Multimed Tools Appl (2016) 75:93–113
DOI 10.1007/s11042-014-2273-y

A shot detection technique using linear regression of shot transition pattern

Debabrata Dutta · Sanjoy Kumar Saha · Bhabatosh Chanda

Received: 6 July 2013 / Revised: 4 September 2014 / Accepted: 8 September 2014 /
Published online: 23 September 2014

Abstract Video segmentation acts as the fundamental step for various applications like, archiving, content based retrieval, copy detection and summarization of video data. Shot detection is first level of segmentation. In this work, a shot detection methodology is presented that evolves around a simple shot transition model based on the similarity of the frames with respect to a reference frame. Frames in an individual shot are very similar in terms of their visual content. Whenever a shot transition occurs a change in similarity values appears. For an abrupt transition, the rate of change is very high, while for gradual it is not so apparent. To overcome the effect of noise in similarity values, line is fit over a small window using a linear regression. Thus slope of this line exhibits the underlying pattern of transition. A novel algorithm for shot detection, hence, is developed based on the variation pattern of the similarity values of the frames with respect to a reference frame. First an algorithm is proposed, which is direct descendant of the underlying transition model and applies a threshold on the similarity values to detect the transitions. Then this algorithm is improved by utilizing the slope of linear approximation of variation in similarity values rather than the absolute values, following least square regression. Threshold on the slope is determined with a bias towards minimizing false rejection rate at the cost of false acceptance rate. Finally, a simple post-processing technique is adopted to reduce the false detection. Experiment is done with the video sequences taken from *TRECVID 2001* database, action type movie video, recorded sports and news video. Comparison with few other systems indicates that the performance of the proposed scheme is quite satisfactory.

D. Dutta
Tirthapati Institution, Kolkata, India
e-mail: debabratadutta2u@gmail.com

S. K. Saha (✉)
CSE Department, Jadavpur University, Kolkata, India
e-mail: sks_ju@yahoo.co.in

B. Chanda
ECS Unit, Indian Statistical Institute, Kolkata, India
e-mail: chanda@isical.ac.in

Keywords Video segmentation · Shot detection · Shot transition

1 Introduction

With the rapid advancement of multimedia technology, it has become easier to generate, capture, store, edit and distribute video data. As a result applications like archiving,content based retrieval, copy detection and summarization of video data have emerged as an active area of research. In all such applications, it is important to segment data into meaningful units. Thus, video segmentation becomes the fundamental step. It may be categorized as *temporal* and semantic segmentation. At the lowest level video data is a collection of frames. The temporal unit, i.e., *shot* is a collection of consecutive frames captured in a single session (between camera on and off) of camera operation. Collection of semantically related consecutive shots depicting an event or a part of the story is taken as a *scene* which is a semantic unit. Temporal segmentation i.e. the shot detection is the first level of segmentation. In the applications like content based retrieval or copy detection, a query sequence is submitted which needs to be matched with the video data present in the database. As the different shots in the query sequence may find match with the shots of different sequences in the database, such applications require a simple and low cost shot detection scheme to enable shot level matching. This observation motivates us for the present work.

1.1 Past work

Shot detection stands for determining the boundary frames i.e. in case of a transition from one shot to another the last frame of the previous shot and the first frame of the following shot are to be identified. Shot transition can be broadly categorized as *abrupt* and *gradual*. Abrupt transition or cut occurs more often and is the outcome of natural process where one is appended to another. Thus, transition between the shots is limited within two consecutive frames. Gradual transition is the result of post-shooting process. Unlike cut, it has certain duration. *Dissolve*, *fade-out*, *fade-in* etc. fall into this category. Lot of research has been carried out to detect the shot boundaries. In case of cut, the transition occurs between two consecutive frames which are visually very dissimilar and hence it is easier. The detection of gradual transition is more challenging.

Shot detection algorithms are mostly focused on finding out the visual discontinuity between the frames. For abrupt change, such discontinuity can be readily detected by looking into two successive frames whereas it is not so prominent for a gradual transition unless the frames being compared are temporally well separated. The major tasks of a shot detection algorithm includes representing a frame in terms of visual descriptors, and discontinuity detection based on a similarity measure or a transition model. Smeaton [23] has presented a comprehensive description of different aspects of shot boundary detection methodology including the evaluation policies and measures adopted in TRECVID.

In order to describe the frame-content, variety of features have been used. Pixel-level difference of consecutive frames [9, 10, 15], gray-level or colour histogram [8–10] are quite simple and are widely used. Study presented in [23] also indicates the wide use of colour histogram as the descriptor. Based on the joint histogram of consecutive frames, mutual information is computed [12]. The pixel difference is sensitive to object motions. On the other hand, histograms are global in nature and lacks spatial information. Motion based features have also been tried [2, 11, 19]. Many researchers have dealt with edge and gradient based features [1, 12, 15, 25].

Content of the frames in a video are represented by the feature vector and suitable similarity measures are applied on the feature vectors to compute the similarity between the frames. As discussed in [23], different distance measures like Euclidean distance, Manhattan distance, histogram intersection have been used by the researchers. In order to detect the shot boundaries, the common practice is to compute the change in visual content and to check whether it exceeds a threshold or not [4, 9, 21]. Though the approach is simple but the selection of threshold is crucial. Multiple thresholds (one for abrupt change and another for gradual changes) have also been considered [6, 13]. For detecting gradual transition, use of sliding window is quite common [9, 12, 13]. But the success depends on the proper selection of window size. Le et al. [14], in their approach mapped the task to the problem of text segmentation in natural language processing. Amiri et al. [3] have presented a generalized eigenvalue decomposition based system. Use of various machine learning techniques is also quite common [23]. Decision tree based classification [20], KNN [8], fuzzy clustering [11], hidden Markov model [28], neural network [18], SVM [15, 24] are few such examples. All such techniques have their own merits and demerits in terms of complexity, tuning of various parameters, proper training etc.

A very few works have relied on model to describe the frames, and have detected the boundary frames based on that model. Shot transition model has been presented in [10] based on the post production editing process. In [17], cut and gradual transitions are modeled independently as delta function and rectangular function respectively. Temporal statistics based model [16] and background similarity based model [7] are also tried. In [18], a unified model has been presented to handle both abrupt and gradual transitions. Based on the model, a frame estimation scheme is formulated. Boundary is detected based on model parameters and frame estimation errors. Finally, frames are classified using multi-layer perceptron network.

It is evident from the past study that wide variety of approaches have been tried by the researchers. Most of the aforesaid methods either are too simple to handle complex transition or computationally too expensive. Moreover, many of the methods need nontrivial parameters (e.g., transition threshold or moving widow size) to be supplied by the users. In many of the cases the algorithms are developed for a specific (e.g. cut) type of transition. In this work a simple but novel scheme is presented to detect the shot boundaries. Proposed scheme relies on a shot transition model which assumes that frames in a shot are visually very similar with respect to a reference frame, and the discontinuity in similarity value occurs at a shot transition. Two algorithms are presented of which the first one is a direct implementation of the underlying model. The second and the final one is an improvement over the first and it evolves around the linear approximation of the similarity values. Slope of this linear approximation finally indicates whether there is a transition or not. The organization of the rest of the paper is as follows. Proposed methodology is presented in Section 2. Section 3 contains the experimental result and the paper is concluded in Section 4.

2 Proposed methodology

Proposed methodology draws the motivation from the window based scheme that is generally intended for detecting gradual transition. The simplest technique for detecting cut/abrupt change is based on the difference between the visual descriptors of two consecutive frames. In case of cut, the difference is quite high. It is not so for a gradual transition where two consecutive frames may not differ much. Thus, it is difficult to identify such frames. To address this problem, concept of window is introduced. A sliding window of size N is considered, and a frame is compared with its N-th predecessor. The temporal separation of the gradual transition frames with respect to the reference one makes the difference more significant in comparison to the frames within a shot. Determining the size of window as well as the threshold for detecting the transitions become very crucial.

2.1 Transition model

For easy understanding of the transition model, let us for the time being consider that a video clip consists of only one transition which may be either abrupt or gradual and such a sequence is taken as the window. We will see later (in Section 2.3) that the proposed algorithm can extend the transition model to work with large video.

Frames in the window are represented by suitable descriptors. As the frames in a shot are very cohesive in nature, all the frames of first shot in the window will possess high similarity with the reference frame, which may be the first frame of the window. However, difference between the frames are not zero because of the object motion, occlusion etc. By designing the frame descriptors properly, such effect may be kept very low. To make the reference descriptor more robust to noise and local variations, average of the first K descriptors in the window is taken as the descriptor corresponding to the reference frame. In our experiment, K is taken as five. On the other hand, frames of the next shot will usually exhibit very low similarity with the reference frame. If the similarity values are plotted against the frame numbers, an abrupt change from high to low similarity as shown in Fig. 1a is expected. For gradual transition, the similarity value slowly decreases from high to low as shown in Fig. 1b. Frames in gradual transition phase bears the trace of both the shots – the previous one and the following one. The impact of previous one gradually decreases and that of the next one increases. The task is to identify the boundary frames i.e. the last frame of current shot or the first one of the next shot. Based on the said model, the boundaries can be identified by determining the suitable threshold values denoting high and low similarity.

In case of a conventional window based scheme a frame is compared with N-th predecessor. The difference between the similarity values obtained for the frames in the shot and for those in the gradual transition phase may not be significant and it depends heavily on N. In the proposed scheme the reference frame is temporally well separated (separated by the whole shot) from the frames in transition phase. Thereby, the similarity values of the frames in transition phase will be considerably different in comparison to the frames in the first shot in the window. As a result, selection of proper threshold value is easier.

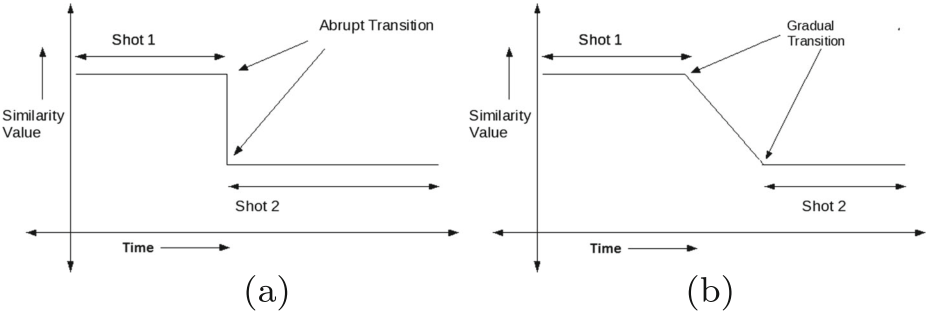


Fig. 1 Transition model: **a** Abrupt change or cut and **b** Gradual transition

2.2 Computation of features

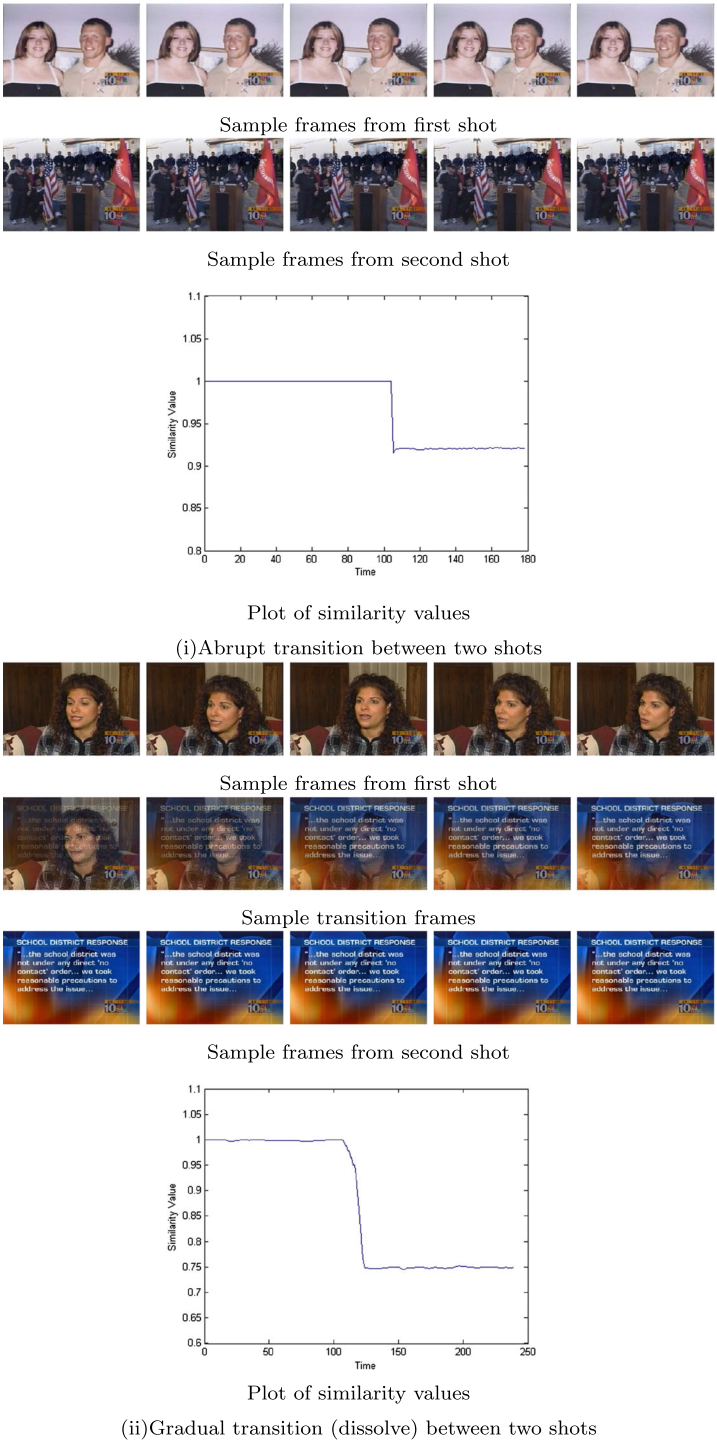
Features to represent the frame-content should be such that minor changes in the content would have marginal impact on the descriptor. It is desired that despite of changes present in the frames within a shot the computed feature values for them are very close to each other. A global feature can serve this purpose well. Another requirement is that the feature must have enough discriminative power to distinguish the *within shot* frames and the *transition* frames in case of gradual transition. As gradual transition takes place over a sequence of frames, such transition frames are also very similar to each other. In this context, a feature, which is global in nature, may fail to provide values to distinguish between *within-shot* and *transition* frames. On the other hand, local features may be very sensitive and for the frames within a shot show significantly different values. Thus, the features are to be chosen carefully to fulfill these diverging requirements.

In our work, we have relied on edge based features. Frame image is first converted to gray-scale, and intensity gradient at each pixel is then computed using Sobel operator. The gradient image thus formed is then thresholded to obtain the edge image. Average gradient is taken as the threshold value, and the edges thus obtained are illumination invariant. During gradual transition, current frame bears the impact of both the previous shot and the next shot. Edges from previous frame gradually fade away and that from next one emphasizes. We intend to capture this phenomena. So the edge image is divided into N_p partitions of equal size. The collection of normalized count of edge pixels in the partitions is taken as the descriptor. Thus, a N_p-dimensional feature vector $< v_1, v_2, \ldots, v_{N_p} >$ is obtained where v_i stands for normalized count of edge pixels in the i-th partition and $\sum_i v_i = 1$. The feature vector reflects the spatial distribution of the edge pixels. For the frames within a shot, edge point distribution remains almost unaltered over the frames unless N_p is quite high. But, for the transition frames the distribution varies significantly as new edges are gradually emphasized and the old one weakens. Thus, appearing/vanishing edges during gradual transition make the distribution quite dynamic which is almost static for frames within a shot. Thus, the computed feature vector meets the diverging requirement. It may be noted that high value of N_p will make the descriptor too sensitive, while a low value for the same will reduce its discriminating power. Hence, a moderate value is chosen to serve the purpose and in our experiment it is taken as 16.

2.3 Boundary detection

Boundary detection process is based on the model presented in Section 2.1. As discussed, we assume that the model considers a window consisting of the frames of two consecutive shots. Thus, there exists only one instance of shot transition. Few examples of sequences with only one transition and corresponding plots of the similarity values of the frames with respect to the reference frame in the sequence are shown in Fig. 2. In order to highlight the characteristics of the transition, the plots are restricted to a subset of transition frames and some preceding and following frames. It is evident that the plots follow the model we propose.

In reality, video sequences are large enough consisting of number of shots. If such a sequence is considered as the window and initial five frame of the sequence acts as the basis of reference frame for the rest then it is more likely that the similarity pattern reflected by the transition frames gets deviated from the model. A few examples of video sequences with multiple transitions and corresponding plots of the similarity values are shown in Fig. 3. Such a sequence contains multiple transitions. First one maintains the desired pattern but



Sample frames from first shot

Sample frames from second shot

Plot of similarity values

(i)Abrupt transition between two shots

Sample frames from first shot

Sample transition frames

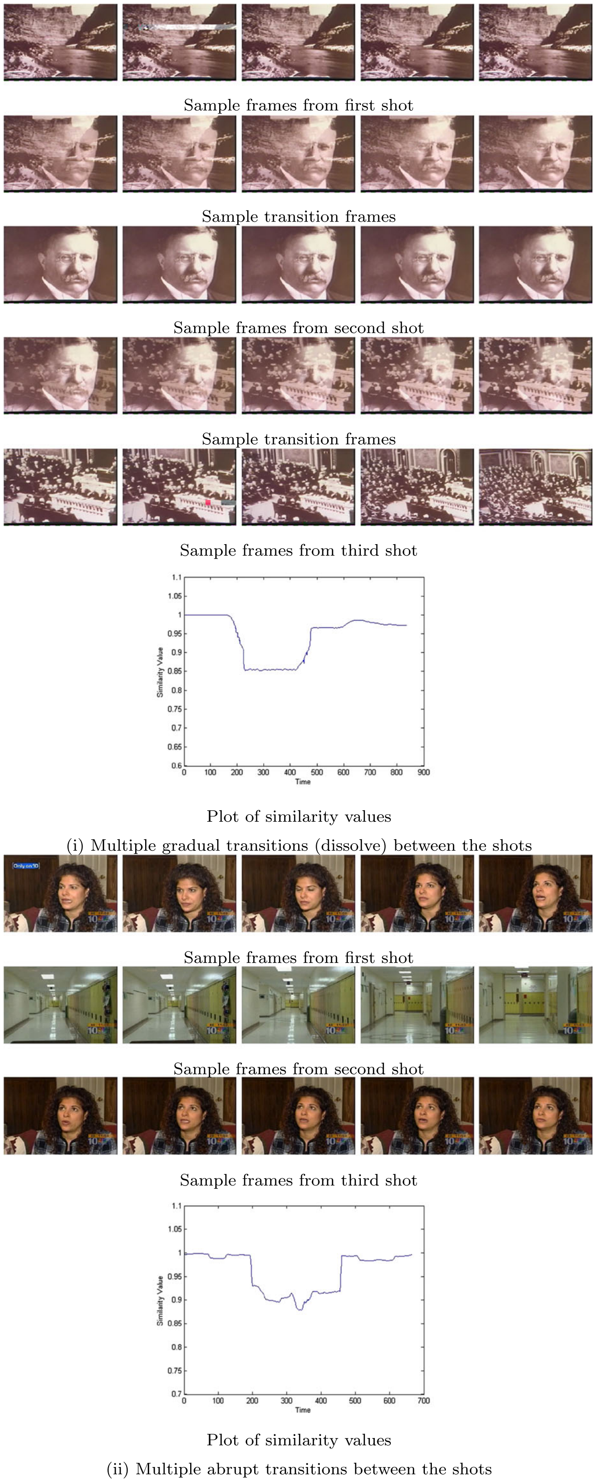
Sample frames from second shot

Plot of similarity values

(ii)Gradual transition (dissolve) between two shots

Fig. 2 Sequences with single transition – (i) abrupt transition and (ii) gradual transition

Fig. 3 Sequences with multiple transitions of different type – (i) multiple gradual transitions and (ii) multiple abrupt transitions



Sample frames from first shot

Sample transition frames

Sample frames from second shot

Sample transition frames

Sample frames from third shot

Plot of similarity values

(i) Multiple gradual transitions (dissolve) between the shots

Sample frames from first shot

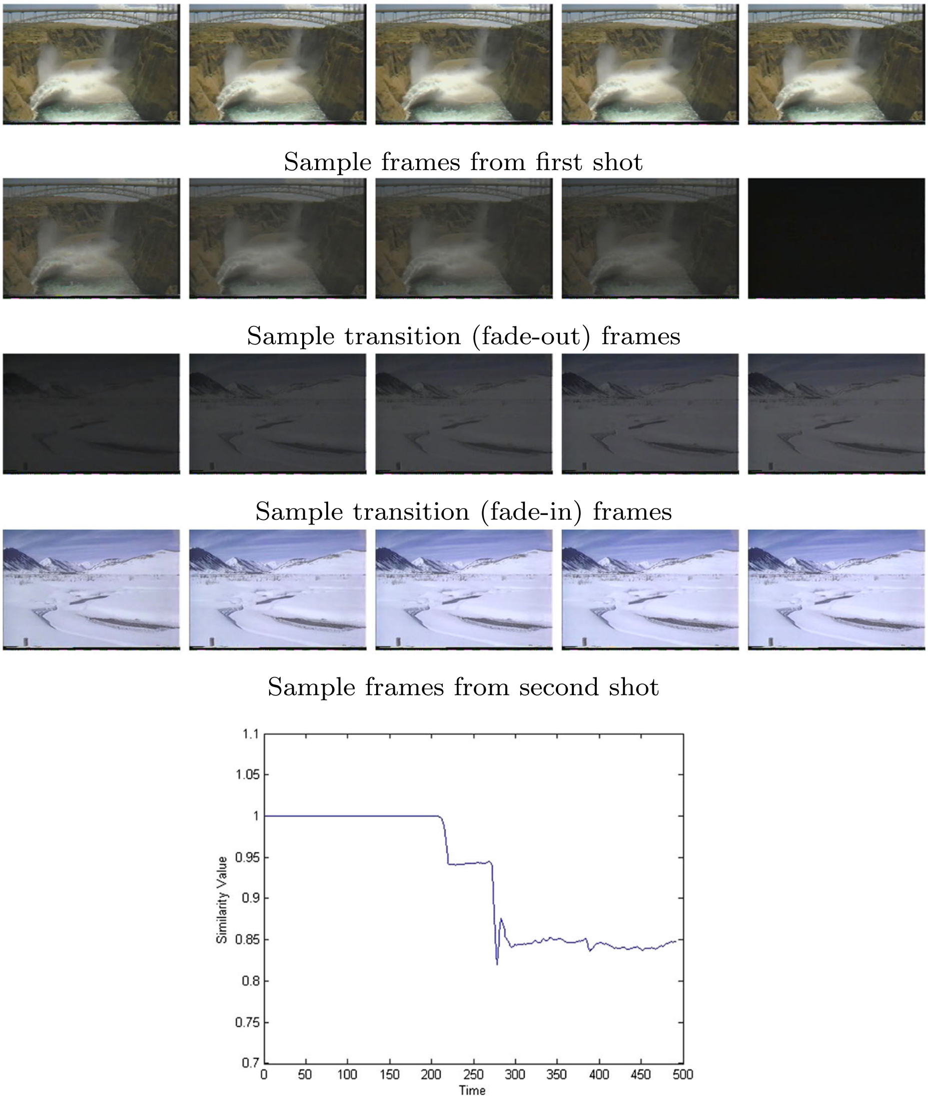
Sample frames from second shot

Sample frames from third shot

Plot of similarity values

(ii) Multiple abrupt transitions between the shots the subsequent transitions may fail to comply the model. Fade-out (fade-in) is also a kind of gradual transition where brightness of frames gradually lessens (becomes more) to a dark (bright) frame. The plots of similarity values as shown in Fig. 4 is apparently quite different from the assumed model, but the algorithms show that they work with such sequences and utilizes the transition characteristics reflected by the model.

Shot boundary detection algorithm starts with a window with the initial N consecutive frames from the video data. N is taken as any value in $[l_{min}, l_{max}]$ where l_{min} and l_{max}


Sample frames from first shot

Sample transition (fade-out) frames

Sample transition (fade-in) frames

Sample frames from second shot

Plot of similarity values

Fig. 4 A sequence with gradual transition (fade-in and fade-out)

denote the estimates for minimum and maximum shot length respectively. Thus, selecting a value for N is not very critical. Following the assumed model, the algorithm tries to identify the first occurrence of shot transition within the window and also the corresponding transition frames. Once those are identified, set of frames in the window prior to the transition phase is taken as a shot. Window is refreshed to have the frames following the transition phase from the current set and also sufficient frames from the original video are augmented to make the window N frames long. The algorithm proceeds with the updated window. In case N is smaller than the length of the shot which is not known apriori, the algorithm is likely to fail in detecting a transition. To come out of it, subsequent N frames from the original video are appended to the current content in the window to double its length and pass it to the process for the next iteration of shot detection algorithm. Thus, the proposed algorithm is built around the transition model that runs iteratively to handle the large sequence.

Consider, a video sequence vid that needs to be segmented. Each frame in the video sequence is represented by the edge based descriptor as described in Section 2.2. Similarity of each frame with the reference descriptor in the window is computed by using a suitable distance measure. In this work, we have adopted the Bhattacharyya distance [5] between the normalized descriptors of corresponding frames. The overall shot detection algorithm, $segment\_video(vid)$ may be described as follows.

```
$segment\_video(vid)$
begin
$S$=first $N$ frames from $vid$
while (all frames in $vid$ are not tested)
    begin
        Compute similarity array $sim$ for $S$
        Smooth $sim$
        $chk = verify\_uniformity(sim)$
        if ($chk$) then $S = S \cup$ {subsequent $N$ frames from $vid$}
        else
            begin
                $t = get\_transition\_sequence(S, sim)$
                $shot$=sequence of frames in $S$ preceding $t$
                if (#($shot$) $< l_{min}$) then consider it as transition
                $seq$=sequence of frames in $S$ following $t$
                if (#($seq$)$>$ 0) then
                $S = seq \cup$ {subsequent ($N$-#($seq$)) frames from $vid$}
                else $S = S \cup$ {subsequent $N$ frames from $vid$ }
            end
        end
end
```

This algorithm may also be referred to as Algorithm 1.

$sim[i]$ denotes the similarity between the reference descriptor and the descriptor of i-th frame in the window. Due to change in photometric condition in the environment, frames with similar content may have different visual appearance. Moreover, descriptors of within shot frames may also reflect small variation. To combat the problem, similarity values are smoothened through mean filtering. Filtering is done by considering a moving window of size 5. $verify\_uniformity()$ checks whether the given set of frames in the window is a part of single shot or not. It returns true if the sequence is uniform or homogeneous i.e. the sequence should not be segmented further else false is returned. This is essential to avoid over splitting of the sequence. Temporally separated frames in a shot may also differ due to various reasons like motion of the objects and occlusion. In the feature space also, it may be reflected to some extent despite of the care taken to design the descriptors. But, such variation is usually quite low. Based on this assumption, $verify\_uniformity(sim)$ works as follows.

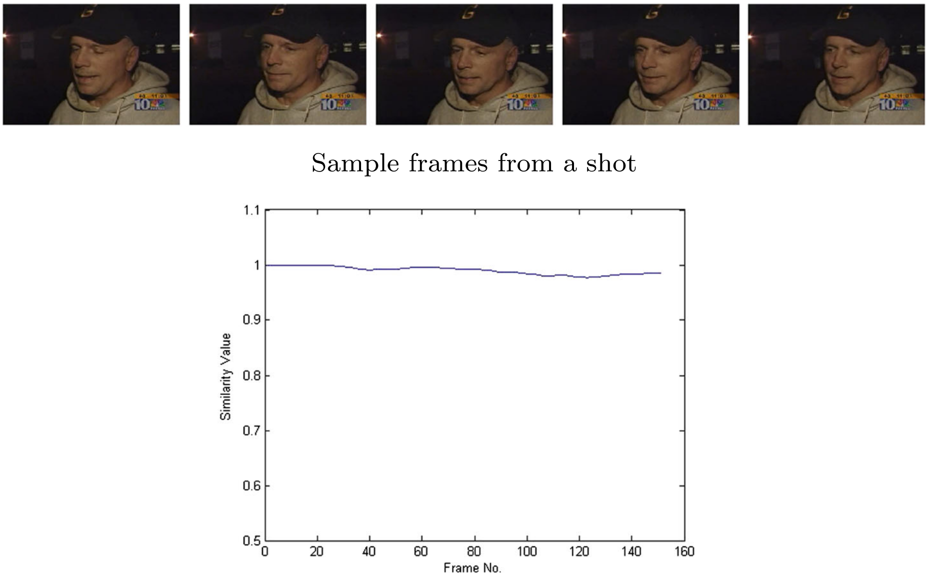
```
$verify\_uniformity(sim)$
begin
    $m_1 = max\{sim[i]\}$
    $m_2 = min\{sim[i]\}$
    if $\frac{m_2}{m_1} > th$ then return $true$ else return $false$
end
```

A very high value for th may declare a shot as non-uniform one resulting into over splitting. On the other hand a low value may fail to discriminate subsequent shots. The value of th is chosen experimentally. The procedure exploits the fact that the variation in the similarity values (elements in sim) is quite low within a shot. It is well reflected in the example shown in Fig. 5.

Once a sequence is considered as a non-uniform one i.e. it contains multiple shots, the task then becomes to detect the transition frames. The function $get\_transition\_sequence(S, sim)$ does this job as follows.

```
$get\_transition\_sequence(S, sim)$
begin
    t={} and $i = 1$
    while ($sim[i] >= high$) {increment $i$}
    $t = t \cup frame_i$
    while ($sim[i] > low$ and $sim[i] < high$){
        $t = t \cup frame_i$
        Increment $i$}
end
```

The frames in the initial shot should show high similarity value, while the frames of the following shots are expected to have low similarity value. For the frames in gradual transition, the similarity values are supposed to be in between. The set t holds the transition frames. For an abrupt change, according to the procedure described above, t is supposed to have only one frame. But, because of smoothing operation on sim, such transition may be blurred and may give rise to multiple transition frames. It also affects the localization of gradual transitions. However, for the applications like content based retrieval system, such compromise does not affect the final result of the application. The $high$ and low values are taken as $\mu + k_1\sigma$ and $\mu - k_1\sigma$ respectively, where μ and σ stand for mean and standard deviation of similarity values respectively and value of k_1 is chosen experimentally. It is assumed that a shot consists of at least a minimum number of frames, l_{min} and a shot of smaller length is discarded and taken as transition. Part of the gradual transition frames may be identified



Sample frames from a shot

Plot of similarity values

Fig. 5 A sequence with single shot

as shot giving rise to false boundary. However, by discarding the shots of smaller length ($< l_{min}$) the cases of such false detection can be minimized. In our experiment we have chosen l_{min} as 30. It may be noted that content of the window (i.e. S) is changed after each iteration. Frames representing the first shot and the first transition are removed from S. Subsequent set of frames from video sequence get appended to S. Under two circumstances, S is updated in a different manner. First case arises when S is uniform and no transition is identified. Second case corresponds to the situation when a shot and following transition is successfully detected and there is no subsequent frame. In both the cases, next set of frames from video data is appended to the current content of S. With the new window S, next iteration proceeds afresh. The similarity array, sim has to be recomputed once the reference frame is changed in any situation. Otherwise, partial computation is required for only the newly appended frames in S.

The algorithm $segment\_video()$ works with large video sequence by taking a part of it in each iteration. At a point of time, the window may have multiple transitions. Few cases are shown in Figs. 3 and 4 which reveal that the similarity values may not always reflect the pattern assumed in the model. Moreover, $high$ and low thresholds are chosen directly from the similarity values. Thus, the presence of number of shots in the window can have significant impact on the pattern of similarity values and consequently threshold values. As a result, performance may suffer. It has motivated us to modify the algorithm to enable the easy selection of threshold and handling of the window with multiple transitions in an elegant way.

The modified algorithm relies on the fact that the frames in a shot are almost similar in terms of feature values. Hence, with respect to a common reference frame, their similarity values are also very close to each other. If straight lines are fitted over similarity values of a subset of contiguous frames then such lines are almost horizontal for the frames within a shot. On the other hand, line segments correspond to transition (abrupt/gradual) regions have higher slope. Thus, thresholding slope magnitude the shot boundaries are detected. The steps of $modified\_segment\_video(vid)$ algorithm are summarized as follows.

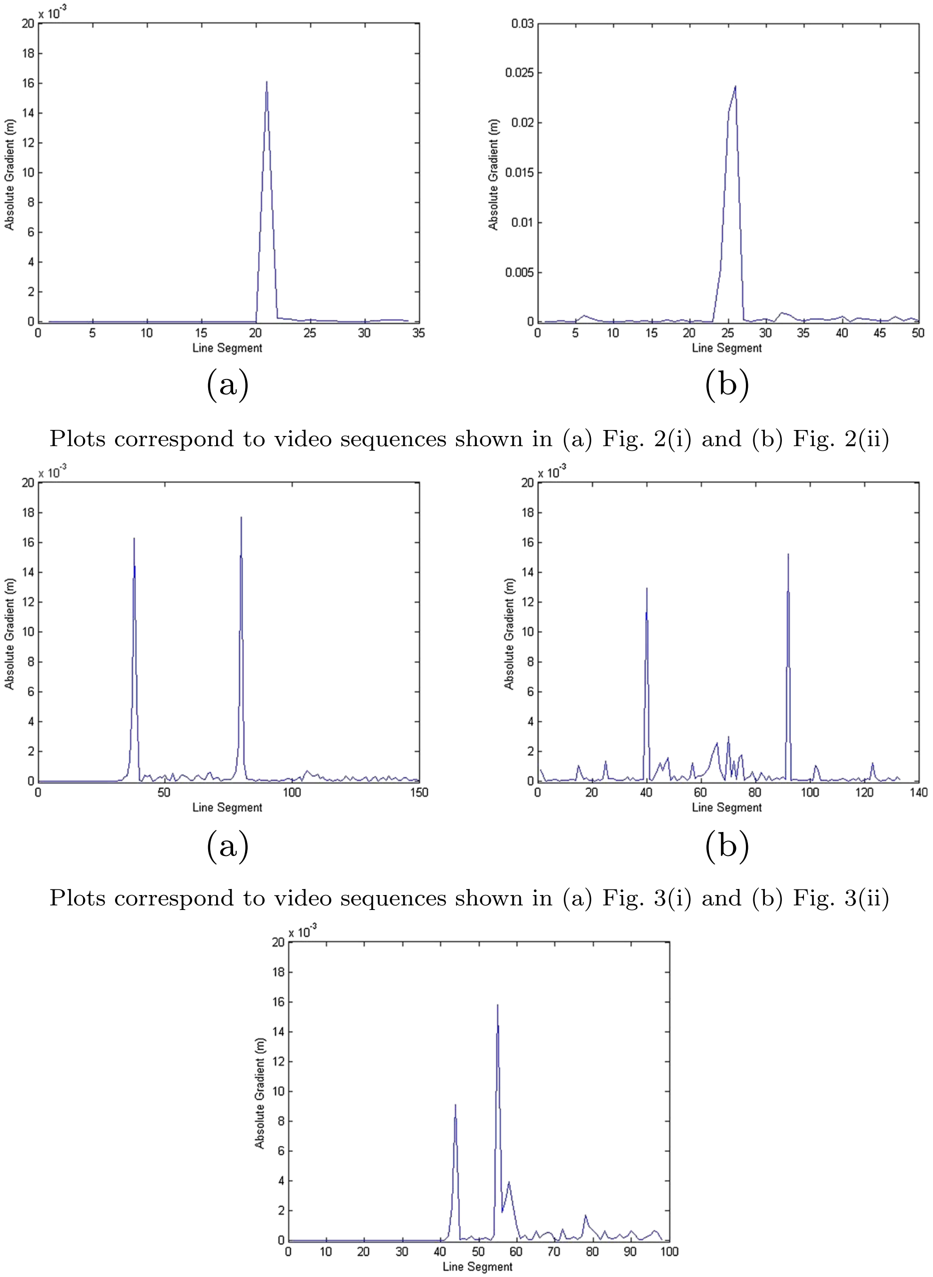
```
$modified\_segment\_video(vid)$
begin
    $S$=first $N$ frames from $vid$
    while (all the frames in $vid$ are not tested)
        begin
            Compute similarity array $sim$ for $S$
            Smooth $sim$
            $chk = verify\_uniformity(sim)$
            if ($chk$) then $S = S \cup$ {subsequent $N$ frames from $vid$}
            else begin
                for each frame $f_i \in S$ {
                    Form $subset_i = sim[j]$ where $j \in [i-k, i+k]$
                    Over $subset_i$, fit a straight line following least square regression
                    Store the magnitude of the slope of the line segments
                    in an array $sim\_slope$ }
                Mark the frames $f_j$ with $sim\_slope[j] > th$ as transition frames
                Detect the shots as set of frames between two consecutive transition
                Shots of length less than $l_{min}$ are taken as transition sequence
                $S = S$ - {set of frames in S prior to last detected shot}
                $S = S \cup$ {subsequent $N$ frames from $vid$}

            end
        end
end
```

This algorithm may also be referred to as Algorithm 2. Frames corresponding to the line segments with high magnitude of slope are taken as the transition frames. It may be noted that multiple transitions and hence multiple shots present in a window can be detected. In $segment\_video()$ algorithm, window for the next iteration starts with the frames following the first transition. In the modified algorithm $modified\_segment\_video()$, next iterations proceeds from the beginning of the last detected shot. For a gradual transition, it may so happen that all the consecutive line segments passing through the transition regions are not marked as boundary. To address the problem any shot of length smaller than l_{min} is considered as transition sub-sequence Fig. 6 shows the plot of the slope magnitude for different video sequences in Figs. 2, 3 and 4. It is evident that slope of relatively higher magnitude corresponds to transition zone in the video sequence. Threshold selection for slope is much easier compared to selection of $high$ and low thresholds for similarity value. In our experiment, th is taken as $\mu_s + k_2\sigma_s$ whereμ_s and σ_s denote average and standard deviation of the values in $slope$. The value of k_2 is determined experimentally.

2.4 Post-processing

The visual content within a shot may vary significantly due to the reasons like camera and/or object motion, presence of high activity. Such variation particularly in comparison to the initial part of the shot may result into oversplitting. To minimize the false boundary detection a post-processing step is proposed here.



(a)

(b)

Plots correspond to video sequences shown in (a) Fig. 2(i) and (b) Fig. 2(ii)

(a)

(b)

Plots correspond to video sequences shown in (a) Fig. 3(i) and (b) Fig. 3(ii)

Plot corresponds to video sequence shown in Fig. 4

Fig. 6 Plot of slope magnitude corresponding to video sequences in Figs. 2, 4 and 3

Let S_t be a detected transition sequence. Usually for abrupt transition $| S_t |= 2$, otherwise it is > 2. f_s and f_l are the first and last frame in S_t respectively. f_s (resp. f_l) is estimated from f_{l+1} (resp. f_{s-1}) to obtain f_{e_s} (resp. f_{e_l}). If either f_s is highly similar with

f_{e_s} or f_l is highly similar with f_{e_l} then S_t is taken as false transition Thus, the steps are follows.

```
begin
$f_{e_s}$ =estimate of $f_s$ based on $f_{l+1}$
$f_{e_l}$ =estimate of $f_l$ based on $f_{s-1}$
$sim_s$ =similarity between $f_s$ and $f_{e_s}$
$sim_l$=similarity between $f_l$ and $f_{e_l}$
if ($sim_s \geq th_{pp}$ or $sim_l \geq th_{pp}$) then
    $S_t$ is a false transition
end
```

In order to estimate f_s from f_{l+1}, f_s is first divided into blocks of size 16×16. Let b_i is a block of f_s centered at (x, y). A match for it is searched in a window of size 33×33 centered at (x, y) in f_{l+1}. Matching is based on the sum of absolute difference of grayscale intensity. The best matched block from the search window is the estimate for b_i in f_{e_s}. Following the similar process f_{e_l} is formed. To generate the feature vector for the frames f_s, f_{e_s} and f_{e_l}, corresponding frame is divided into 16 blocks. For each block 256 bin intensity histogram is formed. All such histograms are concatenated in raster scan order and normalized to form the descriptor. Histogram intersection between the descriptors provides the similarity. Value of th_{pp} is experimentally determined as 0.8.

3 Experimental results

In our experiment we have used a dataset consisting of various sequences taken from *TRECVID 2001* and *2005* test databases as well as some recorded sports and news video sequences from TV and Internet. The recorded sequences are made available at http://www.isical.ac.in/~ecsu/. Three movies of different types are also included in the dataset. Detailed description of the dataset is provided in Table 1. It reflects wide variety containing both types of transitions – abrupt and gradual change. Gradual change includes dissolve, fade-in and fade-out. The dataset has been grountruthed manually to note down the start and end frame numbers of each shot. For abrupt transition, last frame of pre-transition shot and first frame of post-transition shot are marked as transition frames. For gradual transitions, a set of consecutive frames are marked as transition. However, to evaluate the detected boundaries arising out of computational methods a relaxation of ±5 frames is considered for gradual transition. Moreover, a gradual transition of smaller length ($\leq$5 frames), if detected as abrupt one is also considered as correct.

A video sequence is provided as the input to shot detection algorithm. Identified transition frames are compared with the corresponding groundtruth to identify the miss and false boundary detection. The performance is measured in terms of Recall, Precision and F-measure. These are computed as follows.

$$Recall(R) = \frac{Correct}{Correct + Miss}$$

$$Precision(P) = \frac{Correct}{Correct + False}$$

$$F - measure(F) = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Table 1 Description of the dataset

Sequence name	Source	Sequence type	# of shots	# of abrupt transition	# of gradual transition
anni001	Trecvid2001	Documentary	09	00	08
bor01_003	Trecvid2001	Documentary	07	00	06
bor01_010	Trecvid2001	Documentary	10	01	08
bor03	Trecvid2001	Documentary	64	57	06
bor10_001	Trecvid2001	Documentary	13	00	12
bor10_005	Trecvid2001	Documentary	21	00	20
bor10_011	Trecvid2001	Documentary	11	00	10
bor19	Trecvid2001	Documentary	39	00	38
ugs01_004	Trecvid2001	Documentary	21	17	03
ugs01_006	Trecvid2001	Documentary	15	08	06
arb1	Trecvid2005	News	15	14	00
arb3	Trecvid2005	News	41	40	00
chn1	Trecvid2005	News	76	58	17
chn2	Trecvid2005	News	94	91	02
chn4	Trecvid2005	News	72	64	07
eng1	Trecvid2005	News	39	32	06
eng2	Trecvid2005	News	30	25	04
eng3	Trecvid2005	News	44	29	14
001_news	Recorded	News	54	50	03
004_news	Recorded	News	25	22	02
001_sports	Recorded	Sports	22	21	00
002_sports	Recorded	Sports	30	29	00
Mission Impossible-I	Movie CD	Action movie	1463	1437	25
A Beautiful Mind	Movie CD	Romantic movie	1470	1438	31
Hirok Rajar Deshe	Movie CD	Dialogue based and musical action movie	1331	1327	03
Total			5016	4760	231

To obtain the measures in percentage, 100 is multiplied with them. In case of a shot boundary detection algorithm, it is desired to have high recall (i.e. low rate of miss in detecting a boundary) and high precision (i.e. low rate of false boundary detection).

The experiment has been conducted following both the algorithms – $segment\_video()$ (Algorithm 1) and $modified\_segment\_video()$ (Algorithm 2). A moderate value for N is chosen to work with long sequences and it is taken as 800. Both the algorithms first verify whether the given sequence is a homogeneous one (i.e. consisting of one shot only) or not. If it is identified as a non-homogeneous one then the shot boundaries present in the sequence are detected. $verify\_uniformity()$ carries out the test based on a threshold value th. For different values of th, the performance of $verify\_uniformity()$ is shown in Table 2. It is observed that for higher value of th, uniform sequence is splitted into sub-sequences whereas for lower value, non-uniform sequences may get considered as uniform. Analysing the experimental result, th is taken as 0.9 for subsequent experiments.

Table 2 Performance of sequence uniformity test for different threshold values

Sequence	# of	% of correct identification			
type	sequence	$th = .80$	$th = .85$	$th = .90$	$th = .95$
Sequence with no transition	75	98.67	96.00	94.67	78.67
Sequence with transition	75	76.00	84.00	96.00	98.67
Overall	150	87.33	90.00	95.33	88.67

Algorithm 1 considers the frames with similarity value lying between $\mu + k_1\sigma$ and $\mu - k_1\sigma$ are the transition frame(s). In order to choose the optimal value of k_1, an experiment has been carried by varying its value for a number of sequences with abrupt and/or gradual transition. For tuning the parameter, sequences consisting of single transition are taken as input. Higher the value of k_1, more frames may tend to qualify as transition frames and it leads to higher recall with lower precision. To judge the diverging requirement of high recall and high precision, performance is measured in terms of F-measure. The result for different values of k_1 is shown in Table 3. Based on this table the value of k_1 is taken as 1.

In Algorithm 2, lines with slope greater than $\mu_s + k_2\sigma_s$ qualify for transition regions. Value of k_2 is determined following the similar experiment as it is done in case of k_1. Table 4 shows the performance of the algorithm for different values of k_2. Higher the value of k_2 the probability of miss in detection increases but higher precision is achieved. For low values of k_2, recall increases but false detection also rises. Hence, F-measure is considered to determine the suitable value of k_2 and it is found to be 1.5.

In order to study the performance of Algorithms 1 and 2, all the video sequences in the described dataset is used as the input with related parameter values. The experiment is thus conducted in a single run. Each sequence as a whole is provided as input to the algorithms. Table 5 shows the performance of the two algorithms. Algorithm 1 works based on the similarity value whereas Algorithm 2 (without post-processing) is the modified version that works based on the slope of line segment fitted on the overlapped subset of similarity values. It is evident from Table 5 that Algorithm 2 (without post-processing) performs better than Algorithm 1 in terms of both precision and recall as both false detection and miss detection are less for the second algorithm. It has been noted that still the false detections are large enough for sequences with high action where content varies significantly within a shot. Mostly, it gives rise to unwanted detection of abrupt detection. In order to minimize such false detection, post-processing is applied on the output of Algorithm 2 and the performance as shown in Table 5 improves significantly.

In order to compare the performance of the proposed system (Algorithm 2 – with post-processing), we have implemented the system proposed by Zhang et al. [27] and

Table 3 Performance of Algorithm 1 for different values of k_1

Sequence	# of	F-measure in %		
type	sequence	$k_1 = 0.5$	$k_1 = 1.0$	$k_1 = 1.5$
With abrupt transition	90	73.40	77.72	73.91
With gradual transition	75	68.53	74.61	73.86

Table 4 Performance of Algorithm 2 for different values of k_2

Sequence	# of	F-measure in %		
type	sequence	$k_2 = 1.0$	$k_2 = 1.5$	$k_2 = 2.0$
With abrupt transition	90	92.71	95.03	94.63
With gradual transition	75	83.04	89.66	88.72

experimented with the dataset described in Table 1. The system in [27] deals with intensity histograms of R, G and B channels. Each frame is divided into number of blocks and histogram of each block are concatenated. Based on Fisher criterion in linear discriminant analysis a continuity measure is considered to detect the shot boundaries which are classified as abrupt and gradual transition using support vector machine. The scheme is simple enough but suffers from large number of false detection. We have also implemented the system of Tsinghua University [26] which is top performer as described in [23]. It has three components like fade-out/fade-in (FOI) detector, cut detector and gradual transition (GT) detector. Colour histrogram, pixel-wise difference feature, average and standard deviation of pixel intensities and motion vectors are used as the descriptors for visual content. FOI and cut detectors are simple and GT detector is based on finite state automata model. The scheme considers number of thresholds which are critical. It provides high recall for cut but precision suffers due to false detection. Comparative results shown in Table 6 indicates that the proposed methodology performs much better. It may be noted that for the proposed system, an interval for precision, recall and F-measures are provided. To obtain such intervals, number of runs are conducted. 90 % transitions are randomly selected from each video and used as the input in each run. In this way twenty runs are conducted. Finally, the intervals for the parameters are computed using Z-score based technique [22] with 95% confidence level. It is clear that even the lower bounds for the parameters are much higher than the value of corresponding parameter for other systems.

For comparison, we have also worked with a common set of video sequences as used in [1] and these are taken from *TRECVID 2001* dataset. System of Adjeroh et al. [1] follows an adaptive approach based on edge maps and it outperformed number of systems as presented in their work. Result on the same dataset is also presented in [18]. As the results of number of systems are already available, for comparison we have applied the proposed methodology (Algorithm 2 – with post-processing) and also the schemes presented in [26, 27] on the same dataset. The figures of performance metrics for different systems are

Table 5 Performance measure of proposed methodologies (Algorithm 1 and 2) (all figures in scale 0–1)

Transition	# of	Algorithm 1			Algorithm 2					
type	transition				Without post-proc.			With post-proc.		
		P	R	F	P	R	F	P	R	F
Abrupt	4760	0.71	0.91	0.80	0.83	0.96	0.89	0.95	0.96	0.95
Gradual	231	0.68	0.74	0.71	0.90	0.94	0.92	0.93	0.94	0.93
Overall	4991	0.71	0.90	0.79	0.84	0.96	0.90	0.95	0.96	0.95

Table 6 Comparison of shot boundary detection performance (all figures in scale 0–1) on the dataset described in Table 1

Transition type	# of transition	System in [26]			System in [27]			Proposed system (confidence level 95%)		
		P	R	F	P	R	F	P	R	F
Abrupt	4760	0.80	0.95	0.87	0.83	0.88	0.85	0.96±.006	0.98±.006	0.97±.006
Gradual	231	0.66	0.64	0.65	0.62	0.70	0.66	0.91±.017	0.96±.020	0.93±.012
Overall	4991	0.79	0.94	0.86	0.82	0.87	0.84	0.96±.008	0.98±.006	0.97±.006

presented in Table 7. As in case of Table 6, here also an interval for F-measure is provided for the proposed system and it is computed in the similar manner as discussed earlier. It shows that the proposed system performs better than the rest. Among rest, the performance of the unified model based system in [18] is closest to that of the proposed system. The said system [18] estimates a frame from its predecessor and follower using histogram, edge and motion based features. Finally, Multilayer Perceptron based neural network is used for classification of the frames and shots are detected by applying post-processing on classifier output. Although it performs better than the proposed method but the strength of the proposed method lies in its simplicity and low cost. Apart from other overheads, the computational cost of motion based local features used in [18] is of $o(w^2W^2)$ where size of the blocks to be matched is $w \times w$ and search region is of size $W \times W$. Such costly matching has to be carried out for number of blocks and that too for each frame. Similar cost is also incurred in post-processing step of the proposed system. But, it has to carried out only for limited number of frames (at the most two frames per transition). On the other hand in the proposed system, cost for feature computation is $o(n)$ where n is the number of pixels in a frame. In case of machine learning technique based boundary detection, the major hurdle lies in proper training of the system and tuning of the parameters. The success depends heavily on those two. In case of multilayer perceptron network as used in [18], classification cost increases with the increase in number of layers and nodes present in the network. Thus, the proposed similarity matrix based boundary detection technique is relatively simple and of low cost compared to [18].

Table 7 Comparison of shot boundary detection performance in terms of F-measure(all figures in scale 0–1) on the dataset used in [1]

Sequence name	# of shot shot	System in [1]	System in [18]	System in [27]	System in [26]	Proposed system (confidence level 95%)
anni005	38	0.89	0.93	0.79	0.80	0.96 ±.008
anni006	41	0.85	0.94	0.76	0.79	0.96 ±.005
anni009	38	0.90	0.93	0.75	0.78	0.94 ±.005
BOR08	197	0.88	0.92	0.76	0.78	0.95 ±.007
NAD53	83	0.88	0.89	0.76	0.76	0.91 ±.014

4 Conclusion

In this work, we have presented a shot detection methodology developed following a simple shot transition model based on the similarity of the frames with respect to a reference frame. The model assumes that the frames in an individual shot are very similar and whenever a shot transition occurs a discontinuity in similarity values appears. Here, two algorithms are presented and they can work on large video sequence iteratively by taking a sub-sequence in the window. First algorithm directly applies the transition model and considers a threshold on the similarity values to detect the transitions. Its performance suffers if there exists multiple transitions in the window, and the selection of threshold on similarity value is non-trivial task. To overcome these limitations second and the final algorithm is presented which is the improvement over the first one and focuses on the slope of linear approximation of similarity values rather than the absolute similarity values. The said slope is estimated using least square regression. There is a possibility that the sequences which are not so well behaved may be oversplitted and to combat the issue a simple post-processing technique is presented. Experimental result and comparison with other systems indicate that the performance of our final algorithm is quite satisfactory and better. In the current scope of the work different threshold and other parameter values are selected experimentally. In future, effort may be put to formalize the same.

References

1. Adjeroh D, Lee MC, Banda N, Kandaswamy U (2009) Adaptive edge-oriented shot boundary detection. EURASIP J Image Video Process 2009:5:1–5:31
2. Amel AM, Abdessalem BA, Abdellatif M (2010) Video shot boundary detection using motion activity descriptor. J Telecommun 2(1):54–59
3. Amiri A, Fathy M (2009) Video shot boundary detection using generalized eigenvalue decomposition and gaussian transition detection. In: Proceedings of the international conference on computational science and its applications, pp 780–790
4. Bescos J, Cisneros G, Martinez JM, Menendez JM, Cabrera J (2005) A unified model for techniques on video-shot transition detection. IEEE Trans Multimed 7(2):293–307
5. Bhattacharyya A (1943) On a measure of divergence between two statistical populations defined by their probability distributions. Bull Calcutta Math Soc 35:99–109
6. Cernekova Z, Pitas I, Nikou C (2006) Information theory-based shot cut/fade detection and video summarization. IEEE Trans CSVT 16(1):82–91
7. Chen LH, Lai YC, Liao HYM (2008) Movie scene segmentation using background information. Pattern Recognit 41(3):1056–1068
8. Cooper M, Foote J (2005) Discriminative techniques for keyframe selection. In: Proceedings of the ICME, The Netherlands, pp 502–505
9. Grana C, Cucchiara R (2007) Linear transition detection as a unified shot detection approach. IEEE Trans CSVT 17(4):483–489
10. Hampapur A, Jain R, Weymouth T (1995) Production model based digital video segmentation. Multimed Tools Appl 1:1–38
11. Haoran Y, Rajan D, Chia LT (2006) A motion-based scene tree for browsing and retrieval of compressed video. Inf Syst 31(7):638–658
12. Huan Z, Xiuhuan L, Lilei Y (2008) Shot boundary detection based on mutual information and canny edge detector. In: Proceedings of the international conference on computer science and software engineering, pp 1124–1128
13. Huang CL, Liao BY (2001) A robust scene-change detection method for video segmentation. IEEE Trans CSVT 11(12):1281–1288
14. Le DD, Satoh S, Ngo TD, Duong DA (2008) A text segmentation based approach to video shot boundary detection. In: Proceedings of multimedia signal processing, pp 702–706

15. Ling X, Chao H, Huan L, Zhang X (2008) A general method for shot boundary detection. In: Proceedings of the international conference on multimedia and ubiquitous engineering, pp 394–397
16. Liu X, Chen T (2002) Shot boundary detection using temporal statistics modelling. In: Proceedings of the ICASSP, pp 3389–3392
17. Mas J, Fernandez G (2003) Video shot boundary detection based on color histogram. Notebook Papers TRECVID2003
18. Mohanta PP, Saha SK, Chanda B (2012) A model-based shot boundary detection technique using frame transition parameters. IEEE Trans Multimed 14(1):223–233
19. Murai Y, Fujiyoshi H (2008) Shot boundary detection using co-occurrence of global motion in video stream. In: Proceedings of the ICPR, pp 1–4
20. Patel NV, Sethi IK (1997) Video shot detection and characterization for video databases. Pattern Recognit 30(4):583–592
21. Porter S, Mirmehdi M, Thomas B (2001) Detection and classification of shot transitions. In: Proceedings of the 12th British machine vision conference. BMVA Press, pp 73–82
22. Rees DG (1987) Foundations of statistics. CRC Press
23. Smeaton AF, Over P, Doherty AR (2010) Video shot boundary detection: seven years of trecvid activity. Comput Vis Image Underst 114(4):411–418
24. Tsamoura E, Mezaris V, Kompatsiaris I (2008) Gradual transition detection using color coherence and other criteria in a video shot meta-segmentation framework. In: Proceedings of the ICIP, pp 45–48
25. Yoo HW, Ryoo HJ, Jang DS (2006) Gradual shot boundary detection using localized edge blocks. Multimed Tools Appl 28:283–300
26. Yuan J, Zheng W, Chen L, Ding D, Wang D, Tong Z, Wang H, Wu J, Li J, Lin F, Zhang B (2004) Tsinghua university at trecvid 2004: shot boundary detection and high-level feature extraction. In: Proceedings of the TREC Video Retrieval Evaluation (TRECVID), pp 84–196
27. Zhang C, Wang W (2012) A robust and efficient shot boundary detection approach based on fisher criterion. In: Proceedings of the ACM international conference on multimedia, pp 701–704
28. Zhang W, Lin J, Chen X, Huang Q, Liu Y (2006) Video shot detection using hidden markov models with complementary features. In: Proceedings of the international conference on innovative computing, information and control, pp 593–596

Debabrata Dutta obtained his B.Sc. and M.sc. degree in Computer Science from Calcutta University, West Bengal, India and Vidyasagar University, West Bengal, India in 2004 and 2006 respectively. Currently he is pursuing Ph.D. program in Computer Science and Engineering Department of Jadavpur University, West Bengal, India. His areas of interest are Image and Video Processing, Pattern Recognition.

Sanjoy Kumar Saha received his B.E. and M.E. Degree in Electronics and Tele-communication Engineering from Jadavpur University, West Bengal, India in 1990 and 1992 respectively and obtained his PhD from Bengal Engineering and Science University, West Bengal, India in 2006. Currently, he is working as a Reader in Computer Science and Engineering Department of Jadavpur University. His research interests are in the area of Image Processing, Video Processing, Multimedia Data Retrieval and Pattern Recognition.

Bhabatosh Chanda received B.E. in Electronics and Telecommunication Engineering and PhD in Electrical Engineering from University of Calcutta in 1979 and 1988 respectively. His research interest includes Image and video Processing, Pattern Recognition, Computer Vision and Mathematical Morphology. He has published more than 100 technical articles in refereed journals and conferences, authored one book and edited five books. He has received 'Young Scientist Medal' of Indian National Science Academy in 1989, 'Computer Engineering Division Medal' of the Institution of Engineers (India) in 1998, 'Vikram Sarabhai Research Award in 2002, and IETE-Ram Lal Wadhwa Gold medal in 2007. He is also recipient of UN fellowship, UNESCO-INRIA fellowship and Diamond Jubilee fellowship of National Academy of Science, India. He is fellow of Institute of Electronics and Telecommunication Engineers (FIETE), of National Academy of Science, India (FNASc.), of Indian National Academy of Engineering (FNAE) and of International Association of Pattern Recognition (FIAPR). He is a Professor in Indian Statistical Institute, Kolkata, India.