A Heuristic Algorithm for Video Scene Detection Using Shot Cluster Sequence Analysis

P. P. Mohanta*
ECS Unit, Indian Statistical Institute
Kolkata, India
ppmohanta@isical.ac.in

S. K. Saha
CSE Department, Jadavpur University
Kolkata, India
sks_ju@yahoo.co.in

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

ABSTRACT

In this paper, we present a novel scheme for segmenting video data into scenes. Based on visual similarity, the shots are first classified into clusters using modified k-means algorithm. Number of optimal clusters is decided using cluster validity analysis based on Davies-Bouldin index. Each shot is assigned a tag denoting the cluster it belongs to. Thus, the video data is represented by a sequence of cluster tags. The sequence is then analyzed by introducing the concept of stable and quasi-stable state. The elements of the sequence are merged into states and isolated elements are linked with the states to generate the scenes. The scheme is free from the dependency on critical parameters and capable of handling different types of scenes.

Keywords

video segmentation, scene detection, shot clustering

1. INTRODUCTION

A scene in a video data is the collection of semantically related consecutive shots sharing similar visual properties or representing the same event. A one hour video may contain few hundred shots. Thus, in the context of efficient organization and browsing of video data, shot based representation is quite inefficient. Hence, the focus has shifted to scene level representation and segmenting the video data into scenes becomes the fundamental step. Unlike Shots which are characterized by the physical boundary, the scenes are marked by the semantic boundary. Thus, the task of scene boundary detection is far more difficult.

Wide variety of approaches has been tried by the researches to extract scene boundary. Different graph based schemes have been proposed [2, 23, 6, 10, 20, 18, 19, 25]. Bouthemy et al. [2] considered an oriented graph where clusters of the shots are the nodes and edges denote the succession links between the clusters. Yeung et al. [23] have dealt with a shot transition graph where each shot corresponds to node and edge between them reflects the spatio-temporal relation. Based on hierarchical clustering graph is splitted into subgraphs denoting the scenes. Hanjalic et al. [6] considered a scene as a logical story unit consisting of the shots having dissimilarity among them within a threshold. A discrete graph based scheme is presented in [10]. Rui et al. [20] introduced a group threshold to group the shots based on spatio-temporal correlation and the groups were further clustered to form the scene subjected to another threshold value. Lin et al. [13] have proposed a pseudo-object-based shot correlation analysis to group the shots. A new grouping method called expanding window is proposed to cluster correlated consecutive shots into a scene. A sliding window based scheme is presented in [24] where shots are grouped considering visual similarity and temporal condition. The size of sliding window is a critical factor to decide. A two pass algorithm is presented in [18] where the shots are first clustered by computing a backward shot coherence and potential shot boundaries (PSB) are detected by measuring the shot colour similarity. In the second pass, scene dynamics based merging scheme is followed to remove the weak PSBs. But, the selection of window size in computing the backward shot coherence is an important issue. In another approach [19], the scene detection problem has been mapped onto graph partitioning problem and a scene detection process is carried out by forming a shot similarity graph. Zhao et al. [25] have proposed normalized cuts method to determine the optimal scene boundary.

In various schemes [1, 8, 12, 21], along with visual features, audio features are also considered for grouping the shots into scenes. Kang [9] has proposed a hierarchical approach for scene segmentation. In first phase, initial scene boundaries are detected by following a continuous coherence computing model. Subsequently, a refinement process based on k-means clustering and cluster validity analysis is carried out to obtain the result. Several researchers [22, 25] have tried to categorize the scenes into different types like, parallel scene, serial scene and tried to exploit the definitions. But, such categorizations are ill defined.

In [27], shots are first grouped by defining the correlation of a shot with its preceding and following ones. Finally, a group merging scheme is introduced to merge adjacent groups with higher correlation. Chasanis et al. [3] have clustered the shots into groups based on their visual similarity

*Corresponding author

and a label is assigned to each shot depending on the group that it belongs to. Then, a sequence alignment algorithm is applied to detect the change in shot label pattern and thereby segmentation is achieved. Choice of window size in sequence alignment process may bias the outcome. Zhu et al. [26] have considered coherence between the shots within a temporal constraint window to cluster the related shots into the same scene. But, the effectiveness of the scheme heavily depends on the size of the temporal constraint window and threshold values.

From the study of past work, it appears that the detection of scene, the semantic unit of a video, is a challenging task. The collection of consecutive shots depicting a theme may not be always visually similar. The span of the variation in the visual content of a scene depends on the dynamics of alternating sequences and editing process. In this context, it may be pointed out that wide variation may exist in the scene model. Thereby, a methodology to sustain considerable variety is still in demand. Most of the schemes depend heavily on sensitive parameters like, window size, threshold values. These observations have motivated us to develop a strategy with not so critical parameters. In this paper, we have presented a novel heuristic algorithm based on a modified version of k-means clustering and multistage cluster analysis. The paper is organized as follows. The introduction is followed by the description of proposed methodology in section 2. Experimental result is presented in section 3 and concluding remarks are placed in section 4.

2. PROPOSED METHODOLOGY

The characteristic of the scenes consisting of shots in restricted environment like indoor and that consisting of shots in open environment *i.e.*, outdoor are entirely different. In the first case, the contiguous shots forming the scene are taken with a fixed physical settings and they share a visually similar background. Very often such videos are captured by switching between several stationery cameras with different angles looking at the same background. Thus, there exists a repetitive pattern. Mostly, the news video, talk shows, non-action or dialog scenes shot in indoor falls into such category. On the other hand, for the second category, the background may not remain fixed during the total span of the scene. Movement of the camera may also lead to the change in the context continuously. Action scenes, videos captured in outdoor with camera mounted on a trolley belong to such category. Thus, it becomes a major challenge to handle the diverging features of different types of scenes. The performance of the graph based schemes suffers considerably in case of the scenes with varying content.

In our effort to deal with scenes of various types, we propose a methodology based on clustering and analysis of cluster sequence. Each shot is assigned a tag depending on the cluster that it belongs to. Thus, the sequence of shots is mapped onto a cluster sequence which is subsequently analyzed to glue the consecutive elements in the sequence to form the scenes. It is assumed that video data has already been segmented into shots and representative frames (keyframes) for the shots are also identified. The major steps namely, *generation of cluster sequence* and *analysis of the sequence* will be detailed in the subsections 2.1 and 2.2 respectively.

2.1 Generation of Cluster Sequence

Scene is a collection of semantically related contiguous shots. It is very difficult to represent the semantic content and we normally rely on the visual descriptors. Based on those descriptors shots are grouped into clusters. Thereby, a cluster represents the semantically cohesive shots. Depending on the cluster to which a shot belongs, a tag is assigned. Thus, the video data is represented by a sequence of cluster tags. The major issues involved at this stage are the representation of the shots and the clustering methodology.

2.1.1 Shot Representation

Shot representation is a crucial issue in the context of shot grouping. Various schemes have been considered to represent the shots. In [17], beginning and ending frames of a shot are taken as shot representative. Keyframes are also used in [6, 4] to represent the shots. Once the representatives are chosen, similarity between the representatives of the shots is taken as the shot similarity. Even, the similarity between two arbitrary frames of the shots has also been tried in [10]. Thus, comparing two shots heavily depends on their representation and it is quite difficult to represent the shots with a limited number of frames. On the other hand, comparing every pair of frames in two shots is prohibitively expensive.

For representing a shot, keyframes appear as the best choice as they denote most precise but general view of the overall shot content. But, in the context of shot similarity measurement, the keyframes alone may not reflect the complete flavor of a shot, particularly if the shot is eventful one. As a result, the shots, quite similar in their content, can have their keyframes with considerable spatial isolation in feature space and it may declare them to be dissimilar. Such possibility may be further enhanced by the bias/shortfall of the keyframe detection algorithm. Thus, to exploit the representation strength of the keyframes and also to reduce the biased characteristics, if any, of the keyframes we have considered number of sampled frames in the shot along with the keyframes as the representative of the shot.

As it is quite difficult to represent the semantic, we rely on the visual content described in terms of low level features. Each frame representing the shots are described by a feature vector. In this work, we consider features based on intensity histogram, wavelet decomposition, edge detection and colour correlogram. First, second and third order moments of each of the R, G and B histograms are considered as features. Thus 9 features are obtained. To compute the wavelet statistics also, we consider each of the R, G and B planes independently. For each plane, the image is decomposed up to three levels into one low pass and three high pass subbands [14]. Mean and variance of the high-pass subbands are taken as features. It may be noted that the information of the low-pass subband has already been reflected in the intensity histogram based features. Hence, to compute the wavelet based features we have concentrated only on the high-pass subbands. Thus, 18 wavelet based features are obtained. In order to compute edge based features the gradient image is first obtained. The pixels with gradient value more than the average are taken as the strong edge points. The image is divided into 16 grids and normalized count of strong edge points in the grids are taken as the features. To obtain the colour correlogram based features each of R, G and B scales are divided into 4 bins. Thus the colour values of the pixels are mapped onto 64 possible values. Then a 64×64 co-occurrence matrix is formed and based on it contrast, homogeneity, energy and entropy are computed as in [7]. Combining all the features a 47-dimensional feature vector is obtained and a shot is finally represented by a set of feature vectors corresponding to the keyframe(s) and sampled frames.

2.1.2 Clustering Methodology

Once the representative frames (*i.e.* keyframes and sampled frames) of all the shots in the video and the corresponding feature vectors are obtained, a standard clustering techniques , say, k-means algorithm is applied. In that case, role played by a sampled frame and that of a keyframe becomes equivalent. Thereby, the significance of the keyframe is ignored. Moreover, the representative frames of a shot may be distributed in multiple clusters in an arbitrary fashion and linking the shot/subshot to a cluster may become difficult. To address these issues, we adopt a modified version of k-means algorithm for shot clustering.

In our scheme, we incorporate the following modifications in the conventional k-means algorithm. A discrimination is made between the roles played by the keyframe elements and other sampled frame elements. Keyframe elements play the leading role and takes part in clustering process. Whereas, the sampled frame elements follow the keyframe element of the corresponding shot. in joining/leaving a cluster. Essentially, this is same as k-means clustering of keyframe elements only. But, instead of playing a completely passive role, the sampled frame elements takes part in computing the cluster centre and thereby the proposed clustering scheme becomes a little different from the conventional one. The concept is analogous to leader-followers movement. Leader decide which party/team to form or join or leave and the followers follow him and join or leave the same party or team accordingly. But once they join, the followers play equal role as the leader to change or shift the centre of gravity of that party or team.

Keyframes are more inclined toward the major activity in a shot. When a shot is semantically continued to another similar shot, a shift in activity may occur which may be reflected by the significant differences in the corresponding keyframes. Thus, clustering based on only keyframe elements may put them in different clusters. Here comes the importance of the role of sampled frame elements. As these elements take part in computation of cluster centre, a smoothing effect is imposed by shifting the cluster centre to minimize the strong bias of the keyframe elements. Thereby, it increases the probability of inclusion of similar shots in the same cluster in spite of their inherent spatial variation.

The major advantage of adopting clustering is that it has enabled us to judge the shot similarity without going for the critical task of threshold selection. On the other hand, the major problem of the scheme is to determine the optimal value of k, *i.e.*, the number of clusters. To address it, we rely on cluster validity analysis.

2.1.3 Determination of Optimal Number of Clusters

The optimal number of clusters for a given video data is determined by Davies-Bouldin index [5] based cluster validity analysis. The index is computed based on similarity measure between the clusters which are based on the dispersion measure of a cluster (s_i) and the cluster dissimilarity measure (d_{ij}). The similarity measure between i-th and j-th clusters (R_{ij}) is defined as

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

where,

$$d_{ij} = d(v_i, v_j)$$

and

$$s_i = \frac{1}{||c_i||} \sum_{x \in c_i} d(x, v_i)$$

where, c_i denotes the i-th cluster and $||c_i||$ is the number of elements in the cluster. v_i represents the centre of the i-th cluster. $d(v_i, v_j)$ stands for the distance between v_i and v_j. We have considered Euclidean distance. Once the clusters are formed, for validity analysis, we consider only the keyframes as the elements in the clusters and ignored the presence of sampled frames. Finally, Davies-Bouldin index is computed as

$$\frac{1}{n_c} \sum_{i}^{n_c} R_i$$

where, n_c is the number of clusters and

$$R_i = max\{R_{ij}\}, j = 1 \ldots n_c, i \neq j$$

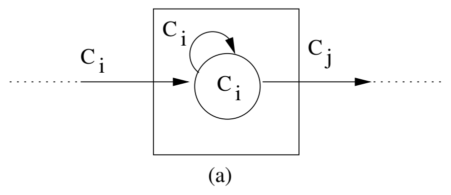
The index measures the average of similarity between each cluster and its most similar one. As it has been discussed in [11], lower the value of the index, better is the cluster configuration denoting compact and well separated clusters.

In order to determine the optimal number of clusters, we repeatedly execute the clustering process by varying k from a min to a max value. The case for which the index is minimum, is taken as the optimal number of clusters. The value of max is taken as $\frac{T}{t_{sc}}$, where, T is the duration of the video and t_{sc} is the minimum scene duration. In our experiment, min is taken as 2 and t_{sc} is taken as 45 seconds. It may be noted that the assumptions are non-critical. In case of a large video, in order to reduce the number of iterations, optimal number may be decided based on the local minima of the index.

2.2 Analysis of Cluster Sequence

The shots in a scene are semantically and temporally cohesive. Suppose clustering by some means takes care of the semantic aspect and ideally all the shots sharing a common semantic should belong to the same cluster. But they may not be part of same scene unless there is a temporal cohesion between them. The basic purpose of analysis phase is to glue the semantically cohesive shots by incorporating the temporal constraint. Moreover, measuring semantic cohesion in terms of visual descriptor and clustering has limitations. It may so happen that semantically similar shots are placed into different clusters. On the other hand, a scene consisting of alternating sequence or having considerable variations may have shots with different cluster tags. Thus, the analysis of cluster sequence becomes a very crucial step to cope up with the variety of situations.

The sequence under analysis is the collection of elements where the elements are the cluster tags corresponding to the shots in the video. The elements are arranged in accordance to the chronological order of appearance of the shots in the video. So the shots in a video is represented as a sequence of ordered pairs $((s_n, t_n)|n = 1, 2, \ldots)$ where s_n is the cluster



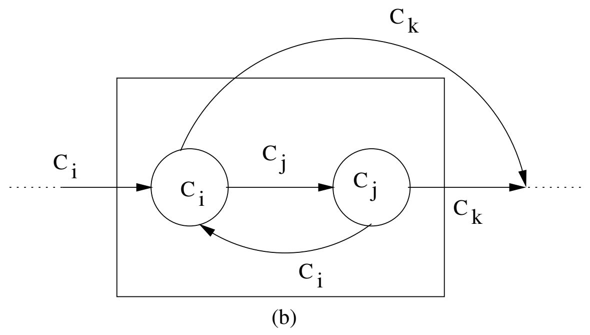


Figure 1: Formation of stable state – (a) by merging consecutive same cluster tags C_i, or (b) by merging same transitions between C_i and C_j, $i \neq j$

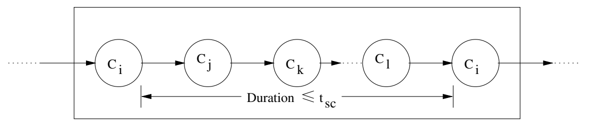


Figure 2: Formation of quasi-stable state

tag of the n-th shot, *i.e.*, $s_n \in \{C_1, C_2, \ldots, C_i, \ldots, C_j, \ldots\}$ and t_n is the timespan (number of frames) of the shot. We introduce the concept of *stable state* and *quasi-stable state*. A stable state (SS) is formed by merging either the subsequence of same elements as shown in Fig. 1(a) or the subsequence of same transition between a pair of elements as shown in Fig. 1(b). C_i, C_j etc. in Fig. 1 denote the cluster tags.

As shown in Fig. 2, a quasi-stable state (QS) is formed by merging the subsequence bounded by same cluster tag C_i with one or more intermediate elements other than C_i and total duration corresponding to the intermediate subsequence is not more than t_{sc}, the minimum scene duration. The major steps of the analysis phase are as follows.

- Identification of stable states.
- Merging non-adjacent but similar stable states.
- Identification of quasi-stable states.
- Merging quasi-stable states with stable states.
- Generation of candidate scene boundary.
- Generation of final scene boundary.

The details of the steps are described in the following subsections.

2.2.1 Identification of stable states

Identification of stable states consists of two steps: (a) *merging subsequence of shots with same cluster tags* and (b) *merging subsequence representing the frequently occurring cluster transition.* We consider the transitions between a pair of elements in the sequence and further assume that the transitions are symmetric *i.e.* transitions $C_i \rightarrow C_j$ and $C_j \rightarrow C_i$ are equivalent. The algorithm for steady state identification is as follows.

- Merge the subsequence of shots with same cluster homogeneous tags to form stable state. Timespan t is updated accordingly.
- Initialize all the elements of occurrence matrix, M by zero.
- *remaining_cluster* = C_i, the set of all cluster tags in the sequence.
- For each transition $C_i \rightarrow C_j$ or $C_j \rightarrow C_i$ in the sequence
  - increment $M[i][j]$ such that $i >= j$
- For each element $M[i][j]$ in M
  - if (($M[i][j] >$ *min_count*) and $(i, j) \in$ *remaining_set*) then
    - Form stable state by merging the subsequence of $C_i \rightarrow C_j$ transitions and its equivalent one.
    - Remove C_i and C_j from the *remaining_set*.
    - Assign a new tag to the merged subsequences and compute their timespan.

Thus, the sequence of cluster tags is converted into the sequence of elements where elements are either a cluster tag or a stable state. Scenes consisting of shots with similar visual content or with alternating sequence will have a tendency to converge in the stable states. It may be noted that once the transitions involving C_i and C_j are merged, merging any other transitions involving them is restricted. Thereby, the transitive effect is avoided at this stage. The potential transitions for merging are chosen by looking into the global pattern. Allowing transitivity may lead to uncontrolled localized growth of steady state at this early stage and may jeopardize the analysis at the very beginning. In our experiment *min_count* is taken as 2.

2.2.2 Merging non-adjacent but similar stable states

Due to the shortcomings of visual descriptor and/or clustering process, similar shots in a scene may bear different cluster tag. Even within an uniform scene, one or more shots can have variation putting them into different clusters. Moreover, the switching latency for an alternating sequence may also be quite high. In such cases, the homogeneity of cluster tag in the sequence breaks. Thus, a scene, instead of converging to a single stable state, it may get splitted into number of such states. To overcome the problem, we try to merge the similar states within a timespan of t_{sc} as shown in Fig. 3.

- SS_i and SS_{i+1} are two consecutive stable states with cluster tag C_i and C_{i+1} respectively.
- CS be the intermediate subsequence between SS_i and SS_{i+1}.
- t be the total timespan between SS_i and SS_{i+1}.

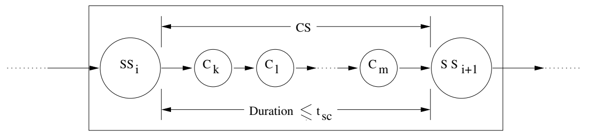


Figure 3: Merging of stable state

- if ($t < t_{sc}$ *and* $C_i == C_{i+1}$) then Merge SS_i, CS and SS_{i+1}.
- Repeat the above steps for all consecutive stable state pairs.

2.2.3 Identification of quasi-stable states

In the previous two steps similar cluster tags and transitions formed by alternate cluster tags are merged into stable states. So in the first case the periodicity of the tags is one and in the second case it is two. According to the definition of quasi-stable state, this periodicity of a certain tag in a subsequence may be more than two and also uneven. So a quasi-stable state may be identified as a subsequence where a cluster tag occur within a distance not more than t_{sc}. More formally, suppose cluster tag for n-th shot, s_n is C_i and that for s_{n+r} is also C_i, we mark s_n to s_{n+r} as a quasi-stable state if the time interval between them is not more than t_{sc}. An example of quasi-stable state is shown in Fig. 2. Remaining shots or cluster tags in the sequence are treated as isolated tags or shots.

2.2.4 Merging quasi-stable states with stable states

At this stage, the sequence may be thought of as the collection of stable states, quasi-stable states and isolated clusters. As in many cases, the quasi-stable states may arise out of the variation within a scene, we try to link such quasi-stable states with neighboring stable state(s). The steps are as follows.

```
• For each quasi-stable state $QS_i$
    – Within time interval $t_{sc}$, look for $PS$ and $FS$, the preceding and following stable states.
    – $CS_p = null$, $CS_f = null$, $CQ$ = set of cluster tags in $QS_i$
    – if ($PS$ exists) then
        ∗ $CS_p$ = set of cluster tags in $PS$
        ∗ $IS_p$ = intermediate sequence between $PS$ and $QS_i$
    – if ($FS$ exists) then
        ∗ $CS_f$ = set of cluster tags in $FS$
        ∗ $IS_f$ = intermediate sequence between $QS_i$ and $FS$
    – $N_p = |CQ \bigcap CS_p|$, $N_f = |CQ \bigcap CS_f|$
    – if ($N_p == N_f$ and $N_p > 0$) then
        ∗ merge $PS$, $IS_p$, $QS_i$, $IS_f$ and $FS$.
    – else
        ∗ if ($N_p > N_f$) then merge $PS$, $IS_p$ and $QS_i$.
        ∗ else if ($N_F > 0$) then merge $QS_i$, $IS_f$ and $FS$.
```

It may be noted that a quasi-stable state may be linked either with the preceding or following stable state or both or none. At the time of linking, intermediate subsequence is also merged along with.

2.2.5 Generation of candidate scene boundary

The sequence now consists of the elements mostly denoting the stable states and the quasi-stable states which could not be merged with neighboring stable states. There may also exist few isolated tags which did not qualify earlier to become the part of stable/quasi-stable states. As the states have grown further through merging, there is a possibility for accommodating such isolated clusters into the states. Moreover, such elements, of their own also do not qualify to define a scene. Hence, we opt for merging them with the neighboring states. Steps for assigning the isolated cluster tags to adjacent stable states are as follows.

```
• For each isolated element $C_i$
    – Let $S_p$ ($S_f$) be the preceding (following) state within time interval $t_{sc}$.
    – Let $C_p$ ($C_f$) be the set of cluster tags present in $S_p$ ($S_f$).
    – if ($C_i \notin C_p$ and $C_i \notin C_f$) then
        ∗ if ($|C_p| > |C_f|$) then merge $C_i$ and intermediate subsequence with $S_p$ else merge $C_i$ and intermediate subsequence with $S_f$.
    – if ($C_i \in C_p$ and $C_i \notin C_f$) then
        ∗ merge $C_i$ and intermediate subsequence with $S_p$.
    – if ($C_i \notin C_p$ and $C_i \in C_f$) then
        ∗ merge $C_i$ and intermediate subsequence with $S_f$.
    – if ($C_i \in C_p$ and $C_i \in C_f$) then
        ∗ if ($S_p$ and $S_F$, both are quasi-stable states) then merge $S_P$, $C_i$ and $S_f$ along with intermediate subsequences.
        ∗ if (either $S_p$ or $S_F$ (not both) is quasi-stable state) then merge $C_i$ with the quasi-stable state along with intermediate subsequence.
        ∗ if ($S_p$ and $S_F$, both are stable states) then
            · if ($|C_p| > |C_f|$) then merge $C_i$ and intermediate subsequence with $S_p$ else merge $C_i$ and intermediate subsequence with $S_f$.
```

Once the isolated tags are linked with the states, the sequence consists of only stable and quasi-stable states. The stable states and the quasi-stable states are primarily taken as the candidate scenes.

2.2.6 Generation of final scene boundary

In the final stage of analysis, it is checked whether the candidate scenes are of considerable duration or not. The duration of each candidate scene is verified against t_{sc} and a candidate scene with smaller duration is merged with preceding/following candidate scene or both. Let S_c be the candidate scene, S_p and S_f are the preceding and following the candidate scene respectively. C_c, C_p and C_f are the set of cluster tags in S_c, S_p and S_f respectively. Based on these notations the merging algorithm is stated as follows.

Table 1: **Performance of the proposed scheme**

Data	Time (min.)	Scene Detection			% Precision	% Recall
		correct	miss	false		
movie1	25	19	2	3	86.36	90.29
movie2	37	11	1	3	78.57	91.66
movie3	15	09	0	1	90.00	100.00
overall		39	3	7	84.78	92.86

```
• For each candidate scene $S_c$
    – $t$ = time duration of $S_c$
    – if ($t < t_{sc}$) then
        ∗ if ( $|C_c \bigcap C_p| > |C_c \bigcap C_f|$) then merge $S_c$ with $S_p$.
        ∗ if ( $|C_c \bigcap C_p| < |C_c \bigcap C_f|$) then merge $S_c$ with $S_f$.
        ∗ if ( $|C_c \bigcap C_p| == |C_c \bigcap C_f|$) then merge $S_p$ , $S_c$ and $S_f$.
```

Thus, the sequence becomes a collection of states with considerable duration. Each state (stable or quasi-stable) represents a scene.

It may be noted that the proposed methodology relies on the parameter t_{sc} denoting minimum scene duration. But, it does not demand a precise value of it. Too high value for t_{sc} is prohibitive as it may restrict optimal number of clusters to a low value. Moreover, the linking of states and other elements may span over a long duration leading to merging of scene. On the other hand, a low value is less detrimental. First of all, in determining the optimal number of clusters, it will have no impact other than increasing the number of iteration. During analysis, low value of t_{sc} restricts the span of linking activity within a shorter period. Thus, in the worst case, it may lead to initial over segmentation which may be taken care of in the later stages. So the effect is less pronounced as the growth of states reduces the interval between them. Thus, a moderate approximation for t_{sc} which is neither too high and nor too low is good enough for the scheme to work. Moreover, such approximation is not too critical.

3. EXPERIMENTAL RESULTS AND DISCUSSION

In order to carry out the experiment, we have worked with part of three movie videos namely *Matrix Revolution*, *Mission Impossible* and *Catch Me If You Can*. These are being referred to as movie1, movie2 and movie3 respectively in Table 1. The video data is first segmented into shots following the scheme presented in [15] and for each shot keyframe(s) are detected based on the methodology in [16]. Movie1 consists of 43,400 frames and is segmented into 425 shots. For movie2 number of frames and shots are 57,285 and 539 respectively. Movie3 has 28,671 frames with 151 shots. To verify the performance of the proposed methodology, video data is manually groundtruthed for scenes and scene boundaries.

Here the performance of scene detection algorithm is measured in terms of precision and recall which are computed as

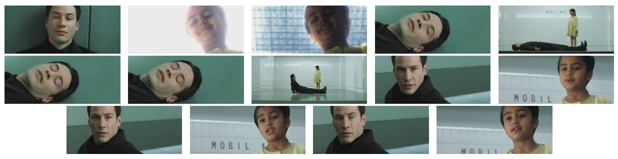
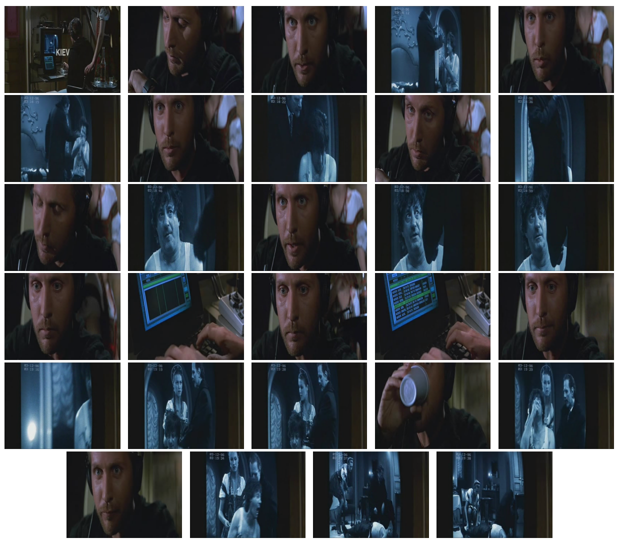


Figure 4: Keyframes of a detected scene in movie1


(a)

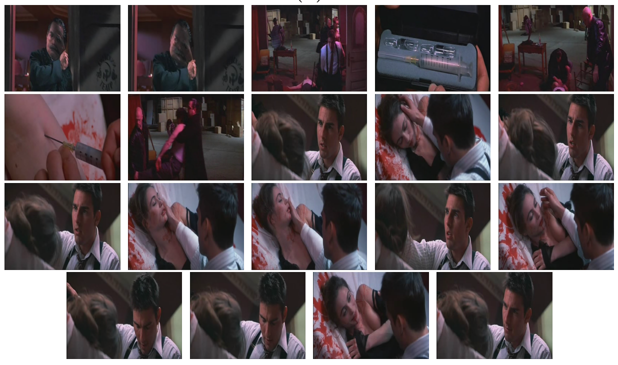
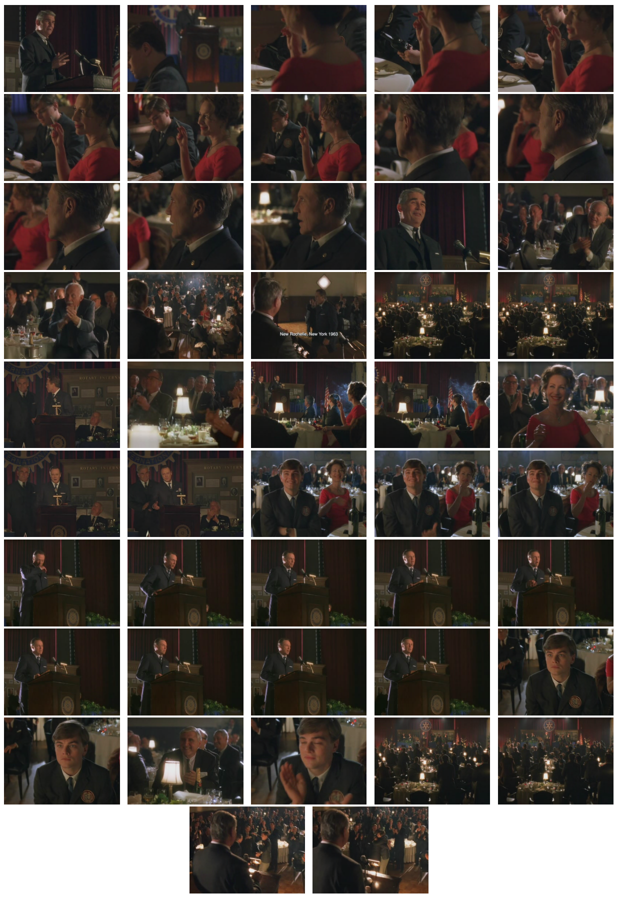

(b)

Figure 5: (a) and (b) are the keyframes of the shots in two consecutive scenes in movie2


(a)

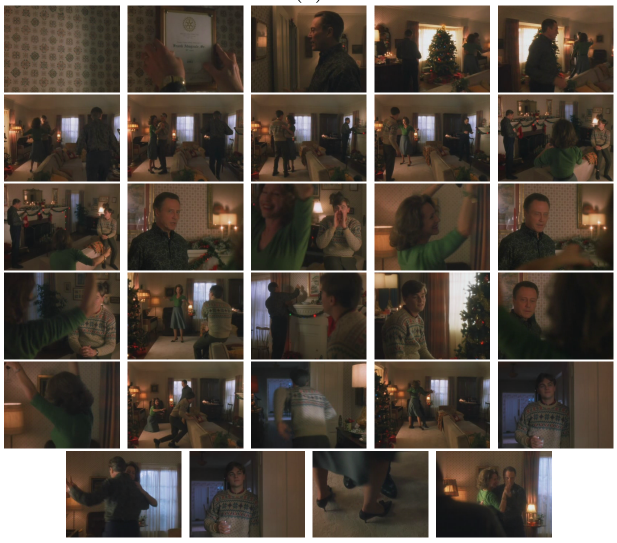

(b)

Figure 6: (a) and (b) are the keyframes of the shots in two consecutive scenes in movie3

$\frac{c}{c+f}$ and $\frac{c}{c+m}$ respectively, where c, f and m denote number of correctly detected scenes, number of falsely detected scenes and number of scenes that could not be detected (miss in detection) respectively. Table 1 presents the result of the experiment described above and shows that the performance of the proposed scheme is good enough in detecting the scenes. High recall indicates that the chance of missing a scene is quite low. But, it appears that the scheme is prone to over segmentation leading to relatively low precision. In order to avoid it, the merging policy described in section 2.2.6 has to be improved in future. It has been observed that for a fast video (as in case of movie1), the shots and the scenes are of small duration and it is reverse for a slow movie/video. The long scenes with significant variations in the content (as in case of movie2) in some cases may get splitted. Each such segment may qualify to stand as a scene on their own and thereby could not be merged. Instead of duration based merging, a localized scene dynamics has to be incorporated to surmount this problem. For a few detected scenes, the keyframes of the shots have been shown in Fig. 4, 5 and 6. The figures show that proposed methodology is capable of detecting scenes with alternating shots and the scenes with variation in the contents.

4. CONCLUSION

We have presented a methodology for segmenting video data into scenes. It is capable of handling scenes of various types. The shots are classified into clusters, where the optimal number of clusters is determined using cluster validity analysis. The video data is then mapped onto a sequence of cluster tags. The concept of stable and quasi-stable and a subsequent linking procedure is introduced to generate the scene boundaries. The multistage analysis procedure is found to withstand the limitations imposed by the visual descriptors in representing the semantic content and the limitation of the clustering process including the number of clusters. Thus, the scheme can successfully handle wide variety of scenes. Though the methodology relies on a parameter denoting the minimum scene duration, the procedure does not demand precise value of it and the analysis procedure has its inherent strength to cope up with the impreciseness of the parameter. Thus, it is free from the dependency on any critical parameter.

5. REFERENCES

[1] Y. Ariki, M. Kumano, and K. Tsukada. Highlight scene extraction in real time from baseball live video. In *Proc. ACM Intl. Workshop on Multimedia Retrieval*, pages 209 – 214, 2003.

[2] P. Bouthemy, C. Garcia, R. Ronfard, G. Tziritas, E. Veneau, and D. Zugaj. Scene segmentation and image feature extraction for video indexing and retrieval. In *Proc. Visual*, pages 245 – 252, 1999.

[3] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *IEEE Trans. on Multimedia*, 11(1):89 – 100, 2009.

[4] J. M. Corridoni and A. D. Bimbo. Structured representation and automatic indexing of movie information content. *Pattern Recognition*, 31(12):2027 – 2045, 1998.

[5] D. L. Davies and D. W. Bouldin. Cluster separation measure. *IEEE Trans. on PAMI*, 1(2):224 – 227, 1979.

[6] A. Hanjalic, R. L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval. *IEEE Trans. on CSVT*, 9(4):580 – 588, 1999.

[7] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for images classification. *IEEE Trans. on Syst., Man and Cybern.*, SMC-3:610 – 621, 1973.

[8] J. Huang, Z. Liu, and Y. Wang. Integration of audio and visual information for content-based video segmentation. In *Proc. ICIP*, pages 526 – 530, 1998.

[9] H. B. Kang. A hierarchical approach to scene segmentation. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 65 – 71, 2001.

[10] J. R. Kender and B. L. Yeo. Video scene detection via continuous video coherence. In *Proc. IEEE Intl. Conf. on CVPR*, pages 367 – 373, 1998.

[11] F. Kovacs, C. Legany, and A. Babos. Cluster validity measurement techniques. In *Proc. Intl. Symposium of Hungarian Researchers on Computational Intelligence*, pages 65 – 71, 2001.

[12] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Scene determination based on video and audio features. In *Proc. Intl. conf. on Multimedia Computing and Systems*, pages 685 – 690, 1999.

[13] T. Lin, H. J. Zhang, and Q. Y. Shi. Video content representation for shot retrieval and scene extraction. *Intl. J. of Image Graph*, 1(3):507 – 526, 2001.

[14] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on PAMI*, 18:837 – 842, 1996.

[15] P. P. Mohanta, S. K. Saha, and B. Chanda. Shot boundary detection using frame transition parameters and edge strength scatter. In *Proc. PReMI*, pages 641 – 648, Kolkata, India, 2007.

[16] P. P. Mohanta, S. K. Saha, and B. Chanda. A novel key-frame detection technique using statistical run test and majority voting. In *Proc. ICVGIP*, Bhubaneswar, India, 2008.

[17] W. Qi, H. Jiang, X.-R. Chen, and H.-J. Zhang. Integrating visual, audio and text analysis for news video. In *Proc. ICIP*, 2000.

[18] Z. Rasheed and M. Shah. Scene detection in hollywood movies and tv shows. In *Proc. CVPR*, pages 343 – 348, 2003.

[19] Z. Rasheed and M. Shah. Detection and representation of scenes in video. *IEEE Trans. on Multimedia*, 7(6):1097 – 1105, 2005.

[20] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-content for video. *ACM Multimedia Systems*, 7(5):359 – 368, 1999.

[21] H. Sundaram and S. F. Chang. Video scene segmentation using video and audio features. In *Proc. ICME*, pages 1145 – 1148, 2000.

[22] W. Tavanapong and J. Zhou. Shot clustering techniques for story browsing. *IEEE Trans. on Multimedia*, 6(4):517 – 526, 2004.

[23] M. Yeung and B. L. Yeo. Segmentation of video by clustering and graph analysis. *Intl. J. of Computer Vision and Image Understanding*, 71(1):94 – 109, 1998.

[24] L. Zhao, , S.-Q. Yang, and B. Feng. Video scene detection using slide wimdows methobd based on temporal constraint shot similarity. In *Proc. ICME*, pages 649 – 652, 2001.

[25] Y. J. Zhao and T. Wang. Scene segmentation and categorization using ncuts. In *Proc. CVPR*, pages 343 – 348, 2007.

[26] S. Zhu and Y. Liu. Video scene segmentation and semantic representation using a novel scheme. *Multimedia Tools and Application*, 42:183 – 205, 2009.

[27] X. Zhu, A. K. Elmagarmid, X. Xue, L. Wu, and A. C. Catlin. Insight video: Toward hierarchical video content organization for efficient browsing, summarization and retrieval. *IEEE Trans. on Multimedia*, 7(4), 2005.