ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2015.0147

SHOT BOUNDARY DETECTION USING HILBERT TRANSFORM BASED FEATURES

G.G. Lakshmi Priya[1] and S. Domnic[2]

[1]School of Information Technology and Engineering, Vellore Institute of Technology, India
E-mail: gg_lakshmipriya@yahoo.co.in
[2]Department of Computer Applications, National Institute of Technology-Tiruchirappalli, India
E-mail: domnic@nitt.edu

Abstract

The first and foremost step in the shot boundary detection is the extraction of features from the video sequences. To obtain better performance in shot boundary detection, a new method is proposed in this paper, where texture and local binary information extracted from the hilbert transformed frames are processed and represented as features. The similarity between the frames is constructed as continuity signal. The boundaries between the shots are identified by applying the shot transition identification procedure on the continuity signals. The proposed method is evaluated over the TRECVID 2007 SBD dataset and the performance is compared to the top performers of TRECVID 2007 SBD task.

Keywords:

Shot Boundary Detection, Hilbert Transform, Local Binary Information, Texture Feature, Shot Change Transition Identification

1. INTRODUCTION

Multimedia applications have extremely expanded over the past decades. In the new generation of multimedia databases, digital video data are hard to index, browse, search and retrieve due to its abundant availability. Manual annotation of video information is possible but it is a difficult and time-consuming task. Automatic processing of video data motivates the researchers in finding the methodologies that organize and manage video databases.

The basic step for managing the large video databases is to segment the video sequences into shots. The initial step to understand the video is to divide the contents into shots on which analysis is performed. This segmentation process is generally referred to as shot boundary detection [1]. A shot is a sequence of frames generated during a continuous camera operation and represents a continuous action in time and space. Video editing procedures produce abrupt, gradual shot transitions and special effects like zooming and panning.

Among the types of shot transitions considered, (cut, fade, dissolve, wipe), the cut is an immediate change from one shot to another and can be seen as the shortest distance between two shots. There are two types of fades: fade-in and fade-out. A fade-out occurs when the picture information gradually disappears, leaving a blank screen. A fade-in occurs when the picture gradually appears from a blank screen. A fade-in from or fade-out to black is the most common and it is possible to have fade-in or fade-out from any other colour. A dissolve occurs when one whole picture fades away while another whole picture is appearing. A wipe occurs as a line moves across the screen, with the new shot emerging behind the line. A gradual transition occurs over multiple frames and is the product of fade-ins, fade-outs, dissolves or wipes.

In literature, there have been tremendous works [1]-[20] reported on shot boundary detection. Initial research works [5], [9], [18], [20] are mainly on detection of the abrupt shot transitions. The recent works [2], [4], [7], [8], [10]–[15], [17], [21] have been devised toward gradual shot boundary detection. The detection of gradual transition is more difficult when compared to that of abrupt transition. This is because, the difference between the sequences of frames are temporally well separated for cuts, but not for the gradual transitions.

In order to detect shot change in the video sequences almost all shot detection algorithms have reduced the large dimensionality of the video domain by extracting a small number of features from each video frame. These features are extracted either from the whole frames or from a subset of it. Many research works [5], [18] have been proposed to detect cut by making use of features like pixel-wise, color histogram based, gabor filtering etc. However, the algorithms proposed in these works have not been adopted to detect gradual transitions. In addition, features used in paper [5] have certain problems in shot boundary detection. For example, in the histogram-based method if two consecutive frames have quite same histogram while their contents are dissimilar extremely, they may result in missed hit. Nevertheless, it is quite easy to compute and mostly insensitive to translational, rotational and zooming camera motion. For the above reasons it is widely used in many research works [12].

To overcome this problem, other features [5], [13] such as statistical based, motion based, Information theory based features have been used to detect cuts in the video sequences. An algorithm for fade and dissolve detection was proposed by Fernando et al [4] using statistical features of the frames to identify these special effects in uncompressed video. For detection of fade and dissolve, B-spline interpolation curve fitting techniques were used [10]. The authors have made use of ‘goodness of fitting’ to determine the presence of gradual transitions. The work proposed in the paper [13] is based on the Information theory where the mutual information and joint entropy of the transition from frame k to frame k+1 are calculated for R, G, B components. A small value on the mutual information identifies a possible cut and the joint entropy value identifies fade. However, these works detect any two of cut, fade, and dissolve transitions of the video sequences but not all three transitions. To detect both abrupt and gradual transitions, other researchers [2], [11], [14], [15], [20] have used features like edge information, Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), wavelet, multiple features, etc. Edges are invariant to illumination changes and motion in the video sequences. Their main disadvantage is computational cost, noise sensitive and high dimension. Fang et al [14] have used texture as a feature along with histogram and block based approach. They have used fuzzy based technique for cut detection. The work proposed in the paper [15] is based on the variance distribution of edge information in the frame sequence where both fades and dissolves are identified. In this work, the normalized correlation coefficient is used for identifying hard cut. The main drawback of this method is sensitiveness to camera, object motion and extensive content change within the shot. A method [20] that utilizes multiple features for shot boundary detection was proposed, which uses a predefined threshold value for each stage of the detection process. The major drawback of this method is it relies on setting various threshold values throughout the process. As an improvement to the method [20], Lian in his work [21] has used multiple features (pixel wise differences, color histogram, motion) and his method detects cut and gradual transition in serial manner using different threshold values. The number of thresholds used is comparatively less than the previous method [20]. Also, other shot boundary detection methods are illustrated in the TRECVid shot boundary detection reports [19]. Transform coefficients (DFT, DCT, wavelet) are a classic way to describe the texture as feature of the video frames [22]. Their greatest problem is they are generally inconsistent to camera zoom. The methods given above can detect cut as well as gradual transition, but the performance is not fair due to the complex nature of gradual transitions, occurrence of object/ camera motion, illumination.

In order to improve the performance in the video shot detection process, a new method is proposed, which extracts features from Hilbert transformed frames [23]. The experimental results are compared to the top performers of TRECVID 2007 SBD task. The rest of the paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 reports the experimental results in comparison with some of the existing methods. Section 4 provides conclusions and future work.

2. THE PROPOSED METHOD

A new method for video shot boundary detection is proposed using the texture and local binary information. To extract the texture feature, Gray Level Co-occurrence Matrix (GLCM) is constructed from the hilbert transformed frames. The detailed description of Hilbert transform and GLCM is discussed in section 2.1 and 2.2, respectively. The binary information is two level quantization of the transformed frame. Based on the binary information of the blocks, Region of Interest (ROI) between the consecutive frames are identified and the ROI count (ROICount) is calculated as discussed in section 2.4. The texture feature provides the spatial information of the frame, whereas the ROIcount gives the temporal information between the consecutive frames of the video sequence. The combination of the spatial and temporal features is performed and the continuity signal is constructed for shot boundary detection process as discussed in the next sections.

2.1 HILBERT TRANSFORM

Hilbert transform [23] is one of the integral transform like Laplace and Fourier transform. It is named after David Hilbert, who first introduced it to solve a special case of integral equations in the area of mathematical physics. The purpose of this transform is to provide an alternative view of the time-frequency energy paradigm of data. The Hilbert transform $\hat{x}(t)$ of a real time continuous function $x(t)$ is defined for all t by,

$$\hat{x}(t) = H[x(t)] = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(t)}{t-\tau}d\tau \tag{1}$$

Hilbert transform can be motivated in three different ways: the use of Cauchy integral, Fourier transforms in the frequency domain and the phase shift of π/2. The frequency domain (Fourier transform) analysis is easy to calculate compared to that of the Cauchy integral. When an input signal is given, then the Hilbert transform can be computed using Fast Fourier Transform (FFT) as,

$$\hat{x}(t) = H[x(t)] = IFFT[-j\,sgn(\overline{\omega}_n)FFT[x(t)]] \tag{2}$$

where, FFT represents the Fast Fourier transform, IFFT represents the Inverse Fast Fourier transform, represents the nth frequency of the Discrete Fourier Transform (DFT), sgn is the signum function.

2.2 GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)

The texture filter functions provide information about the texture of a frame but fail to provide information about the shape. A statistical method that considers the spatial relationship of pixels is the Gray-Level Co-occurrence Matrix (GLCM), which is also known as the gray-level spatial dependence matrix [24]. GLCM is calculated by finding the frequency of the gray level pixel intensity value i that occurs in a specific spatial relationship to a pixel with the value j. Each element at (i, j) in the resultant GLCM is the sum of the number of times that the pixel with the value i occurred in the specified spatial relationship to a pixel with value j in the input frame. Here, the co-occurrence matrix is computed based on two parameters, which are the relative distance d between the pixel pair (i, j) and their relative orientationθ. d is measured in pixel number. Normally θ, is quantized in four angles (0°, 45°, 90°, 135°). Let $P(i, j, d, \theta)$ represents the GLCM for a frame $i(m,n)$ for distance d and direction θ can be defined as

$$P(i,j,d\theta(= \sum_{p=1}^{m}\sum_{q=1}^{n}\begin{cases}1, if\ I(p,q)=i\ and\ I(p+d\theta_0, q+d\theta_1)=j\\ 0,\ otherwise\end{cases} \tag{3}$$

For a chosen value of distance d, four angular GLCM are considered as $P(i, j, d, 0°)$, $P(i, j, d,$ 45o), $P(i, j, d, 90°)$, $P(i, j, d, 135°)$. For each value θ, its $d\theta_0$ and $d\theta_1$ values are (0, 1) for 0°, (1, 1) for 45°, (1, 0) for 90°, (1, 1) for 135°.

Various texture features [24] can be extracted from GLCM. Among them, four features Contrast (f_1), Correlation (f_2), Energy (f_3) and Homogeneity (f_4) are considered. The contrast feature f_1 measures the intensity contrast between a pixel and its neighbor over the whole image and is calculated using Eq.(4). The range of contrast depends on the size of the matrix. i.e. Range = [0, (size(GLCM, 1) -1)2]. The correlation feature f_2 is a measure of gray tone linear dependency in the image. It measures how a pixel is correlated to its neighbor over the whole image and it is calculated using Eq.(5). Correlation is 1 or -1 for a perfectly positively or negatively correlated image. The energy feature f_3 provides the sum of squared elements in the GLCM. It is also known as uniformity of energy. Energy is calculated using Eq.(6). The homogeneity feature

f_4 measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Homogeneity is 1 for a diagonal GLCM and calculated using Eq.(7).

$$f_1 = \sum_{i,j} |i-j|^2 \, p(i,j) \tag{4}$$

$$f_2 = \sum_{i,j} \frac{(i-\mu\, i)(j-\mu\, j)p(i,j)}{\sigma_i \sigma_j} \tag{5}$$

$$f_3 = \sum_{i,j} p(i,j)^2 \tag{6}$$

$$f_4 = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{7}$$

where, μ is mean, σ is the standard deviation of GLCM (P). However, for a constant frame the texture features are $f_1 = 0$, $f_2 =$ NAN, $f_3 = 1$, $f_4 = 1$. The four features f_1, f_2, f_3, f_4 are the functions of distance and angle. For a chosen distance d, four angular GLCM are measured and hence a set of four values for each of the four features are obtained. On the whole, 16 feature measures are generated. To reduce the number of features, the average of four angular GLCM is taken using Eq.(8) and then the four features are extracted as the Average GLCM (AGLCM).

$$AGLCM = \frac{1}{4} \sum_{\theta=0,45,90,135} p(i,j,d,\theta) \tag{8}$$

2.3 HILBERT TRANSFORMATION OF FRAMES

Most color images and videos are represented in RGB space, which is perhaps the most well-known color space. In order to perform the Hilbert transformation of RGB frames, the following steps are carried out:

1) Initially, Average Color Component of the RGB frame is extracted using Eq.(9)

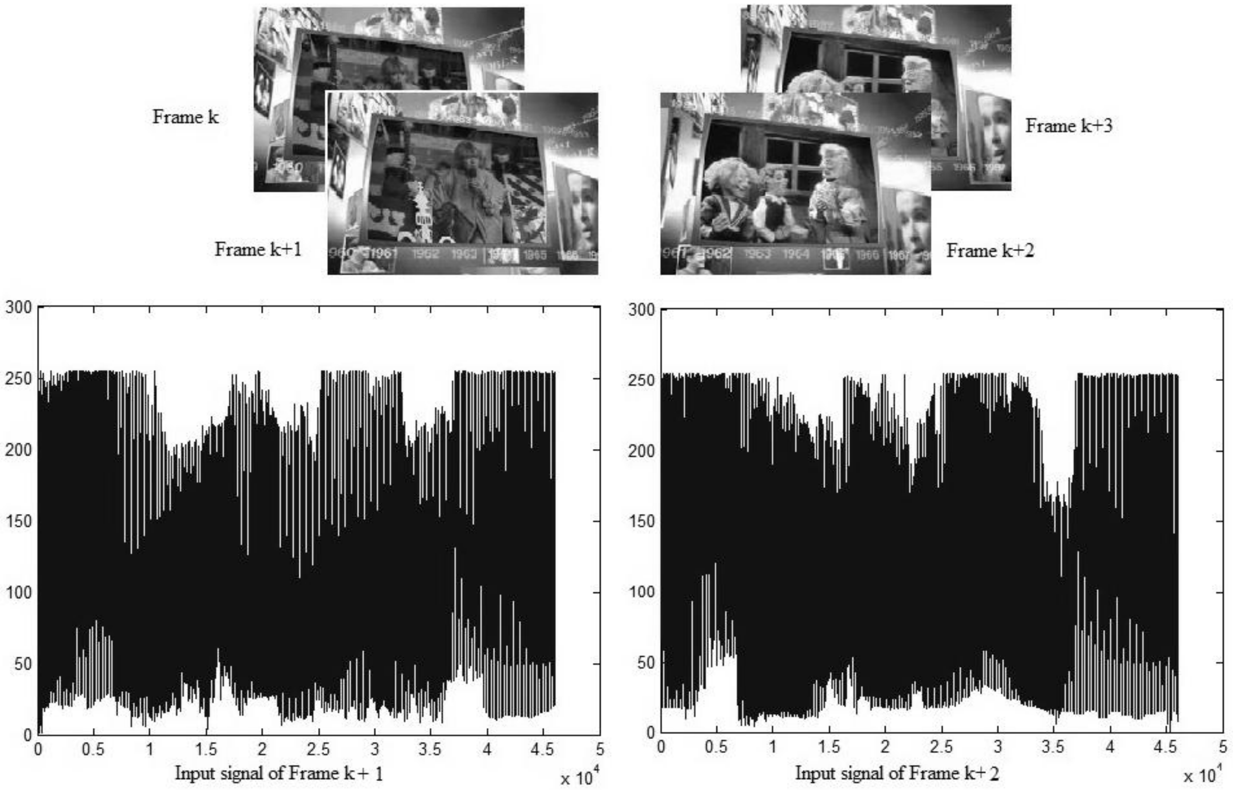
$$\begin{gathered} Acc(i,j) = \big(R(i,j)+G(i,j)+B(i,j)\big)/3 \\ i = 1,2,..M,\ j = 1,2,..N \end{gathered} \tag{9}$$

where, R, G, B are the color components of the frame of size M × N in RGB color space.

2) Perform Hilbert transform of $Acc(i, j)$ as given in Eq.(10) using Eq.(1).

$$\begin{gathered} x(i,j) = Acc(i,j) \\ \hat{x}(i,j) = H[\,x(i,j)\,] \end{gathered} \tag{10}$$

An example is given in Fig.1. In the example, the frame information is converted as input signals and output is the Hilbert transform of the input. The result shown in Fig.1 is output results, using the frequency domain approach of Hilbert transform of a function. After converting the frames into Hilbert transformed frames, the features (ROICount, texture) are extracted.



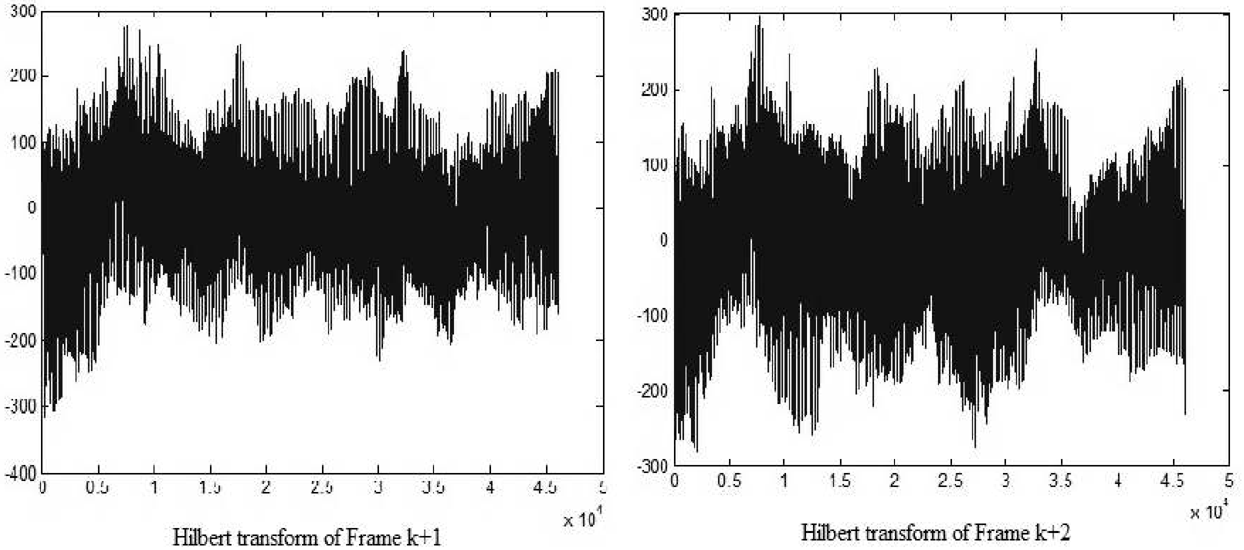


Fig.1. Example for Hilbert transform

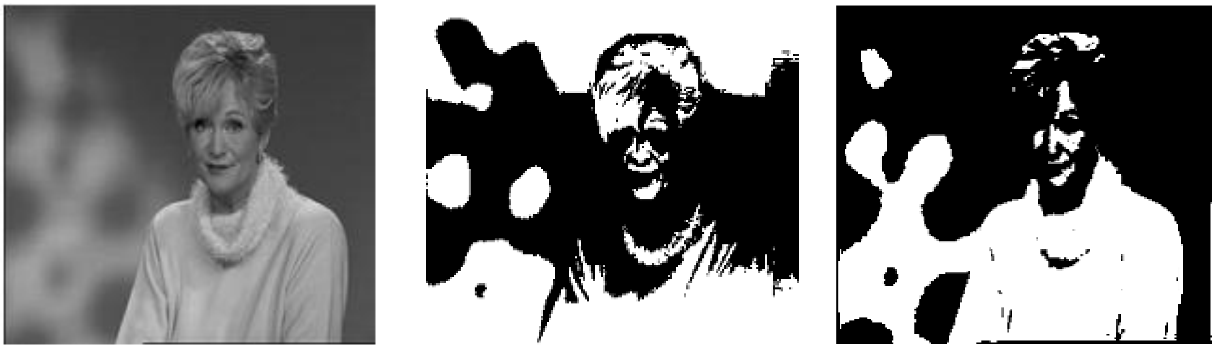


Fig.2(a) Fig.2(b) Fig.2(c)

Fig.2(a). Original frame, (b). Binary frame, (c). Black and white frame

2.4 ROI COUNT FEATURE

In general, video is a sequential collection of frames whose intensity values are represented as a set of real numbers R. More features like color, edge, texture etc., information can be extracted from each frame. Instead of using the whole frames' intensity values as features, transformed information can be considered. In this work, the transformed information is used for feature extraction, which is obtained by using Eq.(1) as a linear filter over the whole video sequences. It can be observed from Fig.1, that the transformed result is the condensed representation of the original information. As per the Hilbert transformation processing, the output oscillates around zero. These transformed frames can be represented as binary representation of 0's and 1's as given in Eq.(11).

$$g(t) = \begin{cases} 1, if \quad \hat{x}(t) > 0 \\ 0, if \quad \hat{x}(t) \leq 0 \end{cases} \tag{11}$$

where, $\hat{x}(i,j)$ is the Hilbert transformed value. The RGB frame is shown in Fig.2(a) and the resultant binary frame $g(t)$ is shown in Fig.2(b), which is entirely different from the black and white frame shown in Fig.2(c). The shot boundary between the frames is identified by calculating the similarity / dissimilarity of the consecutive frames. If the relationship between the frames is more, it means they are similar.

In order to find the ratio of similarity / dissimilarity between the frames, region based differences are calculated. When the frames belong to the same shot, the difference between the regions might be less compared to that of the frame regions from different shots. Those regions having large differences are considered to calculate similarity / dissimilarity between frames. These regions are called as ROI blocks, which may contribute for possible occurrence of transitions in the detection process. After identifying the ROI blocks, the number of ROIblock present in the frame is counted, which represents the dissimilarity between the frames. The steps performed for ROI block selection and ROIcount calculation are given in algorithm 1.

Algorithm 1 Procedure ROIcount(*k*, *k*+1)

1) Divide each binary frame into 4 × 4 non-overlapping blocks ie. blocksize = 16
2) Calculate the number of blocks per frame as

$$numblocks = \frac{m*n}{blocksize}$$

Here $m*n$ is the frame size

3) Perform XOR operation between the corresponding blocks of the consecutive binary frames using Eq.(13)

$$region(i) = g(i,k)\oplus g(i, k+1) \tag{13}$$

where, i =1,…,numblocks, $g(i,k)$ and $g(i, k+1)$ are the values of the i^{th} block of k and $k+1^{th}$ of the resultant binary frames computed using Eq.(11)

4) if region(i) is greater than the threshold TG (half the block size) then

5) region is considered as the Region of Interest (ROI) block.

6) end if

7) $\gamma(k, k+1)$ = Count of the number of ROI blocks

As given in step 3, XOR operation between the blocks of the consecutive binary frames is performed as shown in Fig.3. When the total number of dissimilar values between the blocks of the two consecutive frames is considered, a new frame is constructed using region(i), (i = 1,….,numblock). The number of ROI blocks (ROIcount) will range from 0 to numblocks in newly constructed frame. If there are k frames in the video sequence, then the whole process will be executed for k-1 times.

2.5 TEXTURE FEATURES- GLCM

The next step is to extract texture feature from the Hilbert transformed frames $\hat{x}(i,j)$. GLCM based texture features are extracted from the 16 level quantized Hilbert transformed frame as discussed in section 2.2. A four dimensional vector F = [f_1, f_2, f_3, f_4] for d = 1 is constructed from AGLCM and the resultant F is the texture feature representing each frame. In order to find the relationship between the consecutive frames, dissimilarity / similarity between these frames are to be calculated.

3. CONSTRUCTION OF CONTINUITY SIGNALS

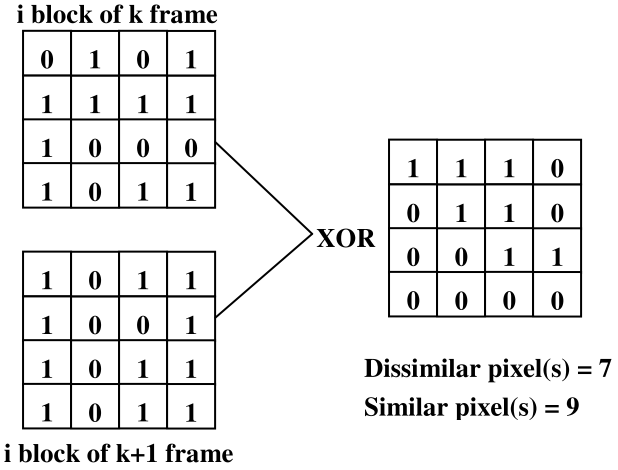
After extracting the features, the next step is construction of continuity signals. To identify whether the shot transition has occurred between the frames, the similarity / dissimilarity between the features extracted from the frames are calculated and represented as continuity signals. The ROICount value is extracted from the two consecutive frames, whereas the texture features F are extracted from the individual frames. So, the texture feature similarity between the consecutive frames is calculated as given in Eq.(14).

$$\beta(k,k+1)=\sum_{i=1}^{4}\left|f_i(k)-f_i(k+1)\right| \tag{14}$$

The continuity signal is constructed by fusing the normalized similarity/dissimilarity of individual feature (ie, ROICount(γ) and F (β)). However, the fusion of features considered in this work does not take place directly. Instead, based on the feature's level of contribution, weights are assigned. The level of the contribution can be identified by finding the significance of each feature. Assigning a proper weight to each feature is a process for estimating how much important the feature are. There are number of methods for feature weighting in machine learning [25]. Weights are assigned to the features using the best feature weighting method [25]. The construction of the continuity signals is performed by combining the features (γ,β) along with its weight is given in Eq.(15).

$$\delta(k,k+1)=\omega_1\gamma(k,k+1)+\omega_2\beta(k,k+1) \tag{15}$$

where, ω_1 and ω_2 are the weights assigned to the features. The resultant value is the continuity signal, which is given as input for shot change identification procedure.



0	1	0	1
1	1	1	1
1	0	0	0
1	0	1	1

1	0	1	1
1	0	0	1
1	0	1	1
1	0	1	1

1	1	1	0
0	1	1	0
0	0	1	1
0	0	0	0

Fig.3. XOR between the blocks of the consecutive frames

4. SHOT CHANGE IDENTIFICATION PROCEDURE

Most of the existing abrupt shot transition detection methods use a threshold parameter to distinguish shot boundaries and changes. The common challenge is the selection of the threshold value for identifying the level of variation, which in turn defines a shot boundary. In this work, threshold th is used for cut detection and is calculated as

$$th=\alpha\mu+\frac{\sigma}{\sqrt{N}} \tag{16}$$

where, μ is the mean of δ, σ is the standard deviation, N is the total number of frames in the video sequence and is the constant. However, the threshold based detection will not suit for gradual transitions, as the ROIcount for these transitions varies gradually and are comparatively less than that of the cut frames. So the frames other than the cut frames are considered for further processing. In general, the duration of the most gradual transitions is more than 1 sec, which means the duration of gradual transition is around 20-45 frames (varies depending on the frame rate/sec). When fast motion of camera/object occurs, the nature of the pattern is slightly similar to that of the gradual transitions and sometimes they are mistaken as shot boundaries. In order to differentiate gradual transition from these effects, separate detectors are employed. Initially, the start and end of the transitions, t_s and t_e respectively are set by considering the ROIcount ie., when the ROIcount starts increasing gradually, the frame k is set as t_s and on reaching the middle of the transition, the count value starts decreasing till it reaches t_e, the end of the transition.

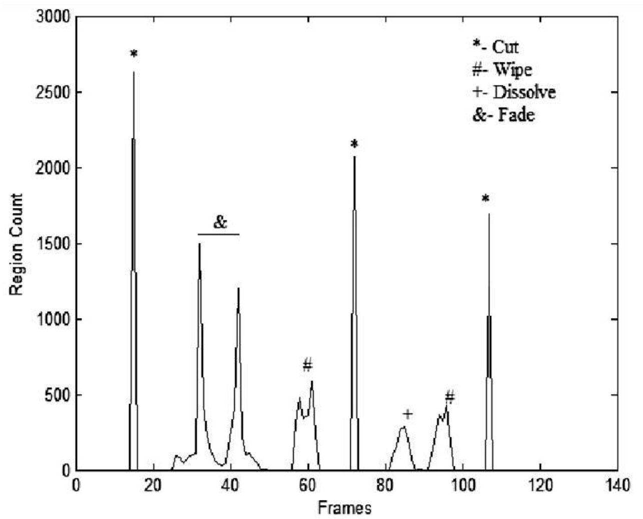


Fig.4(a)

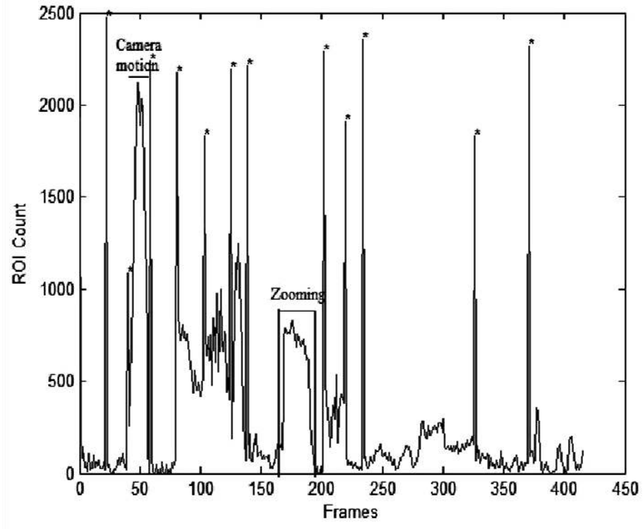


Fig.4(b)

Fig.4. Patterns of ROICounts for Abrupt and gradual transitions

As per Fig.4(a), pattern generated for fade and wipe has two peaks. Therefore, additional condition is required to check this situation. The abrupt and gradual transitions like fades, dissolves and wipes are identified using the identification procedure discussed in algorithm 2. Peaks of the continuity values are obtained by using the peak finding procedure as given below:

Algorithm 2 Shot Change Identification Procedure

Peak count: number of peaks between ts and te, framecount: number of frames between the first and second peaks, maxROI: maximum of ROIcount between the region

```
Calculate the threshold th using Eq.(16) and find peaks of $\delta$ using Peak Finding Procedure.
If peakvalues > th then
    the corresponding frame $k$+1 are declared as the cut frame.
else
    Identify the start $t_s$ and $t_e$ as discussed in section 4
    If($t_e$-$t_s$) > 20 & < 45 then
        If(peakcount = 1 & mean(ROICount[(ts) : (te)]) < ½th) then
            the region is declare as the dissolve transition region
        else if (peakcount = 2 & framecount between peaks > 5 and <15) then
            If (maxROI < th) then
                Declare as fade region
            else if (maxROI < ½th) then
                Declare as wipe region
            endif
        endif
    endif
endif
If the frames does not satisfy any of the conditions, then those frames are declared as no Transition frames
```

Peak Finding Procedure:

For each continuity value δ(K), K=1,..,Frames
If (δ(K)- δ(K-1))>½th&&((δ(K)- δ(K+1))> ½th
then peak δ = (δ(K)

5. EXPERIMENTAL RESULTS AND DISCUSSION

To examine the effectiveness of the proposed methods and to reveal their advantages over the state of the art methods, experiments are carried over the benchmark dataset and evaluated using performance evaluation criteria.

5.1 DATASET DESCRIPTION

In early years, due to the lack of large annotated video collections, the SBD methods were evaluated on a relatively small dataset. From 2001, the National Institute of Standards and Technology (NIST) has started a benchmark of content based video retrieval ie TRECVID [26], where SBD is one of the evaluation tasks. Till 2007, various SBD systems are proposed by the participants and the results are evaluated [26].

To enable comparison with TRECVID 2007 shot boundary detection techniques, the proposed method was tested on the TRECVID 2007 SBD evaluation data [26] which contains 17 sequences, totally about 7 hours containing both abrupt and gradual transitions. There are both color and black / white videos in these sequences. The description of the test videos are given in Table.1 and its ground-truth are taken from TRECVID website. Also, the proposed shot boundary detection system is tested with some sequences (listed in Table.2) taken from TRECVID 2001 dataset [27].

Table.1. Description of TRECVID 2007 SBD Task Dataset

Number of Videos	17
Video duration	About 7 hours
Frame Rate	25 fps
Resolution	352 × 288 pixels
Total frames	637,805
Total transitions	2,463

Cuts	2,236 (90.8%)
Dissolves	134 (5.4%)
Fade Out/In	2 (< 0.1%)
Other	91 (3.7%)

More specifically, five videos are selected for comparison because these videos are relatively complex and the referred systems have reported performance on each of these videos separately. The global methods to measure the performance of the shot boundary detection algorithm are Precision (P), Recall (R) and F1-Score (F1). The precision and recall are calculated using the following equations.

$$Precision(P) = \frac{No.of\ transitions\ correctly\ reported}{No.of\ transitions\ reported} \times 100$$

$$Recall(R) = \frac{No.of\ transitions\ correctly\ reported}{No.of\ transitions\ in\ reference} \times 100 \quad (17)$$

Recall is the rate of misclassification and it is high, when the number of misclassification is low. Precision reveals the false alarm, lesser the false hits results in higher precision. To rank the performance of different algorithms, F1-score is used, where it is harmonic average of Recall and Precision as shown in Eq.(18).

$$F1 - Score = \frac{2.P.R}{(P+R)} \quad (18)$$

The higher these ratios are, the better is the performance of the shot boundary detection process. The performance of the proposed method is compared with the top performers of the TRECVID 2007 SBD task participants as given in Table.3 with precision, recall and F1-score. It is observed from Table.3 that the proposed work yields F1-score 91.7% for overall transition, while considering gradual transitions, the proposed works F1-score is 63.5%, which is comparatively better than the other two methods. On the whole, the proposed work yields better result for gradual transition compared to the third ranking participants [31] of TRECVID 2007 SBD task.

Also, in order to reveal the efficiency of the proposed method, comparison with few related recent methods like edge oriented method [33], Color layout descriptor [34] are carried out. The best results of the above specified existing methods are considered and are shown in Table.4 for the video sequences listed in Table.2. It can be observed from Table.4, that the proposed work yields F1-score 89% for overall transition which is comparatively better than that of the existing related works.

Computational time for the proposed method, which includes Hilbert transform of the frame, constructing new frame from the transformed frame and feature extraction are listed in Table.5, where the speed is in milliseconds per frame. In general, shot boundary detection process is the basic step for many applications. However, the computation time for the whole detection process depends on the feature extraction. The proposed feature extraction method is easy to implement and the computation time for feature extraction listed in Table.5 is more convincing.

Table.2. Description of Videos from TRECVID 2001 SBD Task Dataset

Video sequence	Duration in secs	Frames	Abrupt transition	Gradual transition	Overall transition
Anni005	195	5655	27	11	38
Anni006	553	16037	47	27	74
Anni009	145	4205	18	20	38
Bor08	180	5335	22	13	35
Nad53	195	5803	17	31	48
Total	1268	37035	131	102	233

Table.3. Comparison of Proposed Method with Various Existing Methods

Transition	Proposed Method			Karlsruhe at TRECVID 2007 [32]			BRAD at TRECVID 2007 [31]		
	P	R	F1	P	R	F1	P	R	F1
Abrupt	91.8	91.8	91.8	94.0	93.8	93.9	98.2	97.3	97.7
Gradual	68.1	60.2	63.5	44.2	20.4	27.9	42.5	58.7	49.3
Overall	93.7	89.7	91.7	92.0	87.6	89.7	91.9	94.1	92.9

Table.4. Comparison of Overall Transitions' F1-Score

Video Sequences	Proposed method			Edge oriented method [33]			Color Layout descriptor [34]		
	P	R	F1	P	R	F1	P	R	F1
Anni005	81	92	86	87	91	89	77	89	83
Anni006	81	91	85	82	89	85	83	85	84
Anni009	85	93	89	87	93	90	79	89	84
Bor08	91	91	91	86	91	88	93	89	91
Nad53	87	96	91	81	97	88	85	84	85

Average	85	93	89	85	92	88	83	87	85

Table.5. Computation Time for the Proposed Feature Extraction Method

Sl. No.	Stages	Time (secs)
1	Hilbert transformation	0.00884
2	New frame construction	0.01298
3	Feature extraction	0.00488

6. CONCLUSION

In this paper, a new shot boundary detection method is presented, which uses ROICount and GLCM texture features extracted from Hilbert transformed frames. Experimental results are evaluated over TRECVID 2007 SBD dataset and publicly available dataset in terms of precision, recall and F1-score. The proposed method yields better results in gradual transition compared to the top performers of the TRECVID 2007 SBD task. When the results are compared with recent works, the proposed work yields better recall and F1-score for over all transition. The proposed algorithm is successful in detecting the gradual transition and in reducing the false rate caused by the object and camera motion compared to the other related works. However, the proposed method has lower performance, when gradual transition occurs simultaneously along with object and camera motion and addressing these issues is our future work.

REFERENCES

[1] H. J. Zhang, A. Kankanhalli and S. W. Smoliar, "Automatic partitioning of full-motion video", *Multimedia Systems*, Vol. 1, No. 1, pp. 10-28, 1993.

[2] R. Zabih, J. Miller and K. Mai, "A feature-based algorithm for detecting cuts and classifying scene breaks", *Proceedings of the ACM Multimedia*, pp. 189-200, 1995.

[3] S. Pfeiffer, R. Lienhart, G. Kuhne and W. Effelsberg, "The MoCA Project-Movie Content Analysis Research at the University of Mannheim", *Informatik98*, pp. 329-338, 1998.

[4] W. A. C. Fernando, C. N. Canagarajah and D. R. Bull, "Fade and Dissolve Detection in Uncompressed and Compressed Video Sequences", *Proceedings of International Conference on Image Processing*, Vol. 3, pp. 299-303, 1999.

[5] R. Lienhart, "Comparison of automatic shot boundary detection algorithms", *Proceedings of IS & T / SPIE Storage and Retrieval for Image and Video Databases VII*, Vol. 3656, pp. 290-301, 1999.

[6] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 2, pp. 90-105, 2002.

[7] Y. Tan, J. Nagamani and H. Lu, "Modified Kolmogorov-Smirnov metric for shot boundary detection", *Electronics Letters*, Vol. 39, No. 18, pp. 1313-1315, 2003.

[8] Eyas El-Qawasmeh, "Scene change Detection schemes for video indexing in uncompressed domain", *Informatica*, Vol. 14, No. 1, pp. 9-36, 2003.

[9] A. Whitehead, P. Bose and R. Laganiere, "Feature based cut detection with automatic threshold selection", *Proceedings of the Content Based Image and Video Retrieval*, 2004.

[10] Jeho Nam and Ahmed H. Tewfik, "Detection of Gradual Transitions in Video Sequences Using B-Spline Interpolation", *IEEE Transactions on Multimedia*, Vol. 7, No. 4, pp. 667-679, 2005.

[11] H. Lu and Y. Tan, "An effective post-refinement method for shot boundary detection", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 11, pp. 1407-1421, 2005.

[12] R. Joyce and B. Liu, "Temporal segmentation of video using frame and histogram space", *IEEE Transactions on Multimedia*, Vol. 8, No. 1, pp. 130-140, 2006.

[13] Z. Cernekova, I. Pitas and C. Nikou, "Information theory based shot cut/ fade detection and video summarization", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 1, pp. 82-91, 2006.

[14] H. Fang, J. Jiang and Y. Feng, "A fuzzy logic approach for detection of video shot boundaries", *Pattern Recognition*, Vol. 39, No. 11, pp. 2092-2100, 2006.

[15] H. W. Yoo, H. J. Ryoo and D. S. Jang, "Gradual shot boundary detection using localized edge blocks", *Multimedia Tools and Applications*, Vol. 28, No. 3, pp. 283-300, 2006.

[16] Y. Kawai, H. Sumiyosh and N. Yagi, "Shot Boundary Detection at TRECVID 2007", *TRECVID 2007 Workshop*, 2007.

[17] Costas Cotsaces, Zuzana Cernekova, Nikos Nikolaidis and Ioannis Pitas, "*Color Image Processing Methods and Applications- Color based video shot boundary detection*", CRC Press, pp. 526-548, Ch. 23, 2007.

[18] Tudor Barbu, "A novel automatic video cut detection techniques using Gabor filtering", *Computer and Electrical Engineering*, Vol. 35, pp. 712-721, 2009.

[19] A. F. Smeaton, P. Over and A. R. Doherty, "Video shot boundary detection: Seven year of TRECVid activity", *Computer Vision and Image Understanding*, Vol. 114, No. 4, pp. 411-418, 2010.

[20] Yoshihiko Kawai, Hideki Sumiyoshi and Nobuyuki Yagi, "Shot Boundary Detection at TRECVID", *Systems and Computers in Japan*, Vol. 38, No. 13, pp. 1-14, 2007.

[21] Shiguo Lian, "Automatic video temporal segmentation based on multiple features", *Soft Computing*, Vol. 15, pp. 469-482, 2011.

[22] K. McDonald and A. F. Smeaton, "A comparison of score, rank and probability- based fusion methods for video shot retrieval", *International Conference on Image and Video Retrieval*, W-K Leow et al. (Eds.), *Lecturer Notes in Computer Science*, *Springer,* Vol. 3568, pp. 61-70, 2005.

[23] Ahmed O. Abdul Salam, "Hilbert transform in image processing", *Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 111-113, 1999.

[24] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification", *IEEE Transactions on Systems Man Cybernetics*, Vol. SMC-3, pp. 610-621, 1973.

[25] C. H. Lee, F. Gutierrez and D. Dou, "Calculating feature weights in naive bayes with kullback-leibler measure", *Proceedings of IEEE 11th International Conference on Data Mining*, pp. 1146-1151, 2011.

[26] NIST Agency, "*TREC Video Retrieval Evaluation: TRECVID*", TRECVID Dataset website, Available at: http://trecvid.nist.gov/.

[27] Interaction Design Laboratory, "The Open Video Project", Available at: http://www.open-video.org/.

[28] Zhu Liu, Eric Zavesky, David Gibbon, Behzad Shahraray and Patrick Haffner, "AT & T Research at TRECVID 2007", *TRECVID 2007 Workshop*, 2007.

[29] Jinhui Yuan, et. al., "THU and ICRC at TRECVID 2007", *TRECVID 2007 Workshop*, 2007.

[30] Markus Mhling, Ralph Ewerth, Thilo Stadelmann, Christian Zfel, Bing Shi and Bernd Freisleben, "University of Marburg at TRECVID 2007: Shot Boundary Detection and High Level Feature Extraction", *TRECVID 2007 Workshop*, 2007.

[31] J. Ren, J. Jiang and J. Chen, "Determination of Shot Boundary in MPEG Videos for TRECVID 2007", *University of Bradford, TRECVID 2007 Workshop*, 2007.

[32] H. K. Ekenel, M. Fischer, H. Gao, K. Kilgour, J. S. Marcos and R. Stiefelhagen, "University at Karlsruhe (TH) at TRECVID 2007", *TRECVID 2007 Workshop*, 2007.

[33] Don Adjeroh, M. C. Lee, N. Banda and Uma Kandaswamy, "Adaptive Edge-Oriented Shot Boundary Detection", *EURASIP Journal on Image and Video Processing*, pp. 1-13, 2009.

[34] D. Borth, A. Ulges, C. Schulze and T. M. Breuel, "Keyframe Extraction for Video Tagging Summarization", *Proceedings of Informatiktage*, pp. 45-48, 2008.