ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 7, October 2013

Design of User Interface for Shot Boundary Detection using Classical Methods

Prof. Anand P. Mankodia[1] , Prof. Satish K. Shah[2]

Ph.D. Student, U. V. Patel College of Engineering, Ganpat University, Mehsana, Gujarat, India[1]

Ex-Professor & Head of Electrical Engineering Department, MS University of Baroda, Gujarat, India[2]

Abstract: Video summarization refers to creating a summary of a digital video. With the advent of digital multimedia, a lot of digital content such as movies, news, television shows and sports is widely available. Also, due to the advances in digital content distribution (DTH satellite reception) and digital video recorders, this digital content can be easily recorded. However, the user may not have sufficient time to watch the entire video (Ex. User may want to watch just the highlights of a game) or the whole of video content may not be of interest to the user(Ex. Golf game video). In such cases, the user may just want to view the summary of the video instead of watching the whole video.

Thus, the summary should be such that it should convey as much information about the occurrence of various incidents in the video. Also, the method should be very general so that it can work with the videos of a variety of genre.

Current research topics on video includes video abstraction or summarization, video classification, video annotation, content based video retrieval. In nearly all these application one needs to identify shots in video which will correctly and briefly indicate the contents of video. This paper compares some of the popular shot boundary detection techniques; discuses the merits and demerits of each of the techniques and some experiments.

Key Words: Color Histogram Difference Based, Edge Change Ratio Based, Pixel Based Techniques, Cut Detection, Shot Detection

I. METHODS OF VIDEO SUMMARIZATION

a) Video Summarization using singular value decomposition.
b) Video Summarization using motion descriptors.
c) Video Summarization using clustering.
d) Video Summarization using shot boundary detection.

The proposed method in this report is Video Summarization using shot boundary detection.

II. SHOT BOUNDARIES

In this work we are mostly interested in the shot level of the structural hierarchy of videos. The shots are clearly visually separable by humans. Most of the transitions between shots can be classified into abrupt cuts and gradual fades, dissolves and wipes [1].

Digital editing allows us to create countless types of additional transition effects. Some directors use these for stylistic effect. However, most of the cuts in typical news broadcasts can be classified into the aforementioned four classes. For shot boundary detection purposes it is usually sufficient to be able to recognize the instance when a transition takes place, and the transition classes are not that important. Knowing the most typical transition classes can however help us gain insight into the problem at hand. Some shot boundary detection algorithms also use separate detectors for separate transition classes [2], and therefore knowing the characteristic properties of each transition type can be useful.

III. METHODS FOR SHOT BOUNDARY DETECTION

There are many approaches for shot boundary detection like block based methods, feature based methods, histogram based methods etc. Out of all these algorithms proposed algorithm in this report are: Color histogram difference method, Edge change fraction method, Pixel difference method.

3.1 Histogram difference[1]

In this algorithm color histogram difference is taken between consecutive frames, Color histogram difference is nothing but a color component R,G,B difference of two consecutive frames. If this difference is greater than some threshold then shot change is occurred. Generally histogram based method are used because of their global aspect. The main advantage of this method is that they are robust against camera or object motion.

3.1.1 Grey level histogram difference

This method based on gray-level histograms. Images are compared by computing a distance between their histograms, as shown in the following equation:
Detection if:

$$(\sum_{v=0}^{V}|H(I_t, v) - H(I_{t-1}, v)|) > T \ldots (3.1)$$

Where T is threshold,I_t and I_{t-1} are the frames taken at time interval t and t-1.so if this equation is satisfied then shot change is detected.

3.1.2 Color histogram difference

This method uses 64 bins for color histograms (2 bits for each color component in RGB space).Using the notation $H_{64}(I_t, v)$, the detection is defined by:

Detection if:

$$\left(\sum_{v=0}^{63} |H_{64}(I_t, v) - H_{64}(I_{t-1}, v)|\right) > T \ldots (3.2)$$

Similarly for the RGB computes three histogram differences, considering separately the three color components of the RGB space. The highest value is compared to a threshold for shot change detection.

3.2 Edge change fraction method [1]

Another important feature that has been proved to be useful in detection of shot boundaries is Edges. The ECR attempts to compare the actual content of two frames. It transforms both frames to edge pictures, i. e. it extracts the probable outlines of objects within the pictures. This is done by using the edge detection operators. Afterwards it compares these edge pictures using dilatation to compute a probability that the second frame contains the same objects as the first frame.

The basic idea of edge change fraction (ECF) comparison method is summarized as following.

- Detect edges in two consecutive frames f_n and f_{n+1} respectively. I have used Canny's edge detector operator for edge detection.
- Count the number of edge pixels δ_n and δ_{n+1} in frame f_n and f_{n+1}.
- Define the entering and exiting edge pixel E_{n+1}^{in} and E_n^{out} .

Suppose we have two images Im_n and Im_{n+1}. The entering edge pixels E_{n+1}^{in} are the fraction of edge pixels in Im_{n+1} which are farther than a fixed distance r away from the closest edge pixel in Im_n. Similarly exiting edge pixels E_n^{out} are the fraction of edge pixels in Im_n which are farther than a fixed distance r away from the closest edge pixel in Im_{n+1} .

- Compute the Edge change fraction ECF_n between the frames f_n and f_{n+1}:

$$ECF(n, n+1) = max\left(\frac{E_{n+1}^{in}}{\delta_{n+1}}, \frac{E_n^{out}}{\delta_n}\right)$$

3.3 Pixel difference method [1]

This is both the most obvious and most simple algorithm of all: The two consecutive frames are compared pixel by pixel, summing up the absolute values of the differences of each two corresponding pixels. Pixel difference method reacts very sensitively to even minor changes within a scene: fast movements of the camera, explosions or the simple switching on of a light in a previously dark scene result in false hits. On the other hand, pixel difference method hardly reacts to soft cuts at all; it detects all visible hard cuts. Pixel comparison between two consecutive frames is one of the first methods described in literature was from Nagasaka et al in 1991. Shot changes are detected using a simple global inter-frame difference measure, defined as: Detection if:

$$\left(\left|\sum_{i=1}^{X}\sum_{j=1}^{Y} P(I_t, i, j) - \sum_{i=1}^{X}\sum_{j=1}^{Y} P(I_{t-1}, i, j)\right|\right) > T \ldots\ldots\ldots (3.3)$$

IV. Simulation Results

Simulation results are taken for three different types of video using three different techniques; viz, color histogram difference technique, edge change fraction method and Pixel difference method.

4.1 Color histogram difference technique

As discussed before in this technique color histogram difference i.e. respective R,G,B component difference of two consecutive frames is taken, and this difference is compared against a threshold if it is greater than the threshold then shot change is occurred. Results for three different types of video are shown below.

4.1.1 Movie trailer

Figure shows the plot of color histogram difference versus frame number. Here video is movie trailer which contains 5455 frames.

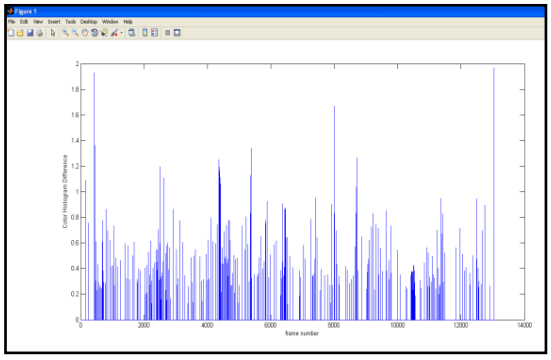


Fig.4.1. Results of movie trailer video (Movie.avi, 5455 frames)

4.1.2 Sports video

Figure shows the plot of color histogram difference versus frame number. Here type of video used is sports which contain 13104 frames.

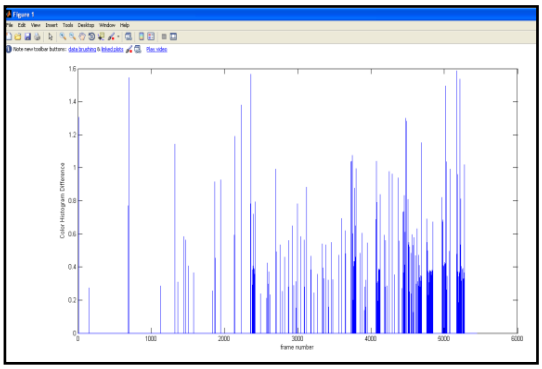


Fig.4.2. Results of Sports video (Sports.avi, 13104 frames)

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 7, October 2013

4.1.3 Animation

Figure shows the plot of color histogram difference versus frame number. Here type of video used is Animation which contains 4775 frames.

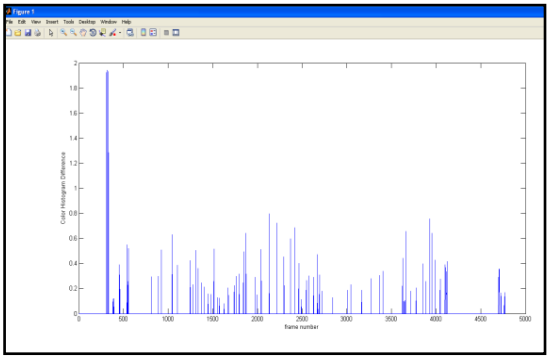


Fig.4.3. Results of Animation video
(Animation.avi, 4775 frames)

From the above all results of different types of video using color histogram difference technique following table is made.

Type of video	Number of frames	Size (MB)	Time (sec)	Computation time (sec)	Computa-tion Factor
Movie Trailer	1)5455	30.3	182	347.39	1.908
	2)4083	26.1	170	164.16	0.965
Sports	13104	85.9	437	661.29	1.513
Animation	4775	22.2	199	237.71	1.195

Table 4.1. Color histogram difference based method

4.2 Edge change fraction method

Another important feature that has been proved to be useful in detection of shot boundaries is Edges. In this method First the edges of two consecutive frames are detected by using the canny edge detector, then number of edge pixels are calculated in consecutive frames, then number of entering and exiting edge pixels are calculated, then ratio of entering edge pixel & edge pixel of higher frame and ratio of exiting edge pixel & edge pixel of lower frame is calculated, maximum value of these two quantities gives edge change fraction. Results of three different types of videos are shown below:

4.2.1 Movie trailer

Figure below shows the plot of edge change fraction versus frame number; here type of video used is movie trailer which contains 5455 frames:

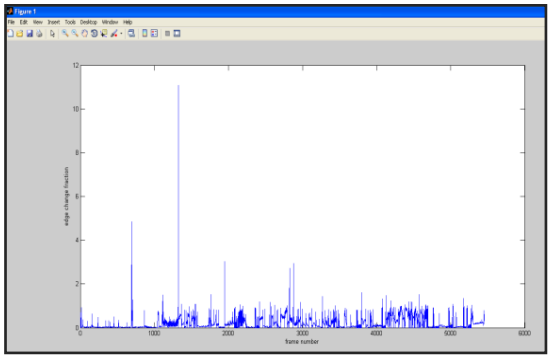


Fig.4.4. Results of movie trailer video
(Movie.avi, 5455 frames)

4.2.2 Sports video

Figure below shows the plot of edge change fraction versus frame number; here type of video used is a sport which contains 13104 frames:

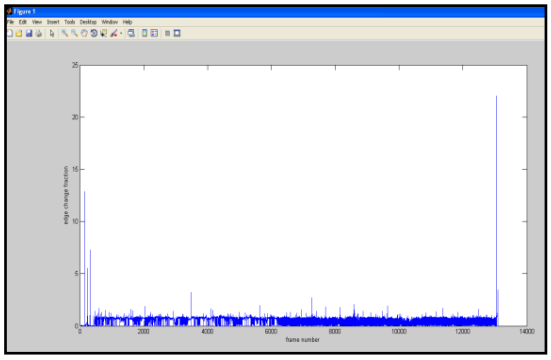


Fig 4.5. Results of Sports video
(Sports.avi, 13104 frames)

4.2.3 Animation

Figure below shows the plot of edge change fraction versus frame number; here type of video used is animation which contains 4775 frames:

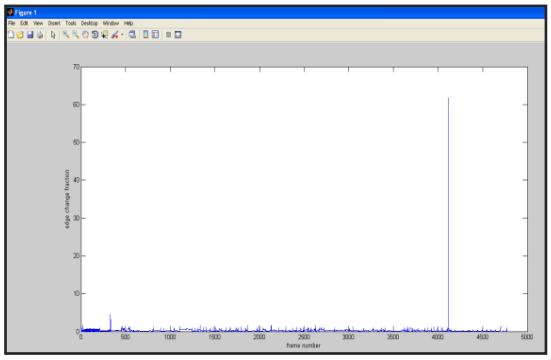


Fig.4.6. Results of Animation video
(Animation.avi, 4775 frames)

From the above all results of different types of video using edge change fraction method following table is made.

Type of video	Number of frames	Size (MB)	Time (sec)	Computation time (sec)	Computation Factor
Movie Trailer	1)5455	30.3	182	5125.43	28.162
	2)4083	26.1	170	1985.88	11.682
Sports	13104	85.9	437	7962.79	18.222
Animation	4775	22.2	199	2378.75	11.953

Table 4.2. Edge change fraction based method

4.3 Pixel difference method

This is both the most obvious and most simple algorithm of all: The two consecutive frames are compared pixel by pixel, summing up the absolute values of the differences of each two corresponding pixels. Pixel difference method reacts very sensitively to even minor changes within a scene: fast movements of the camera, explosions or the simple switching on of a light in a previously dark scene result in false hits. On the other hand, pixel difference method hardly reacts to soft cuts at all; it detects all visible hard cuts. Results of three different types of videos are shown below:

4.3.1 Movie trailer

Figure below shows the plot of sum of absolute inter-frame difference versus frame number, here type of video used is movie trailer which contains 5455 frames:

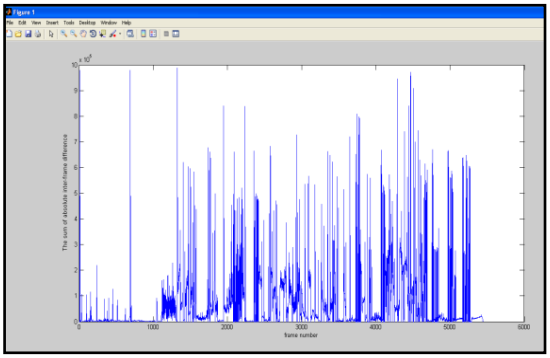


Fig.4.7. Results of movie trailer video (Movie.avi, 5455 frames)

4.3.2 Sports video

Figure below shows the plot of sum of absolute inter-frame difference versus frame number, here type of video used is sports which contains 13104 frames:

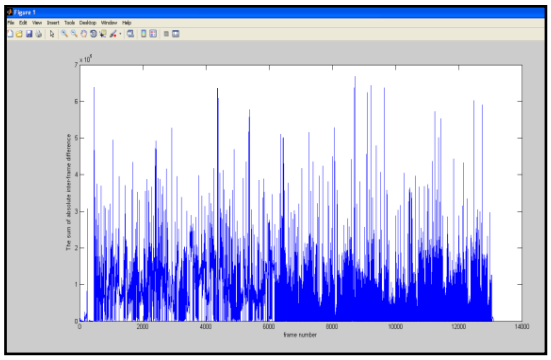


Fig.4.8. Results of Sports video (Sports.avi,13104 frames)

4.3.3 Animation

Figure below shows the plot of sum of absolute inter-frame difference versus frame number, here type of video used is animation which contains 4775 frames:

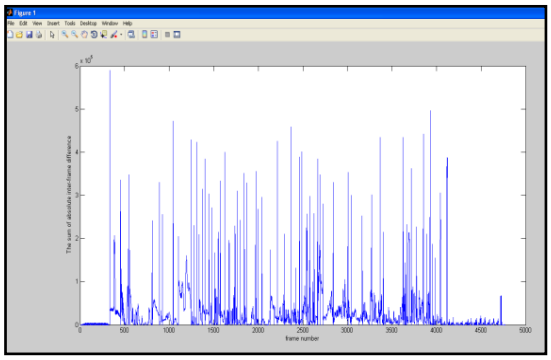


Fig.4.9. Results of Animation video (Animation.avi,4775 frames)

From the above all results of different types of video using pixel difference method following table is made.

Type of video	Number of frames	Size (MB)	Time (sec)	Computation time (sec)	Computation Factor
Movie Trailer	1)5455	30.3	182	560.70	3.081
	2)4083	26.1	170	289.58	1.703
Sports	13104	85.9	437	1203.66	2.754
Animation	4775	22.2	199	728.80	3.662

Table 4.3 Pixel difference method

For calculating desired values of scene change, manual checking of frames ranging from 1 to 1000 has been done. After completion of manual checking of data, algorithm is applied and takes the result of same number of frames. Comparison between the manually checked and results shown by the algorithm makes one analysis table for 1000 frames of each different types of video.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 7, October 2013

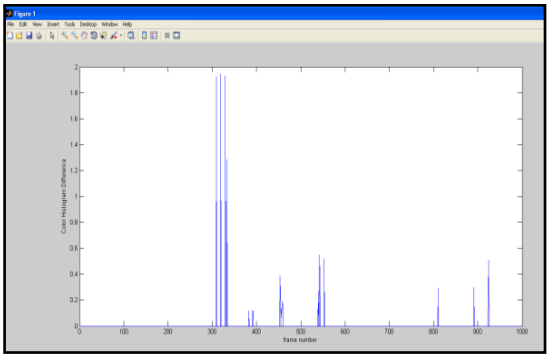
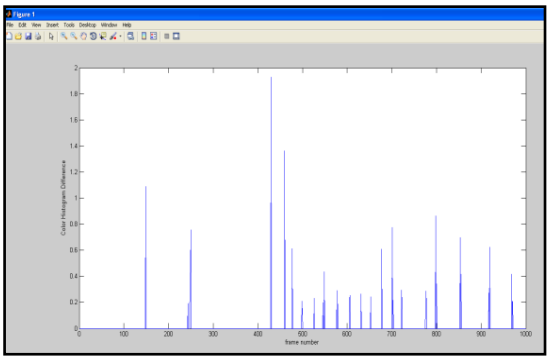


Fig.4.10. Results of Movie trailer video (1 to 1000 frames)



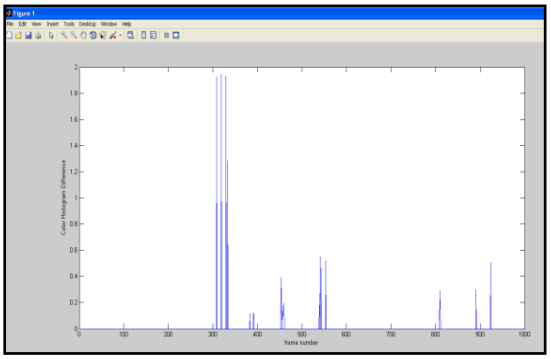


Fig.4.11. Results of Sports video (1 to 1000 frames)

Fig.4.12. Results of Animation video (1 to 1000 frames)

ALGORITHM: Color Histogram Difference method

Type of video	Frames	Desired	Correct	Missed	False Positive	Recall	Precision
Movie Trailer	1-1000	18	16	2	0	0.89	1
Sports	1-1000	23	22	1	2	0.96	0.92
Animation	1-1000	11	11	0	0	1	1

Table 4.4 Analysis of Different result for Color Histogram algorithm

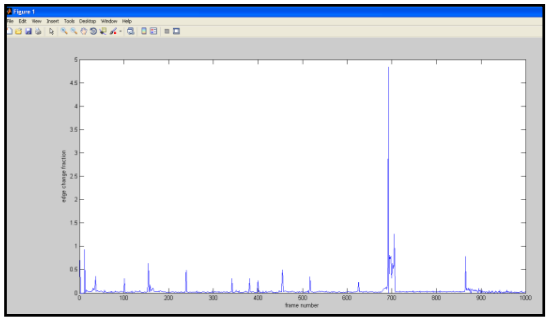


Fig.4.13. Results of movie trailer video (1 to 1000 frames)

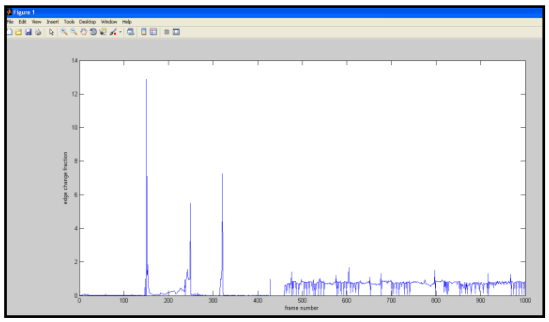


Fig.4.14. Results of Sports video (1 to 1000 frames)

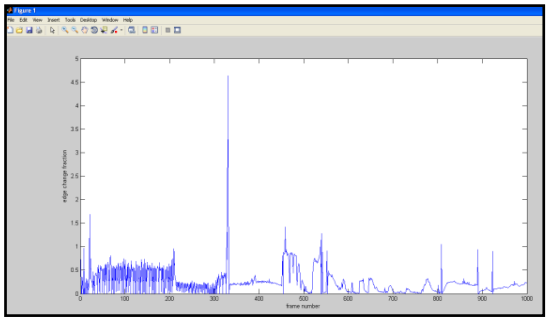


Fig.4.15. Results of Animation video (1 to 1000 frames)

ALGORITHM: Edge change fraction method

Type of video	Frames	Desired	Correct	Missed	False Positive	Recall	Precision
Movie Trailer	1-1000	18	17	1	0	0.94	1
Sports	1-1000	23	23	0	0	1	1
Animation	1-1000	11	10	1	2	0.91	0.83

Table 4.5. Analysis of Different result for Edge change fraction algorithm

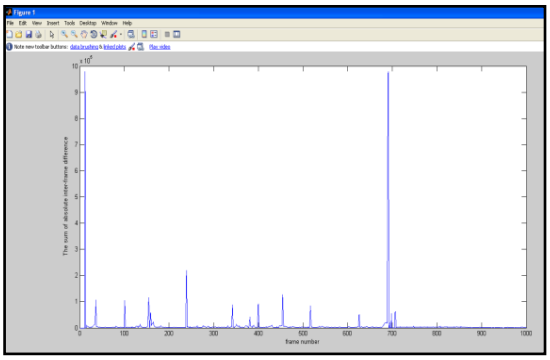


Fig.4.16. Results of movie trailer video (1 to 1000 frames)

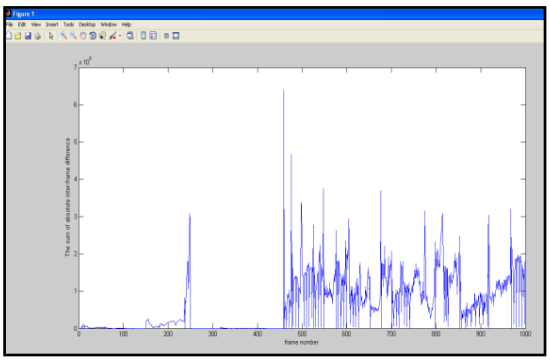


Fig.4.17. Results of Sports video (1 to 1000 frames)

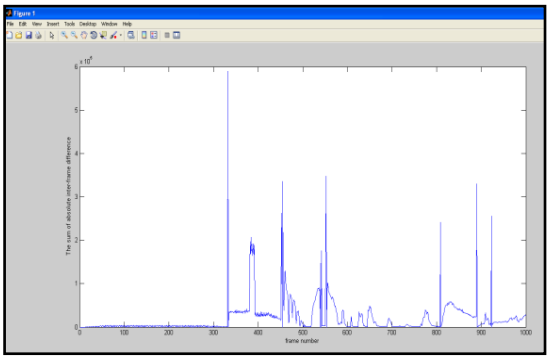


Fig.4.18. Results of Animation video (1 to 1000 frames)

ALGORITHM: Pixel difference method

Type of video	Frames	Desi -red	Cor- rect	Miss ed	False Posit ive	Recall	Preci -sion
Movie Trailer	1-1000	18	16	2	0	0.89	1
Sports	1-1000	23	22	1	0	0.96	1
Animation	1-1000	11	11	0	0	1	1

Table 4.6. Analysis of Different result for Pixel difference algorithm

From these tables the desired frames are the manually calculated frames, correct means the actual detection shown by the algorithm and that is calculated from the results. Missed is the detection which is not detected by the algorithm while false positive (FP) is the detection where there is actually no detection but algorithm shows.

V. GUI IMPLEMENTATION

5.1 Introduction

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth.

5.2 How a Graphical User Interface Works

A graphical user interface provides the user with a familiar environment in which to work. This environment contains pushbuttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that he or she can concentrate on using the application rather than on the mechanics involved in doing things. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time. Such inputs are known as events, and a program that responds to events is said to be *event driven*. The three principal elements required to create a MATLAB Graphical User Interface are,

1. Components: Each item on a MATLAB GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function uicontrol, and menus are created by the functions uimenu and uicontextmenu. Axes, which are used to display graphical data, are created by the function axes.

2. Figures: The components of a GUI must be arranged within a figure, which is a window on the computer screen. In the past, figures have been created automatically whenever we have plotted data. However, empty figures can be created with the function figure and can be used to hold any combination of components.

3. Callbacks: Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a call back. There must be a callback to implement the function of each graphical component on the GUI.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 7, October 2013

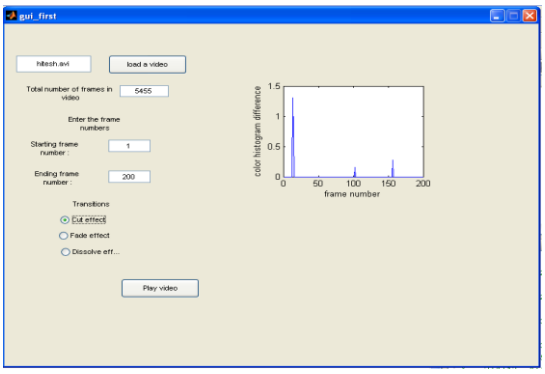


Figure 6.1 Front Panel of GUI

VI. CONCLUSION

Shot boundary detection is temporal video segmentation, and is the process of identifying the transitions between the adjacent shots. The work in video processing and analysis is a very important part of searching and browsing of digital video. We can use the shot boundaries to analyze video data in shot in greater depth such as video indexing, shot similarity etc. A shot is a consecutive sequence of frames captured by a Camera action that takes place between start and stop operations, which mark the shot boundaries. There are strong content correlations between frames in a shot. Therefore, shots are considered to be the fundamental units to organize the contents of video sequences and the primitives for higher level semantic annotation and retrieval tasks. Generally, shot boundaries are classified as cut in which the transition between successive shots is abrupt and gradual transitions which include dissolve, fade in, fade out, wipe, etc., stretching over a number of frames. Cut detection is easier than gradual transition detection.

From the above all experimental results we conclude that results obtained for different videos using different techniques are quite satisfactory. Pixel difference methods are sensitive to camera or object motion, so color histogram difference methods are employed. Color histogram difference methods are robust to camera or object motion. There is difficulty in finding dissolve using the color histogram difference method, so edge change fraction (ECF) method is used which can detect fades, dissolve and wipe.

REFERENCES

[1] Rainer Lienhart "Comparison of Automatic Shot Boundary Detection Algorithms" at Intel Corporation.

[2] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang "A Formal Study of Shot Boundary Detection" in IEEE Transaction on Circuits and Systems for Video Technology, Vol. 17, No. 2, February 2007.

[3] WayneWolf "Key frame selection by motion analysis" in Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96), volume 2, pages 1228–1231, Atlanta, Georgia, USA, May 7-10 1996.

[4] John S. Boreczky and Lawrence A. Rowe "Comparison of video shot boundary detection techniques" in Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases IV, pages 170–179, San Diego/La Jolla, CA, USA, January 28 - February 2 1996.

[5] Lei Zhu, Junfeng Qu, Muhammad Asadur Rahman, and Weihu Hong, "An Integrated Method for Video Shot Boundary Detection", Proc. IEEE SoutheastCon, pp. 151-154, 2010.

BIOGRAPHIES

Prof. Anand Mankodia is working as an Assistance Professor, Electronics and communication department at U. V. Patel College of Engieering, Ganpat University, Kherva, Mehsana, Gujarat, India. At present he is also pursuing Ph. D. in Engineering and Technology from the same university under the guidance of Prof. Satish K. Shah. He is having 11+ years of teaching experience. His primary areas of interest are Digital Image & Video Processing and Embedded Systems. He has guided more than 25 projects at UG/PG level. He is a member of ISTE and served as a faculty convener of ISTE student chapter at the institute. He has attended more than 20 and organized more than 10 seminars, workshops and symposium. He has presented 5+ research papers in national & international conferences.

Prof. Satish Shah is a Former Head of Electrical Engineering Department at FET, MS University of Baroda, (Guj) India with 30+ years of teaching and research experience. His primary areas of interest are Embedded Controller and soft computing. He has guided more than hundred projects at UG/PG level. He is a fellow of technical associations IETE, IE (I), ISA and served as the member, Hon Secretary and Treasurer of local executive committees for a span of six-eight years. He is the Chairman, IETE-Vadodara centre for the year 2010-2012. He is member of IEEE (NY) and a life member of ISTE. He has attended and organized more than 20 seminars, workshops, symposiums under QIP of UGC, AICTE, IETE, IE(I), World Bank Impact project etc. He has conducted Faculty training program on "Intelligent controllers" for the teachers of Engineering Colleges and Polytechnics in Gujarat under TECHSAT program of Gujarat Council of Science & Technology, Gandhinagar. His current areas of interest also include soft computing, Image and Signal processing and smart controller design using DSP. He has presented 20+ research papers in national & international conferences and published 10+ research papers in technical journals.