DCT, Dialogue scene Detection, edge detection technique, FSM model, I-DCT, Sobel operator.

I. INTRODUCTION

Movies constitute a large sector of the entertainment industry as over 9.000 hours of video are released every year [1]. Semantic content-based video indexing offers a promising solution for efficient digital movie management. Event analysis in movies is of paramount importance as it aims at obtaining a structured organization of the movie content and understanding its embedded semantics as humans do. A movie has some basic scene types, such as dialogues, stories, actions, and generic. Movie dialogue detection is the task of determining whether a scene derived from a movie is a dialogue or not. Movie dialogue detection is a challenging problem within movie event analysis, since there are no limitations on the emotional state of persons, the rate at which scenes interchange, the duration of silent periods, and the volume of background noise or music. For example, the detection of dialogue scenes in a movie is more complicated than detecting changes between anchor persons in TV-news, since many different scene types are incorporated in movies depending on the movie director [2].

In another approach for capturing dialogue scene, certain factors must be considered:

- Arrangement of actors in successive frames, and
- Placement of cameras either static or non-static that capture scenes.
- Background of the scene during recording.

Through the analysis of actor arrangement and camera placement, we find that there are only eight basic types of video shot patterns in a three person (call a, b and c) dialogue scene.

- Type A shot : A shot in which only actor a's face is visible throughout the shot;
- Type B shot : A shot in which only actor b's face is visible throughout the shot;
- Type C shot : A shot in which only actor c's face is visible throughout the shot;
- Type D shot : A shot in which actor a and b, both of their faces visible;
- Type E shot : A shot in which actor b and c, both of their faces visible;
- Type F shot : A shot in which actor c and a, both of their faces visible;
- Type G shot : A shot in which actor a, b and c, all of their faces visible;
- Type # shot: A shot that introduces usually cut-away shot not covering any of the above type shots but related to dialogue.

Hence the set of video type shots or VSS (Video shot string) formed is given by $V = \{A, B, C, D, E, F, G, \#\}$.

This paper is organized as follow: Section I gives the introduction of Finite state machine model. Section II,III and IV gives introduction of different schemes used for Dialogue scene detection. Section V explains How to find thresold. Section VI shows simulation results and analysis using different schemes and last section VII concludes the paper and followed by the references.

II. FINITE STATE MACHINE MODEL

The state transition algorithm can be used for detection and extraction of dialogue scene from a movie, where a scene is detected if there exist a path from initial state to final state. First of all dialogue sequences along with comedy scenes are identified based on the camera motion activity, and then these sequences are either accepted or rejected based on the clustering results and are chosen solely from the output of the camera motion analysis. Thus, the detection block looks for sequences that contain predominantly similar shots. It thus uses a state machine to determine the start and end points of these sequences as shown in Fig 1.1 and the symbols related to same figure is shown in Table 1.

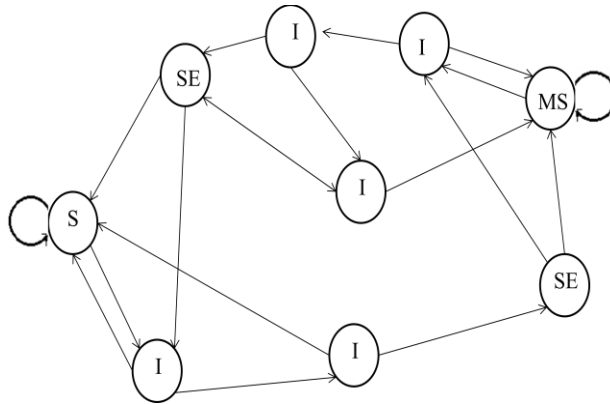


Fig 1.1 State machine model for scene detection.

TABLE I
 STATE MACHINE SYMBOLS

Symbols	Abbreviations
S	Start
I	Intermediate sequence
SS	Scene Start
MS	Maintain scene
SE	Scene end

If the state machine encounters a number of semantic shots in a row, based on conversation or a dialogue or action, then it declares this to be the start of a SS else it encounters intermediate shots. As long as there are similar shots in the sequence, the state machine will remain in some intermediate state. It will then take a number of non-similar shots in order for the state machine to declare an end to the DS. A scene is said to be detected if there exist a path from initial state to final state.

On basis of above discussed state transition model, a finite state machine model to detect at most three person conversations is designed. The model is shown in Figure 1.2.

The concept of Video shot string (VSS) is introduced to extract all of various kinds of scenes in a movie. VSS contains set of video shots that are possibly to occur in movie for three person conversations or action among which each of them represents type of shot in a video and occurs in a sequence from left to right. It probably covers all varieties of scenes occurring in a movie. VSS is string composed of shots present in set V.

We accept those 27 types of shots in VSS as regular language of DS, which can be further expanded to extract all possible DS in movie. Proof of one of the case is given as follows: A set $V = \{A, B, C, D, E, F, G, \#\}$ is considered as regular language, where sequences occurring in Table 2, like $\{ABCABC\}$ is formed by product of language $\{A\}, \{B\}, \{C\}, \{A\}, \{B\}$ and $\{C\}$ and all these languages are concatenated to form reference shot, which can be further expanded by simply appending type of shots from

Table 3. Thus appending a shot to a scene is nothing but a concatenation. Therefore, by definition of regular language, ABCABC is a regular language over V, which can further be proved for all possibilities of Elementary scenes in Table 2.

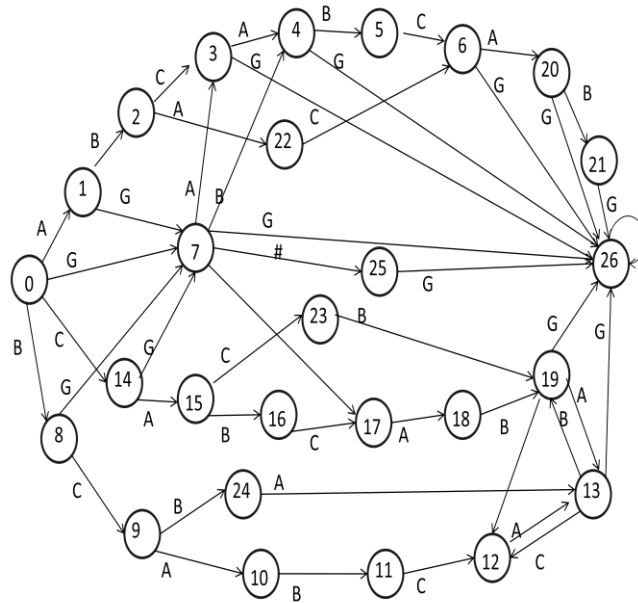


Fig 1.2 FSM designed for extraction of DS between three actors.

TABLE III
PROBABILITY BASED ANALYSIS ON TYPES OF ELEMENTARY SCENES

ELEMENTARY SCENES
ABCABC or BCABCA or CAB CAB
AGBC or BGCA or CGAB or GBCA or GCAB
ABAC or BCBA or CACB
AGA or BGB or CGC
AGB or BGC or CGA or AGC or BGA or CGB
AGG or BGG or CGG
G or G#G
GAG or GBG or GCG

TABLE IIIII
POSSIBILITY OF SHOTS THAT CAN BE APPENDED TO EXPAND

Type of end shot in a scene	Type of shot that can be appended for expansion of the scene
A	B or C or G
B	C or A or G
C	A or B or G
D	C or G
E	A or G
F	B or G
G	A or B or C or G

An elementary scene is considered as a set of video shots that can itself be a dialogue scene or be expanded to a longer dialogue scene and are determined statistically based on probability of scene occurrence and are applied on different movies for detection of action and dialogue scenes. As a result, I have identified 27 types of elementary scenes based on probability of their occurrence, as depicted in Table 2 for different movies under consideration.

On the basis of these shots shown in Table 2, all kind of dialogue as well as action scenes can extracted from movie, by simple expanding and appending Elementary Scenes with eight types of shots from VSS. For example, if the ending shot of one scene is an A type shot, usually a B type shot or a C type shot is appended to expand the scene. Similarly, the editor can append a G type shot to remind the audience of the whole scenario surrounding the dialog scene. Table 2 lists type of shots that can be appended to expand the scene.

III. DISCRETE COSINE TRANSFORM

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in applications related to compression, thus it turns out that cosine functions are much more efficient as fewer functions are needed to represent a typical signal. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the I-DCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of overlapping data.

This algorithm is used for removing redundancy in frames and thereby compressing it for further processing in compressed domain. Thus video processing in compressed domain is possible by this algorithm. It represents the standard that can handle wide range of images depending upon the compression required. The DCT is used in image as well as video compression. The two-dimensional DCT-II of $N \times N$ blocks are computed known as Macro-Blocks, and the results are quantized and entropy coded. In this case, N is typically 8 and the DCT-II formula is applied to each row and column of the block of an image. The result is an 8×8 transform coefficient array in which the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies. All the values obtained for each 8×8 blocks are called DCT coefficients. These coefficients represents whole image in terms of values and reduce the storage data required for it. Thus spatial and temporal redundancy is removed.

The **dct2** function computes the two-dimensional discrete cosine transform (DCT) of an image. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT known as "DCT Coefficients". For this reason, the DCT is often used in image compression applications. The two-dimensional DCT of an M -by- N matrix A is defined as follows:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)q}{2M} \cos \frac{\pi(2n+1)p}{2N} \quad (2.1)$$

where $0 < p < M - 1$ and $0 < q < N - 1$.

$$\begin{aligned} \alpha_p &= 1/\sqrt{M}, p = 0 & \alpha_q &= 1/\sqrt{N}, q = 0 \\ \alpha_p &= \sqrt{2/M}, 1 \leq p \leq M - 1 & \alpha_q &= \sqrt{2/N}, 1 \leq q \leq M - 1 \end{aligned} \quad (2.2)$$

These are DCT coefficients which can be quantized and coded on real images into significant number having smaller magnitudes with little amount of distortion. After applying DCT equation on images they are processed by 8x8 macro blocks by **blkproc()** function in matlab, as a result we get DCT image of reference frame. Now DCT is applied to whole input video into 8x8 blocks and DCT image of whole video is obtained. This DCT of video is Correlated with DCT image of all reference frames based on pre-defined threshold, where threshold is obtained by finding mean and variance of frame from each shot respectively. Maximum correlated frames are stored in variable which shows accuracy of results, discussed in detail in Simulation results chapter. Hence our algorithm is used to process on video in compressed domain. Block diagram of this algorithm is shown as follows:

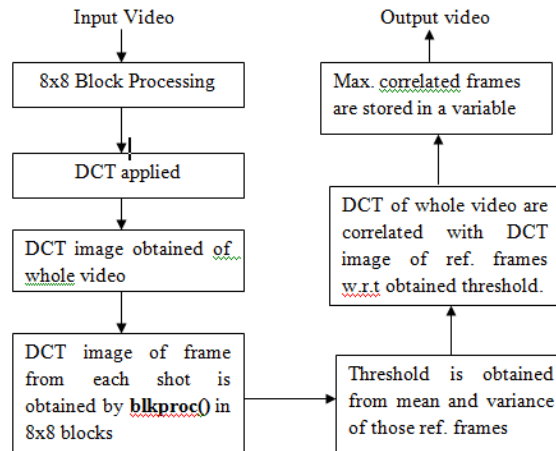


Fig 2.1 Block diagram of scene detection using DCT

The transformation scheme can be directly gauged by its ability to pack input data into as few coefficients as possible. This allows discarding the coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. DCT exhibits excellent energy compaction for highly correlated images. Let us see with an example, that DCT image of any frame after applying this algorithm shows that lower frequency component is concentrated on top most left side which is called dc frequency of an image located at (0,0) position and as we go away from that point in zigzag form the ac frequency starts increasing. It will shown in Simulation results.

IV. SOBEL OPERATOR

This operator consists of a pair of 3x3 convolution kernels as shown in Figure 3.1. One kernel is simply the other rotated by 90°.

-1	0	+1
-2	0	+2
-1	0	+1

$$G_x$$

+1	+2	+1
0	0	0
-1	-2	-1

$$G_y$$

Fig 3.1 Mask Of sobel operator to detect edges

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these G_x and G_y). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3.1)$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y| \quad (3.2)$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (3.3)$$

V. CANNY'S EDGE DETECTION ALGORITHM

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection", 1986. In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T_2 .

VI. THRESHOLD AND SCALING FACTOR FOR DIALOGUE DETECTION

Steps for selection of threshold and scaling factor are discussed as follows:



- For Dialogue scenes, from each of the shots, one frame is selected as reference frame and based on those selected many reference frames for a movie, mean and variance are calculated respectively.
- Now for satisfactory extraction of Dialogue scenes, we need to have a fixed threshold for correlation as well as for de-correlation concept for all the algorithms.
- Thus random value of Scaling factor is selected and applied with obtained mean and variance, for which we get different thresholds for each value of scaling factor.
- Hence value with minimum scaling factor giving desired threshold from all those reference frames, is selected as our final scaling factor and based on this threshold accuracy results are evaluated.
- Depending on all the applied different algorithms, the threshold for same algorithm is almost similar for the whole Genre in all movies for which the varying term is scaling factor which depends on type of shots.
- Hence after particular threshold is obtained, it can be used for correlation in frame based dialogue scene detection, which is highlighted more in Simulation results.

VII. SIMULATION RESULTS

One of the main tool box required for video processing is computer vision toolbox, where Computer Vision System Toolbox of ver4.1 in matlab2011b is used in our work, which provides algorithms and tools for the design and simulation of computer vision and video processing systems. The toolbox includes algorithms for feature extraction, motion detection, object detection, object tracking, stereo vision, video processing, and video analysis. Tools include video file I/O, video display, drawing graphics etc.

The movie database of Dialogue scenes for different movies is obtained, where movies are first segmented into scenes and then shots to extract appearance of our three main characters for dialogue scene. The reason for selecting those movies is because they are popular enough in terms of dialogue. Dialogue scenes those are extracted, generally refers to at most three person conversation, that are obtained by correlating all the incoming shots with that of the reference one and similarly clustered shots are stored in one of the variable in terms of frames, which can be further played as a movie by using `implay` command. The complete flow of work is reflected by the flow chart shown in figure as below: The Flow chart shows the application of different algorithms on movies for frame based detection of dialogue. The pattern of Dialogue scenes are shown for movies in next section.

Based on all the four algorithms discussed above, the simulations are carried out 2 different movies under the Dialogue Genre, and the corresponding obtained data are shown in form of tabular results each for separate algorithm. **Frame based** Dialogue scene detection is carried out and for Action scene **Shot based** approach is used. Thus Five tabular results are generated for each movie, where four tables are for four different algorithms and fifth table showing the accuracy of results, in terms of Recall, Precision and F1.

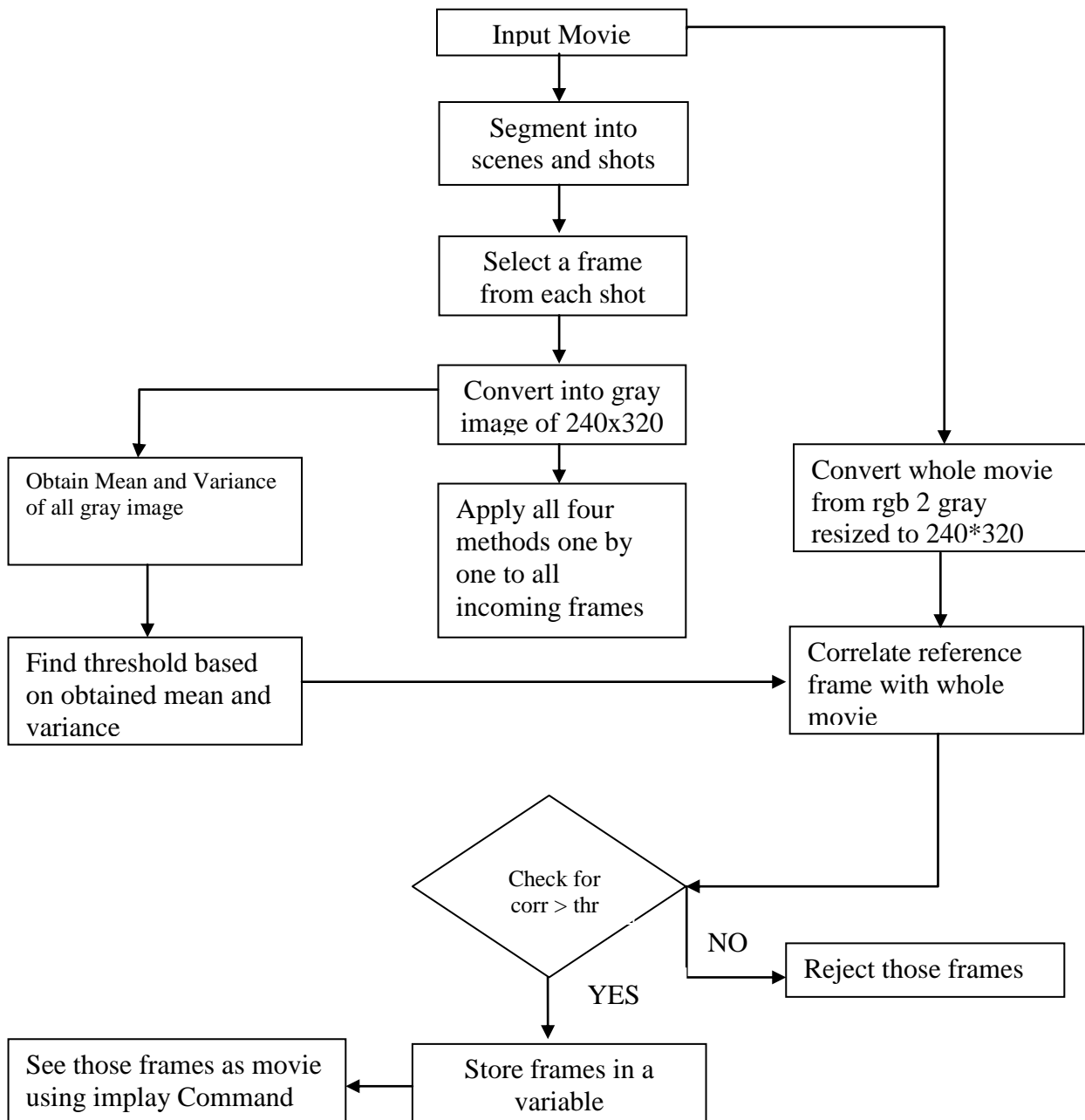


Fig 6.1 General flow chart for dialogue and action scene detection



Fig 6.2 Pattern of Dialogue scene as detected in matlab.

Name	Abbreviation
Movie 1 : Kabhi khushi kabhi Gham	KKKG
Movie 2 : Mukkadar ka Sikandar	MKS
Desired frames	DF
Correctly detected frames	CD
Missed	M
False Positives	FP
Scaling Factor	SF
Scene	SCN
Shot	SHT
Frame	FR

Movie 1: KKKG
Dialogue Scene detection:

Table IV
Based on CORRELATION in FSM model for movie1

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
KKKG	1-1000	1	6	960	958	42	40	45	0.4
	103.8-104.8k	3	8	812	738	81	0	185	
	145.5_146.5k	1	12	544	545	0	0	945	
	164-165k	2	6	770	580	165	0	335	
	167-168k	1	11	690	504	278	0	134	
	189-190k	2	5	654	485	171	0	225	
	194.5-196.5k	2	12	1732	1552	231	36	332	

Table V.
Based on CANNY Operator for movie1

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
KKKG	1-1000	1	6	960	901	99	40	-3.3	0.12
	103.8-104.8k	3	8	812	833	58	64	1.73	
	145.5_146.5k	1	12	544	635	0	86	0.98	
	164-165k	2	6	770	762	50	45	3.62	
	167-168k	1	11	690	560	199	65	12.7	
	189-190k	2	5	654	742	102	187	11.44	
	194.5-1196.5	2	12	1732	1893	26	184	28	

Table VI.
Based on SOBEL Operator for movie1

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
KKKG	1-1000	1	6	960	837	163	40	-38	0.01
	103.8-104.8k	3	8	812	823	91	124	-27	
	145.5_146.5k	1	12	544	557	13	26	-15	
	164-165k	2	6	770	662	162	12	-21	
	167-168k	1	11	690	617	179	85	-34	
	189-190k	2	5	654	703	111	168	-22	
	194.5-196.5	2	12	1732	1913	30	207	-17	

Table VII.
Based on DCT Operator for movie1

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
KKKG	1-1000	1	6	960	1000	0	40	9	0.85
	103.8-104.8k	3	8	812	593	285	65	2	
	145.5_146.5k	1	12	544	545	0	0	9	
	164-165k	2	6	740	941	16	125	4.9	
	167-168k	1	11	690	603	192	103	3.3	
	189-190k	2	5	654	867	3	213	1.3	
	194.5-196.5	2	12	1732	1461	307	32	3.8	

For each shot in dialogue scenes a reference frame is selected ,whose mean and variance is calculated and thus based on that obtained result, one scaling factor is chosen with minimum value to get a pre-defined threshold for the correlation concept in FSM model. Once this threshold is obtained, it is applied on the different algorithms respectively. Thus for different algorithm we get different threshold based on mean and variance of respective frames after applying the algorithm on that frame too. Now based on the results obtained, they are compared with each other in terms of R,P and F1 respectively as follows:

Table VIII.
Performance evaluation between applied algorithms for movie1

Methods (KKKG)	TF	TD	TCD	M	FP	R (%)	P (%)	F1 (%)
Correlation	8000	6162	5362	968	76	84.70	98.6	91.13
Canny	8000	6162	6326	534	671	92.21	90.04	91.49
Sobel	8000	6162	6112	749	662	89.08	90.22	89.65
DCT	8000	6162	6010	803	578	88.34	91.22	89.76

In the above result, it shows that for video KKKG having 8000 frames (12 scenes and 60 shots), in terms of R, P and F1 are almost similar for all algorithms because all those scenes include somewhat static shots having no change in background as well as object having little motion. As seen from above results correlation in fsm model gives poor results in terms of recall compared to all other methods but its precision is best, overall canny gives better recall and precision.

MOVIE 2 : MKS

Table IX.
Based on CORRELATION in FSM model for movie2

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
MKS	3-5k	2	3	1803	1669	140	7	-65	0.3
	8-9.5k	2	6	1266	1268	0	0	5.5	
	173-174k	2	3	884	766	122	0	22	
	7.6-8.6	1	6	936	951	37	61	118	
	42.2-43.7	2	7	1245	1109	232	92	-285	
	132.3-133.8	1	4	1310	1309	4	0	-335	

Table X.
Based on CANNY Operator for movie2

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
MKS	3-5k	2	3	1803	1827	17	37	-27	0.02
	8-9.5k	2	6	1266	1324	11	66	-64	
	173-174k	2	3	884	654	233	0	-36	
	7.6-8.6	1	6	936	953	3	15	-38	
	42.2-43.7	2	7	1245	1279	96	128	-25	
	132.3-133.8	1	4	1310	1336	1	24	-50	

Table XI.
Based on SOBEL Operator for movie2

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
MKS	3-5k	2	3	1803	1805	42	39	-17	0.02
	8-9.5k	2	6	1266	1252	44	30	-14	
	173-174k	2	3	884	799	92	0	-10	
	7.6-8.6	1	6	936	954	0	16	-30	
	42.2-43.7	2	7	1245	1238	104	94	-11	
	132.3-133.8	1	4	1310	1418	0	106	-18	

Table XII.
Based on DCT Operator for movie2

Movie	FR	SCN	SHT	DF	CD	M	FP	SF	THR
MKS	3-5k	2	3	1803	1957	22	173	11.3	0.85
	8-9.5k	2	6	1266	1055	196	0	3.0	
	173-174k	2	3	884	531	356	0	12.3	
	7.6-8.6	1	6	936	678	261	0	8.9	
	42.2-43.7	2	7	1245	1294	0	136	6.7	
	132.3-133.8	1	4	1310	1209	103	0	2.7	

Table XIII.
Performance evaluation between applied algorithms for movie2

METHODS (MKS)	TF	TD	TCD	M	FP	R (%)	P (%)	F1 (%)
Correlation	8500	7444	7072	535	160	92.96	97.78	95.11
Canny	8500	7444	7373	361	270	95.33	96.46	95.9
Sobel	8500	7444	7466	282	285	96.36	96.32	96.34
DCT	8500	7444	6724	938	309	87.76	95.61	91.27



In the above result, it shows that for video MKS, having 8500 frames (10scenes and 29 shots), in terms of R,P and F1 are beyond 90% for all algorithms because those scenes include somewhat static shots very little object in motion, so on such shots having at most 3 persons conversation, edge operator gives powerful results in terms of precision and recall while DCT results in terms of recall is less compared to others. In terms of precision almost all methods gives above 95% , which is appreciable.

VIII. CONCLUSION & FUTURE WORK

In this paper, applied approach as algorithm is used for detecting and extracting Dialogue scenes from a movie, which is reflected in simulation results. A Finite State Machine realization is shown for the detection and extraction of scenes in compressed as well as uncompressed domain by setting some generalized rules.

The FSM model designed is able to detect dialogue as well action scenes using correlation and de-correlation concept by four different algorithms respectively. But this FSM model is not applied to detect any other Genre type of video rather than dialogue .

Hence as future work, with modification or some changes in same model, it can be applied to detect horror scenes or sad scenes or any other part as desired from the movie and that too on large data base.

REFERENCES

- [1] A. A. Alatan, A. N. Akansu, and W. Wolf, "Comparative analysis of hidden Markov models for multi-modal dialogue scene indexing", in Proc. 2000 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. 4, pp. 2401-2404, 2000.
- [2] M. De Santo, G Percannella, C. Sansone, and M. Vento, "Dialogue scenes detection in MPEG movies: A multi-expert approach", in Lecture Notes in Computer Science, vol. 2184, no. 5, pp. 192-201, September.
- [3] L. Chen, S. J. Rizvi, and M. T. Ozsü (2003), "Incorporating audio cues into dialog and action scene extraction", in Proc. Storage and Retrieval for Media Databases, vol. 5021, pp. 252-263, January 2003.
- [4] [L. Chen, et. Al,2002] L. Chen and M. Tamer(2002), "Rule based scene extraction from video" ,ICIP 2002 International IEEE Conference.
- [5] J.F. Canny, "A Computational Approach to edge detection(2007) ", IEEE Transaction Pattern Anal.Machine Intell, 2007.
- [6] Raman Maini, Dr.Himanshu aggarwal (2009), "Study and Comparison of Various Image Edge Detection Techniques", IJIP Journal,2009.