DETECTING FADE AND DISSOLVE TRANSITION IN COMPRESSED DOMAIN

[1]*ANKIT C.GAJERA*, [2]*RAVI G. MEHTA*

[1, 2] **Electronics and Communication Department ,**
Shantilal Shah Engg. College, Bhavnagar,
Gujarat Technological University , Gujarat, India.

[1]*ankit.c.gajera@gmail.com*, [2]*ravigmehta@yahoo.com*

ABSTRACT : *Shot boundary detection and key frame extraction are current research area in video processing. The algorithms which are used here are applied on the compressed domain data means algorithms are applied on compressed domain parameters of image, like D.C.T coefficient, D.C image. These presented algorithms are useful to find gradual changes like fade and dissolve in sequence of frames in compressed domain, which are difficult to detect than sharp transitions in the video. These algorithms use the concept of D.C. Coefficients and D.C image. Actually D.C image is spatially reduced version of image. Because of this concept the processing time for every frame is reduced so time required to run the programmed is less. Here there are two algorithms, one algorithm is used to detect the fade effect and other algorithm is used to detect dissolve effect. Here the concept of D.C image is used for processing of video. Algorithm used for finding fade effect uses calculation of mean of D.C image, because fade effect consist of transition from some content in frame to total black frame and from black frame to frame having some content. Another algorithm used for finding dissolve effect uses calculation of variance of D.C image. In dissolve there is transition from one scene to another scene where in between these two scenes one scene is fades out and another scene fades in.*

KEY WORDS *: Fade, Dissolve, DCT, Compression, D.C. image.*

I. INTRODUCTION

In video processing people are generally concerned about the content of frames and types of changes from one frame to another frame or change in the group of frames and that is all about key frame extraction(KFE) and shot boundary detection(SBD). In current generation most of the videos are in compressed domain so it desirable to detect the sharp and gradual transitions directly in the compressed domain.

A. DESCRETE COSINE TRANSFORM

A discrete cosine transform (DCT) is represented by sequence of finite data points in terms of a sum of cosine functions which are oscillating at different frequencies. There are many numbers of applications those use DCT in science and engineering, from loss compression of audio and images to solve the partial differential equations. The use of cosine rather than sine functions is critical in the following applications: in the compression, it is proved that cosine functions are more efficient than sine, whereas for differential equations the cosines have to take some of boundary conditions.

Like any Fourier-related transform, discrete cosine transforms (DCTs) express a function or a signal in terms of a sum of sinusoids which are at different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DCT is working at a finite number of discrete data points. The difference between DCT and a DFT is that the DCT uses only cosine functions, while DFT uses both cosine and sine. The Fourier-related transforms that operate on a function over a finite domain.

Based on different boundary conditions the applications of the transform are decided, and lead to uniquely useful properties for the various DCT types. In particular, it is well known that any discontinuities in a function reduce the rate of convergence of the Fourier series, so that more sinusoids are needed to represent the function with a given accuracy. The same principle governs the usefulness of the DFT and other transforms for signal compression: the smoother a function is, the fewer terms in its DFT or DCT are required to represent it accurately, and the more it can be compressed. (Here, we think of the DFT or DCT as approximations for the Fourier series or cosine series of a function, respectively, in order to talk about its "smoothness".) However, the implicit periodicity of the DFT means that discontinuities usually occur at the boundaries: any random segment of a signal is unlikely to have the same value at both the left and right boundaries. (A similar problem arises for the DST, in which the odd left boundary condition implies a discontinuity for any function that does not happen to be zero at that boundary.) In contrast, a DCT where both boundaries are even always yields a continuous extension at the boundaries (although the slope is generally discontinuous). This is why DCTs, and in particular DCTs of types I, II, V, and VI (the types that have two even boundaries) generally perform better for signal compression than DFTs . In practice, a type-II

JIKRECE

DCT is usually preferred for such applications, in part for reasons of computational convenience.

DCT II:

$$F(x,y) = \frac{C_u C_v}{2\ 2} \sum_{x=1}^{7} \sum_{y=1}^{7} \cos \frac{(2x+1)u\pi}{16} \sin \frac{(2y+1)v\pi}{16}$$

$$\text{Where } C_u = \frac{1}{\sqrt{2}} \quad \text{if } u = 0; \qquad = 1 \quad \text{otherwise}$$

$$C_v = \frac{1}{\sqrt{2}} \quad \text{if } v = 0; \qquad = 1 \quad \text{otherwise}$$

This transform is exactly equivalent to a DFT of 4N real inputs of even symmetry where the even-indexed elements are zero.

B. D.C. IMAGE

DC image is spatially reduced version of original image. This construction of DC image can be done after calculation of DCT. But in case of DC image the original image is divided in number of blocks. This division of image or frame in number of blocks is dependent on user. Once division of frame is done than DCT of every block is done separately. Then one by one take every module and take the average of all those elements of each blocks DCT. So there will be one value of mean of each block. And all those values of means are elements of DC image. The procedure of calculation is shown below:

First the calculation of DCT of an image is obtained. In this the image (frame) is first divided in the block of size 8*8. After this division in macro block DCT of every block is obtained. So each 8*8 block is having its DCT co-efficient.

The definition of finding DCT of image is shown below that is applied on each 8*8 block of frame which is shown in below equation .

$$F(x,y) = \frac{C_u C_v}{2\ 2} \sum_{x=1}^{7} \sum_{y=1}^{7} \cos \frac{(2x+1)u\pi}{16} \sin \frac{(2y+1)v\pi}{16}$$

$$\text{Where } C_u = \frac{1}{\sqrt{2}} \quad \text{if } u = 0; \qquad = 1 \quad \text{otherwise}$$

$$C_v = \frac{1}{\sqrt{2}} \quad \text{if } v = 0; \qquad = 1 \quad \text{otherwise}$$

In each block there will be one DC co-efficient and one or more A.C co-efficient. And by applying various formulas abrupt and gradual transitions can be detected. In this algorithm DC image of current frame is calculated and compared with DC image of next frame or frames. DC image of frame can be calculated by :

$$D.C(i,j) = \frac{1}{64} \sum_{x=0}^{7} \sum_{y=0}^{7} DCT(x,y)$$

In above equation the DCT(x,y) shows the 64 DCT values of one 8*8 block. And DC(i,j) shows the each component of DC image. In this project work all the methods are applied for detection for various things are directly applied on DC image of frame. Because of use of DC image concept processing time of algorithm reduces. And also it is useful for compression of video.

C. FADE TRANSITION

Gradual transition occurs over multiple frames. For finding gradual transitions care of data of multiple frames have to be taken. This gradual transition includes dissolve, fade in/out, wipe etc.

In fade there are two type, first in which change from total black picture to picture information of scene is called fade in. and second in which transition is from picture information of scene to total black screen.

Figure:1 Fade effect

In above figure fade out effect is shown, in that first frame is having some visual content and 4^{th} frame there is totally black. So pattern of removal of content from some visual content to complete black in the frame is called fade out.

D. DISSOLVE TRANSITION

In dissolve there is transition from one picture to another picture where in between these two picture, one picture is fades out and another picture fades in. Dissolve involves gradually changing the visibility of the picture to the other picture. However, rather than transitioning from a shot to a color, a dissolve occurs as a shot changes into another shot gradually.

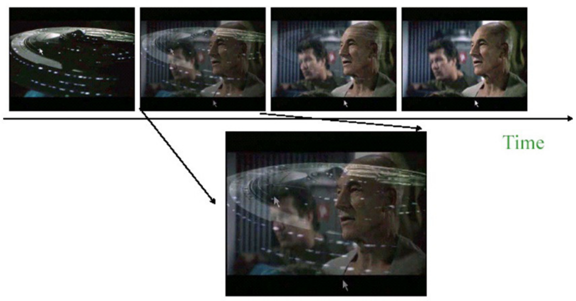

Figure:2 Dissolve Effect

In above figure out of four frames second and third frames shows dissolve effect pattern, in this

IJKRF

second frame is removed and third frame is growing up, but this occurs in more than two frames so it is gradual change.

II. ALGORITHM

1. BLOCK BASED DCT AND DC IMAGE CALCULATION

In detection of sharp and gradual transition like fade and dissolve, first the calculation of DCT of an image is obtained. In this the image (frame) is first divided in the block of size 8*8. After this division in macro block DCT of every block is obtained. So each 8*8 block is having its DCT co-efficient.

The definition of finding DCT of image is shown below that is applied on each 8*8 block of frame.

$$F(x,y) = \frac{C_u}{2}\frac{C_v}{2}\sum_{x=1}^{7}\sum_{y=1}^{7}\cos\frac{(2x+1)u\pi}{16}\sin\frac{(2y+1)v\pi}{16}$$

Where $Cu = \frac{1}{\sqrt{2}}$ if $u = 0$
$= 1$ otherwise

$Cv = \frac{1}{\sqrt{2}}$ if $v = 0$
$= 1$ otherwise

In each block there will be one DC co-efficient and one or more A.C co-efficient. And by applying various formulas abrupt and gradual transitions can be detected. In these algorithms DC image of current frame is calculated and compared with DC image of next frame or frames. DC image of frame can be calculated by.

$$D.C(i,j) = \frac{1}{64}\sum_{x=0}^{7}\sum_{y=0}^{7} DCT(x,y)$$

In above equation the DCT(x,y) shows the 64 DCT values of one 8*8 block. And DC(i,j) shows the each component of DC image.

2. FADE EFFECT DETECION

Now, for detection of gradual transitions like fade, After completing the calculation of DC image, take DC image of two consecutive frames, and perform the following operation for fade effect and this operation is about finding mean of DC image of all frames.

$$Avg(DC\ image) = \frac{\sum DC(i)}{m_1}$$

In this above equation $\sum DC(i)$ is addition of all the values in DC image representation. And m1 represents total number of values in DC image.

3. DISSOLVE EFFECT DETECION

For detection of dissolve effect, apply following operation. This calculation is of variance of DC image and here in this algorithm variation in variance of frames is tracked.

$$Variance(DC\ image) = \bar{m}^2 - \overline{m^2}$$

In the above equation $\bar{m}^2$ represents square of mean of DC image $\overline{m^2}$ represents the mean of squared DC image means first DC image is multiplied with it self and then mean of that squared DC image is taken. Where m_1 is number of values in DC image..

$$m^2 = m \times m$$

$$\bar{m}^2 = \frac{m^2}{m_1}$$

For detection of fade effect, threshold must be nearer to 0 for deciding if there is fade effect or not, several continuous frames are observed, if it has transition from some value to threshold then there is transition and fade is detected.

For detection of dissolve effect concentration is kept on variation in value of variance of frames, calculated from the DC image data. Depending on the threshold value the desired results can be obtained.

When there is gradual transitions in the graph of mean and variance then the detection of fade and dissolve effect is done. In the graph of mean when there is transition from higher to lower value of mean then it is treated as fade out effect, and when there is transition from low value to high value then it is treated as fade in effect. In the case of variance the transition from low to high and high to low both are treated as dissolve effect.

When all the effects are detected, based on the location of cut, fade and dissolve the various shots are detected. Once all the shots are detected then key frames are decided. Here the frames at which shot is changing, key frame is four frames less than shot change frame location.

III. SIMULATION RESULTS

Algorithm for detecting cut, fade, dissolve, key frame extraction, scene change detection and type of scene are applied on various movies and implemented on the platform of MATLAB 7.6.0. In this algorithm one method for each detection is used and obtained various useful results which shows the detection of cut, fade, dissolve, scene change, dialog scene, dance scene and fight scene. These results are as follows.

A. FADE EFFECT DETECTION

JIKRECE

In the figure(3), the fade effect is shown by circles. Here The fade effect is between 1 to 500 and between 1000 to 2500.

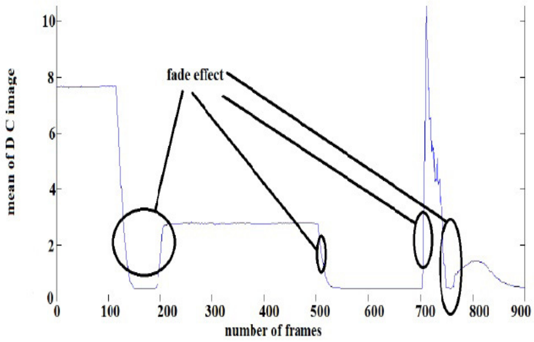

Figure:3 Frames from 1 to 900 for fade detection based mean of DC image of movie

In the figure(4), the fade effect is shown by circles. Here The fade effect is between 1 to 500 and between 1000 to 2500. Also there are various fade effects which are seen by gradual transitions.

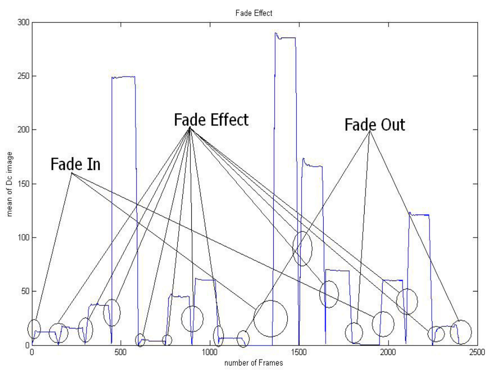

Figure:4 Frames from 1 to 2500 for fade detection based mean of DC image of movie

B. DISSOLVE EFFECT DETECTION

The gradual transitions from low to high and high to low shows the dissolve effect. But these gradual transitions are between more than two frames. Otherwise those transitions are abrupt transitions and called cuts.

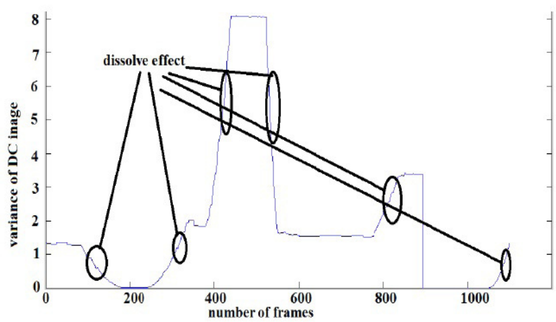

Figure:5 Frames from 1 to 1175 for dissolve effect detection based on variance of DC image of video.

In the figure (5) the dissolve effect is shown by circles. The dissolve effect is between 115 to 160, 250 to 300, 385 to 420 etc..

In the figure (6) the dissolve effect is between 525 to 1500, 2300 to 3000 etc. which are shown by circles in the figure.

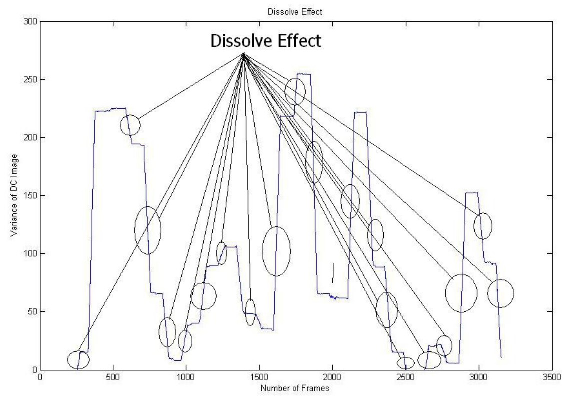

Figure:6 Frames from 1 to 3500 for dissolve effect detection based on variance of DC image of video

IV. QUALITY MEASUREMENT CRITERIA

The quality of performance of this algorithm can be observed from various parameters like precision, recall and F1. Recall is ratio of correct detection to correct and missed detection.

$$Recall = \frac{Correct}{Correct + missed}$$

Precision is ratio of correct detection to correct and false positives detection..

$$Precision = \frac{Correct}{Correct + falsepositive}$$

To find out overall performance of algorithm that combines both recall and precision one most important parameter used is F1 which is defined as follows.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

From the value of F1 one can take the decision about the efficiency of algorithm for particular movie. For detection of cut, fade, dissolve, dialog scene, dance scene, fight scene, one particular method for each detection is applied on various movies.

V. CONCLUSION

As seen from the results the fade in can be seen from the transition from low to high value transition of mean value and the fade out can be seen from the transition from high to low value of transition of mean value • And dissolve effect can be determined by low to high or high to low transition in variance value. In these algorithms the mean and variance of D.C image of frame shows satisfactory results for finding fade and dissolve effect respectively which can be seen by results of F1 which is near by 0.90 for fade effect and 0.75 for dissolve effect. Also use of D.C image concept reduces the processing time because algorithms are operating on the D.C image of frame that is spatially reduced version of original frame.

REFRENCES

1. Scene Detection in Videos Using Shot Clustering and Sequence Alignment Vasileios T.Chasanis, Aristidis C. Likas, and Nikolaos P. Galatsanos, IEEE Transactions On Multimedia, Vol. 11, No. 1, January 2009
2. Vijaykumar Ganapathi, Bhramara Deepthi Chundi, DCT base motion estimation, at http://www1bpt.bridgeport.edu/sed/projects/cs597/Spring2004/v ganapat
3. Rainer Lienhart, Comparison of Automatic Shot Boundary Detection Algorithms, Microcomputer Research Labs, Intel Corporation, SantaClara, CA 95052-8819.
4. Y. Zhuang, Y. Rui, T. S. Huang and S. Mehrotra, Adaptive key frame extraction using unsupervised clustering, in Proc. IEEE Int. Conf. on Image Processing, 1998, pp. 866870.
5. Nilesh V. Patel and Ishwar K. Sethi, Video shot detection and characterization for video databases" Pattern Recognition, vol. 30, no. 4, pp. 583-592, April 1997.
6. Boon-Lock Yeo and Bede Liu, Rapid scene analysis on compressed video", IEEE Trans. on Circuits and Systems for Video Technology,vol.5, no.6, pp.533-544, December 1995.
7. James Monaco, How to Read a Film: The Art, Technology, Language, History, and Theory of Film and Media, Oxford University Press, New York, NY, 1977.
8. F. Idris, S. Panchanathan, Review of image and video indexing techniques, Journal
9. From The Internet Link - http://en.wikipedia.org/wiki/discerete cosinetransform.
10. Costas Cotsaces,Nikos Nikolaidis and Ioannis Pitas,"Video Shot Boundary Detection and Condensed Representation.
11. Lifang Gu, Ken Tsui and David Keightley,"Dissolve Detection in MPEG Compressed Video" 1997 IEEE International Conference on Intelligent Processing Systems October, Beijing. China.
12. Seung-Hoon Han and In-So Kweon,Detecting cuts and dissolves through linear regression analysis,ELECTRONICS LETTERS,October-2003, Vol. 39 No. 22.
13. Video of diss2
14. Video of fade

JKRECE