Nyström approximated temporally constrained multi-similarity spectral clustering approach for movie scene detection

Rameswar Panda, Student Member, IEEE, Sanjay K. Kuanar, and Ananda S. Chowdhury*, Senior Member, IEEE

***Abstract*—Movie scene detection has emerged as an important problem in present day multimedia applications. Since a movie typically consists of huge amount of video data with widespread content variations, detecting a movie scene has become extremely challenging. In this paper, we propose a fast yet accurate solution for movie scene detection using Nyström approximated multi-similarity spectral clustering with a temporal integrity constraint. We use multiple similarity matrices to model the wide content variations typically present in any movie dataset. Nyström approximation is employed to reduce the high computational cost of constructing multiple similarity measures. The temporal integrity constraint captures the inherent temporal cohesion of the movie shots. Experiments on five movie datasets from different genres clearly demonstrate the superiority of the proposed solution over the state-of-the-art methods.**

***Index Terms*—Movie scene detection, Spectral clustering, Similarity matrices, Nyström approximation.**

I. Introduction

With the recent development of inexpensive digital multimedia technologies along with lower cost of publishing and wide potential reach, there has been a tremendous increase in the number of videos over the Internet [1], [2]. For example, one of the most prevalent social media services and web sites like YouTube reported that over 1 billion unique users visit YouTube each month and 300 hours of video, including movies, are uploaded every minute, amounting to nearly 1.5 billion hours of video every year. The task of managing this large amount of video information is an enormously challenging task. Computationally efficient methods are necessary to process, organize, summarize and index this information in a semantically meaningful manner [3].

Movie scene detection is an important problem in the area of multimedia content management. Scene(s), composed of groups of shots, usually emphasize specific concepts (e.g. a fixed setting or the same action) and are hence found to be semantically meaningful. Shot level video segmentation and video summarization, the two existing techniques for handling large amount of video data prevalent in the internet are found to be inadequate to handle the current problem. This is largely because movies are of long durations and have widely varying contents. Shot level segmentation methods are inefficient to organize the chapters of a movie which correspond to various themes. On the other hand, browsing a movie can be often more convenient and meaningful using less number of scenes (say, 100) compared to a large number of key frames (say, 10,000) which can be obtained from typical video summarization methods [4]-[10]. The scene detection problem can be modeled as a shot clustering problem where each cluster should be semantically distinct [11]-[13]. In this paper, we propose a novel fast yet accurate solution for detecting movie scenes using Nyström approximated multi-similarity spectral clustering with a temporal integrity constraint.

• Rameswar Panda is currently with the Department of Electrical and Computer Engineering at University of California, Riverside, USA. This work was completed when Rameswar Panda was a student at Jadavpur University, Kolkata, India. (email: rpand002@ucr.edu)

• Sanjay K Kuanar is with the Department of Computer Science and Engineering, Padmanava College of Engineering, Rourkela, India. (email: sanjay.kuanar@gmail.com)

• Ananda S. Chowdhury is with the Department of Electronics and Telecommunications Engineering, Jadavpur University, Kolkata, India. (e-mails: aschowdhury@etce.jdvu.ac.in)

* Corresponding Author.

A. Related Work

Our work spans following areas of interest - video summarization, video scene detection and multi-view learning. We will review some representative works from these areas.

Video Summarization: There exists several methods for video summarization. Avila et al. used color feature and an improved k-means algorithm to choose the key frames [4]. Kuanar et al. [5], applied both color and texture features with a dynamic Delaunay clustering for the same purpose. Almeida et al. [6], utilized the notion of color similarity between successive frames to extract the key frames in the MPEG compressed domain. Han et al. [7] examined color in combination with human in the loop guidance for personalized video summarization. Recently, a work on scalable video summarization using skeleton graph and random walk is reported [8]. For more comprehensive review of video summarization approaches, please see [9]-[10].

Video Scene Detection: Since, we propose a graph-theoretic solution, we first discuss some prominent graph-based approaches for video scene detection. Yeung et al. [14] represented video as a scene transition graph (STG), where shots are clustered and then each cluster is represented by a node in the graph. The complete link method is used to split the graph into several sub-graphs (i.e. scenes). In [15], a weighted shot similarity graph (SSG) is constructed, where each node represents a shot and the edges between shots are weighted by color and motion similarity information. Then Normalized cut is used for recursive bi-partitioning of SSG, to maximize intra-subgraph similarities while minimizing inter-subgraph similarities. These partitions depict individual scenes present in the video. A similar approach is presented in [16], where shot clustering was achieved using Ncuts algorithm in the first step and the resulting clusters are represented by a temporal graph. In another graph partition based method [17], a one-

dimensional signal is constructed for each feature. Chasanis et al. [13] proposed a spectral graph based approach using visual similarity of individual shots. A label is assigned to each shot depending on the cluster it belongs to. Then, a global sequence alignment algorithm is applied to detect the change in shot label pattern. Odobez et al. [18] proposed a spectral method with automatic model selection for video scene detection. However, the approach is only restricted to scene detection in home videos. Another spectral clustering method for scene segmentation is presented in [19] where authors use the concept of JSEG measure to capture the local information embedded in video shots. Although the method presented in [19] is independent of video genres but the incorporation of temporal information in form of a sliding window while computing the shot similarity greatly influences the detection results. Sakarya et al. [20] introduced a movie scene detection method based on finding dominant sets in shot similarity graph. In this work, two graph partitioning approaches, i.e., tree-based approach and order-based approach, using dominant set clustering are applied for movie scene detection. However, inaccurate estimation of control variables such as temporal distance decay factor and outlier detection factor may adversely affect the scene detection result.

Other than graph-theoretic methods, statistical approaches are also applied for the scene detection problem. A Markov Chain Monte Carlo (MCMC) method is presented in [11]. The authors use three types of updates, i.e., diffusion, merge and split to determine the scene boundaries. However, this method is highly sensitive to model prior and the number of shots. Tan et al. [21] proposed a Gaussian mixture model for scene segmentation where each scene is modeled as a Gaussian density assuming similar visual features for the shots constituting a scene. This method is able to discover scene-level semantics for sports videos. However, for more general video genres, such as movies, only using the features of individual shot is not sufficient. The impact from the neighboring shots (i.e. temporal integrity) should also be considered. Sundaram et al. [22] proposed a computational scene model to achieve video scene segmentation. In this work video and audio scenes were detected separately and these two were used with some cinematic rules in order to construct scenes.

Apart from the above unsupervised approaches, there has been a growing interest in developing supervised or semi-supervised algorithms for detecting video scene boundaries [40], [41], [43]. A generic framework based on semi-supervised learning for video annotation can be seen in [42]. Authors in [42] use the combination of multi-modal information by developing a graph-based semi-supervised multiple instance learning framework for generic video annotation. It jointly explores small-scale expert labeled videos which are obtained from analysis and alignment of well-structured video related text (e.g., movie script and captions) and large-scale unlabeled videos which are obtained by querying related events from the video search engine (e.g., YouTube, Google) to train a discriminative model for video annotation. A novel metric for evaluating scene segmentation methods is presented in [45]. Recently, deep learing based approaches have been developed for scene detection in broadcast videos [45], [46]. A graph based metric learning approach for dyanamic scene segmentation can be seen in [46]. In contrast, we propose an unsupervised framework for movie scene detection that does not require any label information.

Multi-view Learning: In recent years, many methods of clustering from multi-view data by considering different views have been proposed. These views may be obtained from multiple sources or different feature subsets [48]-[54]. Our work is closely related to different multi-view learning methods since we use different sets of features of a movie for efficient detection of scene boundaries. In contrast to the prior works, our proposed approach is different in two significant ways. First, we consider Nyström approximation in constructing the feature similarity matrices which are then used in the spectral clustering. The use of Nyström approximation reduces the computation burden to a large extent with only a marginal compromise in the detection performance. Second, we explicitly use a temporal constraint in our formulation to enforce the temporal cohesion between video shots which is crucial in movie scene detection.

By reviewing the related works on scene detection, we found that the following three key problems in movie scene detection still remain unaddressed to a considerable extent:

i) Proper Selection of Features: It has been observed that different features and homogeneity criteria generally lead to different segmentations of the same video. This problem is even more prominent in case of movie scene detection as different types of content variations (due to variations in shooting and editing effects at various stages of the video life cycle) exist across the shots. So, selection of multiple features and an optimal weight assignment policy for their combination is highly necessary, a task missed by most of the prior works.

ii) Use of Computationally Efficient Technique for Shot Similarity Calculation: Several recently proposed scene detection techniques compute pair-wise similarities for the clustering purpose [12], [13]. Such computation has to be carried out for all possible shot pairs in a video [24]. A case in the point is the movie Gone in 60 seconds with nearly 2900 shots. Processing this movie for five similarity measures can easily involve 42 million pair-wise comparisons! Hence an efficient approximation technique is required for clustering of large data like movie to substantially reduce the computational costs.

iii) Representation of Temporal Cohesion of Video Shots: It can be easily noticed that a video scene has temporal integrity. So, temporal cohesion of movie shots is required to achieve accurate scene detection [12]. Most of the previous shot clustering approaches use temporal distance as a solution for this problem [12], [20]. Due to the absence of prior knowledge about the video content and the duration of scenes, it is difficult to determine an appropriate weight parameter that will account for the contribution of the temporal distance in the computation of the overall similarity between shots.

In this work, we address the above three key problems in our solution pipeline: Firstly, a combination of multiple shot similarity matrices involving color, texture, motion and semantics is proposed to capture the diverse characteristics of different types of movie scenes. For example, color feature can effectively model a stationary scene whereas motion features are essential for action scenes. Secondly, Nyström approximation is employed to reduce the high computational

cost of constructing multiple similarity measures. As a third contribution, we have directly incorporated temporal integrity constraints in the multi-similarity spectral clustering thereby obviating incorporation of temporal distance in form of a similarity matrix. Note that in the later case, prior knowledge is essential which is always difficult to obtain for movie scenes [13]. In addition, from pure theoretical perspective, the proposed Nyström approximated temporally constrained multi-similarity spectral clustering approach has not been applied in the field of pattern clustering to the best of our knowledge. Note that, the convergence is guaranteed for the Nyström approximated eigenvectors but not for the generalized eigenvectors [24].

B. Theoretical Foundations

Our movie scene detection approach is primarily based on spectral clustering and application of Nyström extension for similarity matrix completion. Some useful theories pertaining to our approach are briefly reviewed in this section.

i) Nyström Extension: The Nyström method is a technique for finding numerical approximations to eigenfunction of integral equations of the form [24]:

$$\int W(x,y)\,\phi(y)\,p(y)\,dy = \lambda\phi(x), \tag{1}$$

where $p(y)$ represents the underlying probability density function, $\phi(y)$ indicates the eigenfunction and $W(x,y)$ denotes the similarity between x and y. For finding numerical approximation to (1), one need to choose n_s number of landmark points $Z = \{Z_1, Z_2 \ldots Z_{n_s}\}$ from the given dataset $X = \{X_1, X_2 \ldots X_n\}$ with $n_s << n$. For any given point x in X, using Nyström approximation we can write

$$\frac{1}{n_s}\sum_{i=1}^{n_s} W(x, Z_i)\widehat{\phi}(Z_i) = \lambda\widehat{\phi}(x), \tag{2}$$

where $\widehat{\phi}(x)$ is an approximation to the true $\phi(x)$. Eq. (2) cannot be solved directly as λ and $\widehat{\phi}(x)$ are both unknown. In order to solve Eq. (2), one needs to substitute x with Z_i, and write it in matrix form $A\widehat{\Phi} = P\widehat{\Phi}\wedge$, where A denotes the similarity matrix between landmark points and $\widehat{\Phi}$ represent the eigenvectors of A. $\wedge = diag\left\{\widehat{\lambda_1}, \widehat{\lambda_2} \ldots, \widehat{\lambda_{n_s}}\right\}$ is a diagonal matrix. For an unsampled point, the j^{th} eigenfunction at x can be approximated as

$$\Phi_j(x) = \frac{1}{n_s\widehat{\lambda_j}}\sum_{i=1}^{n_s} W(x, Z_i)\widehat{\phi}_j(Z_i). \tag{3}$$

With the above equation, the eigenvector for any arbitrary point x can be approximated by the eigenvectors of the landmark similarity matrix.

ii) Nyström Extension to Spectral Clustering: Let $A = U_A \wedge_A U_A^T$ be the eigen-decomposition of A. Further, let B denotes the similarity matrix between sample points and the remaining points, with $B \in R^{m\times(n-n_s)}$. From (3), the matrix form of the Nyström extension is then $B^T U_A \wedge_A^{-1}$, where B^T corresponds to $W(Z_i, \cdot)$, the columns of U_A correspond to the $\widehat{\phi}_j(Z_i)$s, and $\wedge_A^{-1}$ corresponds to the $1/\widehat{\lambda}_j$s. Let $W \in R^{n\times n}$ be the similarity matrix between all data points. For simplicity in notation, let us rearrange the points such that n_s randomly sampled points come first and remaining samples come next. Now partition the similarity matrix W as:

$$W = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \tag{4}$$

where $C \in R^{(n-n_s)\times(n-n_s)}$ is the similarity matrix between unsampled points. Using the approximated eigenvectors $\widehat{U} = \left[U_A; B^T U_A \wedge_A^{-1}\right]$, W can be estimated as:

$$\widehat{W} = \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix} = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} \begin{bmatrix} A & B \end{bmatrix} \tag{5}$$

For spectral clustering, the similarity matrix is required to be normalized, i.e., one has to calculate row sums of W to acquire D. Depending on definiteness of A, D can be estimated through the row sums of $\widehat{W}$ in two different ways.

Case 1. A is positive definite: When matrix A is positive definite, all the eigenvalues of matrix A are positive and $A^{-1/2}$is defined. The orthogonalized approximate eigenvectors are obtained by

$$\widehat{V} = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2} U_S \wedge_S^{-1/2} \tag{6}$$

where $S = A + A^{-1/2} B B^T A^{-1/2}$ with eigen decomposition $U_S \wedge_S U_S^T$.

Case 2. A is indefinite: When A is indefinite, then two steps are required to get the normalized solution. Let $\overline{U}_S^T = \left[U_S^T \wedge_S^{-1} U_S^T B\right]$ and $Z = \widehat{U}\wedge^{1/2}$ so that $\widehat{W} = ZZ^T$. $F\sum F^T$ denote the diagonalization of $Z^T Z$. Then matrix $V = ZF\sum^{-1/2}$ contains the leading eigenvectors of $\widehat{W}$. More details about the Nyström approximation and its extension to spectral clustering can be seen at [24].

II. Proposed Framework

Our proposed method consists of four major steps, namely, i) shot detection and representation, ii) shot similarities computation, iii) spectral grouping of shots, and iv) cluster sequence analysis. A block diagram of the proposed method is shown in Fig. 1. Now, we describe these four steps in details under four subsections.

A. Shot detection and representation

We first divide the movie into shots using information theory-based shot detection method by Cernekova *et al*. [27]. This method is shown to yield high detection accuracy on the TRECVID 2003 video test set. Various schemes exist to represent the video shots using a single key frame or a set of key frames [10], [14], [17]. Fig. 2 shows a comparative analysis of shot representation methods. In general, on analyzing Fig. 2, we can conclude that the middle frame is a good choice for representing the movie shots as it can capture the general view of the overall content. Hence, we have taken the middle frame of a shot as its representative thereby avoiding considerable computations in selecting the key frames of a shot [12].

B. Shot Similarities Computation

To properly capture diverse characteristics of different types of movie scenes, we apply multiple feature similarity matrices. Color, texture and motion similarity functions between two shots (i.e., representative key frames) are calculated. Color

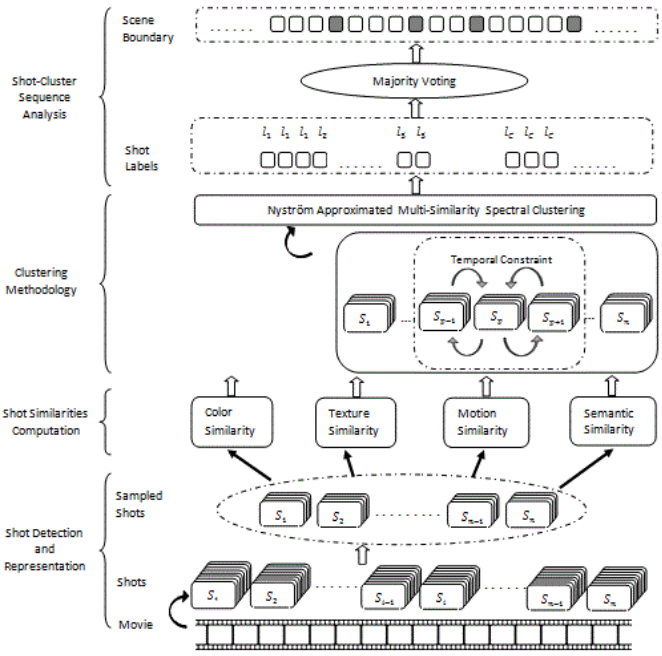

Fig. 1. Flowchart of the Proposed Method. We first divide the movie into shots using an information theory-based shot detection method and represent each shot by its middle frame. We employ Nyström sampling to select a group of shots and compute different shot similarities (color, texture, motion and semantic). Nyström approximation is employed to reduce the high computational cost of constructing multiple similarity measures. We then apply multi-similarity spectral clustering with a temporal integrity constraint to cluster the shots. Finally, shot-cluster sequence analysis is used to detect the precise scene boundaries.

Fig. 2. Shot representative frames using different methods from the movie "Kingdom of Heaven". As can be seen from the fig, there is a little difference among the different methods in representing a shot. Thus, the middle frame is a good choice for representing the movie shots as it can capture the general view of the overall content by avoiding considerable computational overhead in selecting the key frames.

histogram is obtained using the HSV color space, as it is found to be more resilient to noise [28]. We use 16 ranges of H, 4 ranges of S, and 4 ranges of V to form a 256 dimensional color feature vector and an edge histogram descriptor [29] to form a 80 dimensional texture feature vector. Histogram based visual features are found to work well for stationary scene boundary detection [12], [20], [38], [39]. For calculating the color and texture similarity, we adopt the method of [20]. However, for action scenes, these visual features cannot work when scene changes are encountered very frequently and it may result in over-segmentation. Hence, histogram based motion activity analysis is required for action scene detection. The above visual similarities do not take into account the shot semantics. So, we also consider semantic similarity between shots in this work. Semantic similarity between documents is addressed using the Bag of Words Model [30]. Bag of visual words model for a video captures semantic meaning which can improve clustering of shots [31], [32]. We compute visual words using K-means clustering on SIFT features [33] extracted from all shot representative frames of a movie. Each visual word is represented by a cluster. A visual word w_j appears in a shot s_i if there exists some SIFT feature points in the shot representative frame within the j-th cluster. Finally, a shot is represented by:

$$S = \nu_1, \nu_2, ..., \nu_j, ..., \nu_k \tag{7}$$

In (7), ν_j represents the normalized frequency of the j-th visual word and k is the total number of visual words/clusters. We consider 1,00,000 SIFT features of a movie and group them into 1000 clusters. Similar number of visual words is also used by Kumar et al. [23], where the authors clustered the sift features into 1500 visual words. *SemanticSim* function captures the semantic similarity between two shots i and j and is given by:

$$SemanticSim\,(i,j) = 1 - \sum_{l=1}^{k} min\,(\nu_{il}, \nu_{jl}) \tag{8}$$

where ν_{il} is the normalized frequency of the l-th visual word in shot i and k is the total number of visual words. A general shot similarity matrix is represented by:

$$W\,(i,j) = e^{-a*Sim(i,j)} \tag{9}$$

In (9), $Sim\,(i,j)$ represents the similarity function between any two shots i and j and a is a control parameter [26].

C. Spectral Grouping of Shots

Given n movie shots $s_1, s_2, \ldots, s_n$, m similarity matrices $W_k, (k = 1, \ldots, m), w_{i,j;k}$ denotes the similarity between s_i and s_j for the k^{th} feature. Let $V = [v_1, v_2, \ldots, v_m]$ be a weight vector acting as selectors for the similarities. The objective of multi-similarity spectral clustering [25] is to divide these movie shots into c clusters by finding n indicators which minimizes:

$$\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} v_k^2 w_{i,j;k} \left\| f_i - f_j \right\|^2 \tag{10}$$

where $f_i \in \mathbb{R}^c$ represent the cluster indicator variable for the i-th movie shot. We now incorporate a temporal continuity constraint within the above objective function to address temporal cohesion of the shots. The constrained objective function is given by:

$$\begin{aligned} J = &\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} v_k^2 w_{i,j;k} \left\| f_i - f_j \right\|^2 \\ +\delta &\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} v_k^2 \left| w_{i,j;k} - w_{i,l;k} \right| \varphi_{jl} \left\| f_i - f_j \right\|^2 \end{aligned} \tag{11}$$

where δ is a weight parameter for the second term of the objective function and φ_{jl} accounts for temporal integrity

between movie shots j and l. The temporal integrity function must satisfy the following properties while grouping the movie shots:
(1) $\varphi_{jl} = 1$, if $|j-l| = 0$, indicates that the same shot must be in one cluster.
(2) $\varphi_{jl} \to 0$, if $|j-l| \to \infty$, means that if two shots are very far in time order, then the effect of one shot on the other is negligible. In other words, grouping the first and the last shot of a movie into one cluster is highly unacceptable.
(3) φ_{jl} increases when $|j-l|$ becomes smaller, means that neighboring shots that are close in time order to a specific shot, have more effect on clustering as compared to farther shots. We choose the following temporal integrity function which satisfies the above properties:

$$\varphi_{jl} = e^{-|j-l|} \tag{12}$$

This objective function is minimized under the constraint that the weighted sum of v_k's p-norm is normalized, i.e.,

$$\sum_{k=1}^{m} v_k^p = 1; \;\; 1 \le p \le 2, \;\; v_k \ge 0 \tag{13}$$

In addition, for satisfying normalized spectral clustering, we require $f^T D f = 1$, where D is the diagonal matrix with its i^{th} diagonal element being the sum of i^{th} row of W_k. Mathematically, that is expressed as:

$$f^T D f = \sum_{k=1}^{m} \alpha_k v_k^2 = 1 \tag{14}$$

where $\alpha_k = f^T D f$. The goal is to minimize (11) subject to constraints (13) and (14). We construct the corresponding unconstrained objective function by applying Lagrange multipliers:

$$\begin{aligned} J_{\lambda_1,\lambda_2} &= \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} v_k^2 w_{i,j;k} \left\| f_i - f_j \right\|^2 \\ &+\delta \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} v_k^2 \left| w_{i,j;k} - w_{i,l;k} \right| \varphi_{jl} \left\| f_i - f_j \right\|^2 \\ &-\lambda_1 \left(\sum_{k=1}^{m} v_k^p - 1 \right) - 2\lambda_2 \left(\sum_{k=1}^{m} \alpha_k v_k^2 - 1 \right) \end{aligned} \tag{15}$$

Note that in (15), there are two sets of variables, indicators f_i and weights v_k. A good strategy is to solve one set of variables at a time while fixing the other set of variables [25].

Case 1. Weights v_k are given; the goal is to determine indicators f_i. If the weights v_k are given, the problem becomes a standard spectral clustering problem and the similarities are set as $w(i,j) = \sum_k v_k^2 w_{i,j;k}$. Thus, the indicators f_i can be determined from the eigenvectors of the Laplacian Matrix [34].

Case 2. Indicators f_i are known; the goal is to find weights v_k. Let us first assume that indicators f_i are given and fixed. By taking partial derivative of (15) with respect to v_k^p and setting them to zero, we have

$$\begin{aligned} &\frac{\partial J_\lambda}{\partial v_K^p} = \\ &\frac{2}{p} v_k^{2-p} \left(\begin{array}{c} \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j;k} \left\| f_i - f_j \right\|^2 \\ +\delta \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} \left| w_{i,j;k} - w_{i,l;k} \right| \\ *\varphi_{jl} \left\| f_i - f_j \right\|^2 - 2\lambda_2 \alpha_k \end{array} \right) - \lambda_1 = 0 \end{aligned} \tag{16}$$

For simplification, let

$$\begin{aligned} \beta_k &= \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j;k} \left\| f_i - f_j \right\|^2 \\ &+ \delta \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} \left| w_{i,j;k} - w_{i,l;k} \right| \varphi_{jl} \left\| f_i - f_j \right\|^2 \end{aligned}$$

The solution becomes

$$v_k = \left(\frac{\lambda_1}{2p^{-1}} \right)^{\frac{1}{2-p}} \left(\beta_k - 2\lambda_2 \alpha_k \right)^{\frac{1}{2-p}} \tag{17}$$

It is difficult to solve (17) as v_k is dependent on two variables λ_1 and λ_2. So, we first solve for λ_1 and λ_2. Substituting the above in (13), we obtain:

$$\lambda_1 = \left(\frac{p}{2} \right)^{-1} \left[\sum_{k=1}^{m} \left(\beta_k - 2\lambda_2 \alpha_k \right)^{\frac{-p}{2-p}} \right]^{-\left(\frac{2-p}{p}\right)} \tag{18}$$

Furthermore, from (14) and (17), we have

$$\left(\lambda_1\right)^2 = \left(\frac{p}{2} \right)^{-2} \left[\sum_{k=1}^{m} \alpha_k \left(\beta_k - 2\lambda_2 \alpha_k \right)^{\frac{-2}{2-p}} \right]^{-(2-p)} \tag{19}$$

Replacing λ_1 in the above equation from (18), we have

$$\begin{aligned} &\left[\sum_{k=1}^{m} \left(\beta_k - 2\lambda_2 \alpha_k \right)^{\frac{-p}{2-p}} \right]^{-2\left(\frac{2-p}{p}\right)} \\ &\qquad = \left[\sum_{k=1}^{m} \alpha_k \left(\beta_k - 2\lambda_2 \alpha_k \right)^{\frac{-2}{2-p}} \right]^{-(2-p)} \end{aligned} \tag{20}$$

Note that (20) contains only one variable λ_2. Thus, we now have a 1-D search problem which can be solved by Newton-Raphson method [35]. After finding λ_2, we obtain λ_1 from (18). Finally, v_k can be determined from (17). In our current work, we set $p = 1$ [25] and $m = 4$ (four similarity matrices). We have experimentally chosen $\delta = 0.5$ for all the video segments. This alternating process of determining indicators f_i and weights v_k is repeated till the convergence is reached.

As discussed in the theoretical foundations section, Nyström extension can be applied to find the approximated eigenvectors in spectral grouping. In the following section, we show the application of Nyström extension for approximated eigenvector computation in multi-similarity spectral clustering of movie shots. Let W_T denote the combined matrix which can be represented as addition of individual scalar multiplied-similarity matrices. Then, we can write:

$$W_T = v_1 W_{CS} + v_2 W_{MS} + v_3 W_{TS} + v_4 W_{SS} \tag{21}$$

where $(v_1, v_2, v_3, v_4 > 0)$

$$W_T = v_1 \begin{bmatrix} A_{CS} & B_{CS} \\ B_{CS}^T & C_{CS} \end{bmatrix} + v_2 \begin{bmatrix} A_{MS} & B_{MS} \\ B_{MS}^T & C_{MS} \end{bmatrix} + v_3 \begin{bmatrix} A_{TS} & B_{TS} \\ B_{TS}^T & C_{TS} \end{bmatrix} + v_4 \begin{bmatrix} A_{SS} & B_{SS} \\ B_{SS}^T & C_{SS} \end{bmatrix} \quad (22)$$

where A_{CS} and B_{CS} represent the pair-wise similarity matrix between sampled movie shots and similarity matrix between sampled shots and remaining movie shots. Using the properties of matrix algebra [3],

$$\begin{aligned} &(v_1 B_{CS} + v_2 B_{MS} + v_3 B_{TS} + v_4 B_{SS})^T \\ &= v_1 B_{CS}^T + v_2 B_{MS}^T + v_3 B_{TS}^T + v_4 B_{SS}^T \end{aligned} \quad (23)$$

The above equation can be represented as follows:

$$W_T = \begin{bmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{B}^T & \widetilde{C} \end{bmatrix} \quad (24)$$

where $\widetilde{A} = v_1 A_{CS} + v_2 A_{MS} + v_3 A_{TS} + v_4 A_{SS}$, $\widetilde{B} = v_1 B_{CS} + v_2 B_{MS} + v_3 B_{TS} + v_4 B_{SS}$ and $\widetilde{C} = v_1 C_{CS} + v_2 C_{MS} + v_3 C_{TS} + v_4 C_{SS}$. Using Nyström extension, we approximate W_T by $\widehat{W}_T$ in the following manner:

$$\widehat{W}_T = \begin{bmatrix} \widetilde{A} \\ \widetilde{B}^T \end{bmatrix} \widetilde{A}^{-1} \begin{bmatrix} \widetilde{A} & \widetilde{B} \end{bmatrix} \quad (25)$$

As discussed earlier, the eigenvectors in Nyström approximated spectral clustering can be computed in two different ways depending on the definiteness of $\widetilde{A}$. In our movie scene detection problem, the matrix $\widetilde{A}$ is positive definite. Hence, the orthogonal approximate eigenvectors are obtained by

$$\widehat{V} = \begin{bmatrix} \widetilde{A} \\ \widetilde{B}^T \end{bmatrix} \widetilde{A}^{-1/2} U_S \wedge_S^{-1/2} \quad (26)$$

where $S = \widetilde{A} + \widetilde{A}^{-1/2} \widetilde{B} \widetilde{B}^T \widetilde{A}^{-1/2}$ with eigen decomposition $S = U_S \wedge_S U_S$. But, one of the most important aspects of Nyström extension is sampling of shots to extrapolate the complete grouping solution. For this purpose, we adopt random sampling based cross-validation approach to select the landmark shots that will give low clusterability difference of eigenvectors [24]. Now, we show all the steps of our spectral clustering approach in form of an algorithm.

Algorithm 1: Spectral Grouping of Shots
Given n number of shots s_i, m similarity matrices between sampled shots and all other shots ($|AB| = m = 4$), group the shots into c clusters (number of scenes).
Procedure (s_i, m, c)
1: Initialize the weights $v_k = 1/m$
2: **Repeat**
3: ▷ Fix weights and find indicators f_i▷
4: Assume $W_T = \sum_{k=1}^{m} v_k^2 w_{i,j;k}$
5: Find approximated eigenvectors $V_2 \dots V_{c+1}$ using Nyström extension.
6: Indicator f_i = the i^{th} row of $[V_2 \dots V_{c+1}]$
7: ▷ Fix indicators f_i and find weights v_k▷
8: Solve a 1-D search problem of λ_2 in (20)
9: Obtain λ_1 by substituting λ_2 in (18)
10: Weight $v_k = \left(\frac{\lambda_1 p}{2}\right)^{\frac{1}{2-p}} (\beta - 2\lambda_2 \alpha_k)^{\frac{-1}{2-p}}$
11: **Until** Convergence
12: Run K-means on $f_1, f_2, \dots, f_n$ to group the shots into c clusters.
End Procedure

D. Cluster Sequence Analysis

Once the clustering algorithm has grouped the shots into c clusters, a label is assigned to each shot according to the cluster it belongs to. This shot cluster sequence is then analyzed to detect the scene boundaries. A scene boundary exists when two adjacent shot labels are different. The optimal number of scenes (c^*) required for spectral grouping of shots is obtained using MDL principle [36]. However, under-segmentation can happen in the case where $c < c^*$, and over-segmentation in the case $c > c^*$. Hence, to select more robust boundaries, we perform the clustering repeatedly with number of clusters c around the optimal number (c^*) and follow a majority voting procedure as follows:

$$Vote\,(i) = \frac{1}{(2 \times Tw) + 1} \sum_{c=c^*-Tw}^{c=c^*+Tw} Boundary^{(c)}\,(i) \quad (27)$$

where Tw is the temporal window size and $Boundary(i) = 1$ if the i^{th} shot is selected as scene boundary; and set to zero otherwise. In experiments, the value of Tw is set as 4 [36]. The true (i.e., final) scene boundary is detected at shot i if $Vote\,(i)$ is above the threshold of of 0.55 (i.e., 5 out of 9).

Pseudocode: Cluster Sequence Analysis
Input: Temporal window size, Tw
Number of movie shots, n
Output: Final scene boundaries
1: Compute the optimal number of scenes (c^*) using MDL principle.
2: **For** $c = c^* - Tw : c^* + Tw$
3: Cluster all the movie shots into c groups using Algo.1.
4: **For** $i = 1 : n$
5: Set $Boundary^{(c)}(i) = 1$ if i-th and $(i+1)$-th shot are assigned to different cluster; set to zero otherwise.
6: **End For**
7: **End For**
8: **For** $i = 1 : n$
9: Compute $Vote(i)$ using (27).
10: **End For**
11: Assign final scene boundaries at shot i if $Vote(i) \geq \frac{Tw+1}{(2 \times Tw)+1}$.

E. Computatonal Complexity

The proposed method has some advantages for computational complexity. Let n be the total number of shots and n_s be the number of Nyström sampled shots (where $n_s << n$). Following [24], we can write the Nyström approximation takes $O(n_s^3) + O(nn_s)$ operations to build one similarity matrix. The time-complexity of spectral clustering with n shots is $O(n^3)$. So, the overall complexity of our spectral clustering method with Nyström sampling being used for approximating the similarity matrices is: $(O(n_s^3) + O(nn_s)) + O(n^3) = O(n^3)$. Considering the post-processing part (cluster sequence

analysis), the total computational complexity of our proposed method is $O((2 * Tw + 1) * n^3) = O(n^3)$, where is the temporal window size and $Tw << n$. In this paper, we have set Tw to 4 throughout the experiments. Please note that if the Nyström approximation was not used, the complexity of generating a single similarity matrix would have been: $O(n^2) >> O(n_s^3) + O(nn_s)$, the complexity of the same with the Nyström approximation. Usually computation of more than one similarity matrices is necessary to achieve better clustering. For example, in the present paper, we have used 4 similarity matrices, namely, color, texture, motion, and semantic. With the necessity to compute more similarity matrices for large datasets like the movies, it is imperative that the computational benefit with the Nystrom approximation is even more pronounced.

III. Experimental Results and Discussion

In this section, the proposed method is analyzed and compared with two recent graph-theoretic approaches [13], [20]. Five Hollywood movies (without commercials) from the Internet Movie Database [37] (three of which are also used in [20]) are chosen for performance evaluation (see Table I). The authors in both [13] and [20] have created their own ground-truths. Since movie scene is somewhat a subjective concept and is a problem of general interest, we invite five people from different backgrounds (two film study experts, three graduate students) to create the ground-truths for us. We compare each such ground truth with the algorithmically detected result using F_1 measure [11]-[13].

To determine if a detected scene is correct or not, the Best match method [12] is adopted with a sliding window of τ shots as the tolerance factor. The detected scene boundary is regarded as true positive if the offset is less than the tolerance factor τ. We report here mean F_1 value for each movie dataset.

A. Performance evaluation of different similarity matrices

We first evaluate the efficiency of the proposed combination of multiple similarities over a single similarity. The results are shown in Table II. All the reported F_1 measures are calculated with tolerance factor $\tau = 4$. An interesting observation is that combination of color and edge similarity provides better scene detection performance for Biographic/Drama movies (Video ID #1, 3, 5) whereas the color and motion combination provides better detection accuracy for action movies (Video ID #2, 4). It can be seen from Table II that the proposed combination of similarity measures (shown in bold) easily provides best scene detection performance for all movie independent of their genre. We have used the value of the parameter a (in (3)) as 0.1 [26]. We also show the effect of changing the value of the control parameter a on different shot similarities. From Fig. 3(a-d), it can be noticed that the F_1 measure changes only marginally with change in a, which indicates the adopted form of the similarity measures are quite robust.

B. Performance evaluation of Nyström approximation

In this section, we make a comparative performance analysis of our method with and without Nyström approximation based on both execution time ratio and F_1 measure. Our proposed graph clustering based movie scene segmentation method has following computational steps:

1. Shot detection and representation
2. Shot similarities computation
3. Clustering methodology
4. Scene boundaries determination step

Steps 1 and 4 are independent of Nyström approximation whereas steps 2 and 3 are dependent on the approximation. Motivated by [20], we show a comparative performance analysis of our work with and without Nyström approximation based on the execution time ratio over the minimum value indicated by 1.0 (only steps 1 and 4 are included) in Table III. The results given in Table III show that our method with the Nyström approximation significantly reduces the execution time (mean value of 4.0 vs. 11.46). This is due to the fact that the complete grouping solution is efficiently approximated using Nyström extension. The same table also presents a comparative performance analysis with and without Nyström approximation based on F_1 measure. All the reported F_1 measures are calculated with tolerance factor $\tau = 4$. From Table III, we can conclude that the Nyström approximation substantially improves the execution time (mean value decreases from 11.46 to 4.0 if we use it) and only marginally affects the F_1 value (mean value increases from 0.7599 to 0.7732 if we use it). This type of speedup (about 3 fold in the present experiments) is highly important where huge amount of data like that of a movie needs to be processed. We next examine the quality of the Nyström approximation by measuring their approximation errors in terms of Frobenius norm of the difference similarity matrix with and without Nyström approximation and report the average value in Table IV. From the results, it can be observed that the approximation errors are significantly low, which once again validates the use of Nyström extension for the present problem. The proposed method on an average takes about 15 minutes to detect the scene changes in the movie datasets in Table I on a desktop PC with Intel(R) core(TM) i5-2400 processor and 8 GB of DDR2 memory.

C. Performance evaluation of Cluster Sequence Analysis

In this section, we present the effectiveness of our cluster scene analysis step using Table V. The results in Table V show that our proposed method performs better in presence of cluster scene analysis with a mean F_1 value of 0.7599 as compared to a mean F_1 value of 0.7047 when no such step is taken. The difference in result is due to the fact that the number of scenes obtained using the MDL principle [36] can result in over-segmentation or under-segmentation. Moreover, obtaining the accurate number of scenes a priori using any principle for a movie is a difficult task.

D. Performance comparison with other methods

In this section, we make a comparative performance analysis to evaluate the results of the proposed method. For that reason we have implemented two state-of-the-art methods presented in the literature. Both the works are based on graph-theoretic approaches for video scene detection. The first work is presented in [13]. This method computes visual similarity between shots as the maximum color similarity among all possible pairs of their key-frames. The key-frames are extracted using an improved version of spectral clustering algorithm where fast global k-means algorithm is used. This

TABLE I
EVALUATION MOVIE DATASETS (SOURCE: INTERNET MOVIE DATABASE). THE TOTAL DURATION OF THE TEST SET IS 11 HOURS 33 MINUTES 17 SECONDS, CONTAINING TOTAL 10,656 VIDEO SHOTS

Video Name	Video ID	Year	Length	Detected Shots	Genre
1492:Conquest of Paradise	1	1992	02:26:52	1975	Adventure/Biography/Drama/History
Gone in Sixty Seconds	2	2000	01:48:13	2881	Action/Crime/Thriller
A Beautiful Mind	3	2001	02:06:17	1564	Biography/Drama
Kingdom of Heaven	4	2005	02:16:53	2711	Action/Adventure/Drama
The Message	5	1976	02:55:02	1525	Adventure/Biography/Drama

TABLE II
PERFORMANCE COMPARISON WITH DIFFERENT SIMILARITY MEASURES.ALL THE REPORTED F_1 MEASURES ARE CALCULATED WITH TOLERANCE FACTOR $\tau = 4$. BEST PERFORMANCES ARE SHOWN IN BOLD.

Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
Color (C)	0.6714	0.6589	0.7053	0.6865	0.7241	0.6892
Color + Edge (C+E)	0.6828	0.6775	0.7096	0.7099	0.7294	0.7018
Color + Motion (C+M)	0.6441	0.7051	0.7046	0.7340	0.7261	0.7027
Semantic (S)	0.6745	0.6853	0.7059	0.7045	0.7272	0.6994
C+E+M+S	**0.7176**	**0.7283**	**0.7486**	**0.7984**	**0.8069**	**0.7599**

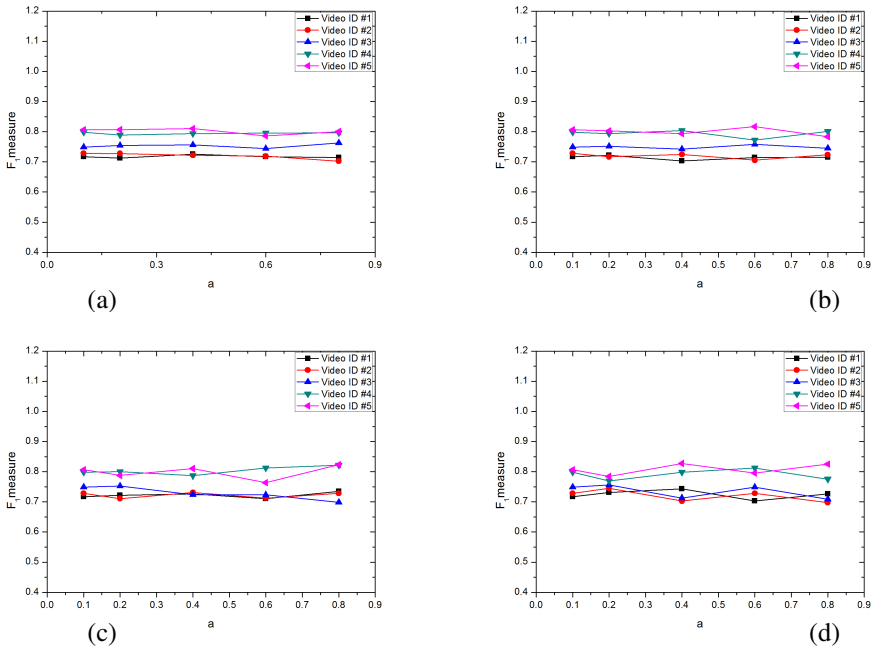

Fig. 3. Effect of varying control parameter a in similarity matrices. (a) Visual similarity (b) Texture similarity (c) Motion similarity (d) Semantic similarity. As can be seen, F_1 measure changes only marginally with change in a, which indicates the adopted form of the similarity measures are quite robust.

TABLE III
PERFORMANCE COMPARISON WITH/WITHOUT NYSTRÖM APPROXIMATION. ALL THE REPORTED F_1 MEASURES ARE CALCULATED WITH TOLERANCE FACTOR $\tau = 4$.

Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
F_1 Measure Comparison						
With Nyström Approximation	0.7176	0.7283	0.7486	0.7984	0.8069	0.7599
Without Nyström Approximation	0.7352	0.7334	0.7502	0.8153	0.8321	0.7732
Execution Time Ratio Comparison						
With Nyström Approximation	3.2	6.1	2.3	6.7	1.7	4.0
Without Nyström Approximation	9.4	16.7	9.6	14.2	7.4	11.46

TABLE IV
AVERAGE NYSTRÖM APPROXIMATION ERROR

Video ID	Error
1	$(2.15 \pm 0.72) \times 10^{-2}$
2	$(1.09 \pm 0.12) \times 10^{-1}$
3	$(1.67 \pm 0.78) \times 10^{0}$
4	$(2.64 \pm 0.08) \times 10^{-2}$
5	$(7.15 \pm 2.43) \times 10^{-2}$

comprehensive shot similarity calculation is not feasible for movie data set. Moreover, shots are grouped into clusters using the same spectral clustering where the number of clusters is estimated based on the magnitude of the eigenvalues of the similarity matrix. Finally, a sequence alignment procedure was applied over shot sequence labels to detect the scene boundaries. However, determination of window size, threshold for global minimum selection in scoring function profile, and the weight parameter α is a tedious task. We have implemented and tested this method using the same movie data set for different values of the above parameters. In our comparisons we found distinct values for each movie that provide the best performance. It can be noticed that F_1 measures reported in [13] are in the range of 0.85 to 0.90 as the method is tested for small duration video clips. But, it is not possible to get this range of values for large movie datasets as there exist wide content variation across any movie due to various dynamics of alternating sequences in addition to the movie editing effects.

The second method has been proposed in [20]. This method clusters shots into groups taking into account both color and motion similarity of video shots. Moreover, the temporal shot similarity function is also used along with visual similarity to cluster the shots into groups. This method is based on the idea of finding dominant movie scene boundary using Dominant sets framework [30]. After determining the most probable movie scene in the first round, two partitioning strategies are examined to obtain the boundaries of the remaining scenes: a treebased approach (TBM) and an order-based approach (OBM). As reported by Sakarya et al. [20], TBM is preferred over OBM from a tradeoff between F-measure performance and computational complexity. Hence, we have implemented and tested TBM method using the same movie data set and human generated ground truths. We set the parameters as $r = 2.24, c = 7$ and different values of d for different movies [20]. It must be noted that the F_1 measures reported in [20] are different to the values presented in our performance comparison as former F_1 measures are according to the frame level comparison of clusters and their ground truth results are not available to make a fair comparison.

Table VII presents a comparative performance analysis of our proposed method, methods in [13] and [20] with varying tolerance factor (i.e., τ varied from 1 to 10). The reported F_1 measure values represent the average of F_1 measures obtained from comparing the results of different methods with the ground-truths of each movie dataset. From Table VII, it is observed that our proposed method outperforms both [13] and [20] even with low τ which indicates that the proposed method is able to achieve more precise boundaries. In sharp contrast to our results, the outputs of [13], [20] suffer from both over/under segmentation problems due to inaccurate scene detections. From Table VII, we can conclude by considering all values of the tolerance factor that the proposed method clearly outperforms [13] and [20] in as many as 46 out of a total of 50 cases.

In order to further demonostrate the superiority of our clustering methodology, we have included comparisons with two popular clustering approaches, namely, a graph partitioning algorithm based on spectral factorization [55] and nearest neighborhood (NN) [56] clustering. Only color feature is used and the cluster scene analysis step is kept the same. Table VI presents the mean F_1 measure obtained by applying both the approaches for five movies. The results clearly indicate that the proposed method performs significantly better than the graph paritioning algorithm and NN clustering for all the movie segments.

E. Performance comparison of methods with same similarity measures

Boundaries between scenes in the previous subsections are determined by the Nyström approximated multi-similarity spectral clustering. In order to explicitly verify the superiority of our clustering methodology, we include a comparison of our method with that of [13], [20] using the same shot similarity matrix. Table VIII shows a performance analysis of our method using same shot similarity measures as in [13], [20]. Table VIII also represents a comparison of these methods in terms of execution time ratio over the minimum value indicated by 1.0 (only step 4 of section III.B is included). The reason for only inclusion of step 4 instead of both step 1 and 4 as in section III.B, is due to the fact that only step 1 is common to all of the three compared methods. Proposed method (C) uses only color similarity as in [13] whereas Proposed Method (C+M) uses both color and motion similarity as in [20]. The same shot detection and representation (i.e., the middle frame) are used in the pre-processing steps for all the implementations. On considering the mean values of F_1 measures in Table VIII, the performance of Proposed Method (C) and [13] are close to each other. However, Proposed Method (C+M) clearly outperforms [20]. The reason for the small performance difference between Proposed Method (C) and Chasenis et. al. [13] is due to the fact that both the methods are based on spectral clustering. The superiority of our method is due to the formulation of spectral clustering with temporal integrity constraint. On the other hand, the reason for the superior performance over the method by Sakarya et. al.[20] is due the robustness of our clustering strategy with temporal information. At the same time, on considering the execution time ratio, our method performs quite well as compared to both of the methods. Specifically, our clustering strategy uses Nyström approximated eigenvectors that substantially reduces the computational burden in detecting precise scene boundaries from long duration movies.

IV. CONCLUSION

In this paper, we presented a novel method for high-level segmentation of movies into scenes using Nyström approximated multi-similarity spectral clustering with a temporal integrity constraint. Multiple shot similarity matrices are used to model the diverse characteristics of different types of movie

TABLE V
IMPACT OF CLUSTER SCENE ANALYSIS. MEAN F_1 VALUE INCREASES FROM 0.7047 TO 0.7599.

Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
Number of Scenes/Clusters (c^*)	92	76	98	112	64	-
Without Cluster Scene Analysis	0.6814	0.6975	0.7169	0.7177	0.7103	0.7047
With Cluster Scene Analysis	0.7176	0.7283	0.7486	0.7984	0.8069	0.7599

TABLE VI
MEAN F_1 PERFORMANCE COMPARISON WITH NN CLUSTERING AND A SPECTRAL FACTORIZATION BASED GRAPH PARTITIONING ALGORITHM. ALL THE REPORTED VALUES ARE COMPUTED USING ONLY COLOR FEATURE AND THE CLUSTER SCENE ANALYSIS STEP IS KEPT SAME FOR ALL THE RESULTS. THE RSULTS CLEARLY INDICATE THE SUPERIORITY OF OUR CLUSTERING COMPARED TO BOTH NN CLUSTERING AND SPECTRAL FACTORIZATION BASED GRAPH PARTITIONING. BEST VALUES ARE SHOWN IN BOLD.

Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
Proposed Method	**0.6714**	**0.6589**	**0.7053**	**0.6865**	**0.7241**	**0.6892**
Graph Partitioning	0.4321	0.5672	0.5863	0.6037	0.6170	0.5612
NN Clustering	0.3771	0.3684	0.4255	0.3720	0.4962	0.4078

scenes. Comprehensive experimentations clearly indicate that the superiority of the proposed method over some recently published works. In future, we will focus on integration of more extensive set of video features to further improve the scene detection results. Another direction of future research will be to assign semantic labeling to the detected movie scenes for more effective movie navigation.

APPENDIX

In the following section, we will show the positive definiteness of combined matrix $\widetilde{A}$.

• Let X be a non empty set. A function (likewise for matrix) $K : X \times X \rightarrow R$ is called positive definite if and only if it is symmetric i,e., $K(x, x^{'}) = K(x^{'}, x)$ for all $x, x^{'} \in X$ and if for an arbitrary finite non-zero vector c:

$$C^T K_{ij} C = \sum_{i,j=1}^{n} C_i C_j K(x_i, x_j) > 0; \tag{28}$$

for $x_1, \ldots, x_n \subseteq X; C_1, \ldots, C_n \subseteq R$.

• Multiplication of a finite positive constant with a positive definite function or matrix is also positive definite. In other words, if K is positive definite then $v * K$ is also positive definite $(v > 0)$.

• Exponential of a positive definite function is also positive definite, i.e., if K is positive definite, then $exp(K)$ is also positive definite.

Lemma 1. Individual similarity matrices W_{CS}, W_{MS}, W_{TS} and W_{SS} are positive definite. (Please see (9))

Proof :

$$\begin{aligned} W_{CS}(i,j) &= e^{-a * ColorSim(i,j)}, a > 0 \\ &= e^{-a\left[1 - \sum_{h=1}^{m} min(H_i(h), H_j(h))\right]} \\ &= e^{-a + a \sum_{h=1}^{m} min(H_i(h), H_j(h))} \\ &= e^{-a} \cdot e^{a \sum_{h=1}^{m} min(H_i(h), H_j(h))} \\ &= v \cdot e^{a \sum_{h=1}^{m} min(H_i(h), H_j(h))}; v > 0 \end{aligned}$$

The above Equation is positive definite if $\sum_{h=1}^{m} min(H_i(h), H_j(h))$ function is positive definite. The function $K(H_i, H_j) = \sum_{h=1}^{m} min(H_i(h), H_j(h))$ is a positive definite function or Mercers kernel. Hence, the matrix W_{CS} is positive definite. Similarly, it can be shown that other similarity matrices are also positive definite.

Lemma 2. The combined similarity matrix $\widetilde{A}$ is positive definite. (Please see (21))

Proof :

$$\widetilde{A} = v_1 A_{CS} + v_2 A_{MS} + v_3 A_{TS} + v_4 A_{SS}$$

$\widetilde{A}$ is positive definite iff $C^T \widetilde{A} C > 0$ i,e.,

$$C^T(v_1 A_{CS} + v_2 A_{MS} + v_3 A_{TS} + v_4 A_{SS})C > 0$$

LHS of the inequality can also be written as:

$$C^T(v_1 A_{CS})C + C^T(v_2 A_{MS})C + C^T(v_3 A_{TS})C + C^T(v_4 A_{SS})C$$

Each individual component, $C^T(v_1 A_{CS})C > 0$ (from Lemma 1). Hence, $\widetilde{A}$ is Positive Definite.

REFERENCES

[1] M. Naaman, Social Multimedia: Highlighting Opportunities for Search and Mining of Multimedia Data in Social Media Applications, *Multimedia Tools and Applications*, vol. 56, pp. 9−34, 2012.

[2] W. Gao, Y. Tian, T. Huang and Q. Yang, Vlogging: A Survey of Video Blogging Technology on the Web, *ACM Computing Survey*, vol. 42, pp. 1−57, 2010.

[3] Y. Pang, H. Yan, Y. Yuan, and K. Wang, Robust CoHOG Feature Extraction in Human-Centered Image/Video ManagementSystem,IEEE Trans. Syst., Man, Cybern. B, Cybern vol. 42, no. 2, pp. 458-468, April 2012.

[4] S.E.F. Avila, A.P.B. Lopes, A. Luz Jr and A.A. Araujo, VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method, *Pattern Recognition Letters*, vol. 32 (1), pp. 56−68, 2011.

[5] S. K. Kuanar, R. Panda and A.S. Chowdhury, Video Key frame Extraction through Dynamic Delaunay Clustering with a Structural Constraint, *Journal of Visual Communication and Image Representation*, vol. 24 (7), pp. 1212−1227, 2013.

[6] J. Almeida, N.J. Leite and R.S. Torres, VISON: video summarization for online applications, *Pattern Recognition Letters*, vol. 33 (4), pp. 397−409, 2012.

[7] B. Han, J. Hamm and J. Sim, Personalized video summarization with human in the loop, *In Proc. of the IEEE Workshop on Applications of Computer Vision*, 2012.

[8] R. Panda, S. K. Kuanar and A.S. Chowdhury, Scalable Video Summarization using Skeleton Graph and Random Walk, *In Proc. of International Conf. on Pattern Recognition*, pp. 3481−3486, 2014.

TABLE VII
MEAN F_1 MEASURE COMPARATIVE PERFORMANCE ANALYSIS OF DIFFERENT METHODS WITH VARYING TOLERANCE FACTOR. PROPOSED METHOD OUTPERFORMS [13] AND [20] IN 46 OUT OF 50 CASES. BEST VALUES ARE SHOWN IN BOLD.

Tolerance Factor τ	Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
1	Proposed Method	**0.5529**	**0.5714**	0.5123	**0.4924**	**0.6726**	**0.5603**
	Chasenis *et al.* [13]	0.5089	0.5217	**0.5225**	0.4712	0.5252	0.5099
	Sakarya *et at.* [20]	0.4915	0.5396	0.4815	0.4356	0.5138	0.4924
2	Proposed Method	**0.6235**	**0.5934**	**0.5775**	**0.5517**	**0.7563**	**0.6204**
	Chasenis *et al.* [13]	0.5799	0.5548	0.5340	0.5489	0.5788	0.5592
	Sakarya *et at.* [20]	0.5586	0.5819	0.5025	0.4785	0.5765	0.5396
3	Proposed Method	**0.6823**	0.6493	**0.6737**	**0.6924**	**0.7825**	**0.6960**
	Chasenis *et al.* [13]	0.6153	0.6087	0.6387	0.6327	0.6021	0.6195
	Sakarya *et at.* [20]	0.6033	**0.6543**	0.6153	0.6144	0.6359	0.6246
4	Proposed Method	**0.7176**	**0.7283**	**0.7486**	**0.7984**	**0.8069**	**0.7599**
	Chasenis *et al.* [13]	0.6745	0.6412	0.7120	0.6690	0.6850	0.6763
	Sakarya *et at.* [20]	0.6368	0.6582	0.7076	0.6877	0.6971	0.6774
5	Proposed Method	**0.7529**	**0.7472**	**0.7486**	**0.8007**	**0.8251**	**0.7749**
	Chasenis *et al.* [13]	0.7100	0.7064	0.7120	0.7153	0.7058	0.7059
	Sakarya *et at.* [20]	0.6854	0.7195	0.7076	0.7348	0.7380	0.7170
6	Proposed Method	**0.7882**	**0.7802**	**0.7701**	**0.8249**	**0.8394**	**0.8005**
	Chasenis *et al.* [13]	0.7692	0.7333	0.7434	0.7356	0.7265	0.7416
	Sakarya *et at.* [20]	0.7374	0.7407	0.7384	0.7459	0.7328	0.7390
7	Proposed Method	**0.8235**	**0.8241**	**0.8021**	**0.8327**	**0.8480**	**0.8260**
	Chasenis *et al.* [13]	0.8047	0.7934	0.7853	0.7544	0.7549	0.7785
	Sakarya *et at.* [20]	0.8044	0.7936	0.7794	0.7521	0.7526	0.7764
8	Proposed Method	**0.8705**	**0.8571**	**0.8449**	**0.8568**	**0.8766**	**0.8611**
	Chasenis *et al.* [13]	0.8047	0.8260	0.8272	0.8369	0.7867	0.8163
	Sakarya *et at.* [20]	0.8156	0.8464	0.8102	0.8085	0.8092	0.8179
9	Proposed Method	**0.8941**	0.8901	**0.8770**	**0.8767**	**0.9056**	**0.8887**
	Chasenis *et al.* [13]	0.8282	**0.8913**	0.8691	0.8415	0.8129	0.8486
	Sakarya *et at.* [20]	0.8379	0.8878	0.8205	0.8309	0.8153	0.8384
10	Proposed Method	**0.9058**	0.9120	**0.8983**	**0.9215**	**0.9287**	**0.9132**
	Chasenis *et al.* [13]	0.8520	0.9021	0.8795	0.8971	0.8450	0.8751
	Sakarya *et at.* [20]	0.8714	**0.9235**	0.8404	0.8634	0.8746	0.8746

TABLE VIII
PERFORMANCE COMPARISON WITH [13], [20] USING SAME SET OF FEATURES. OURS PERFORMS BEST IN TERMS F_1 VALUES AND AT THE SAME TIME IS ALSO FASTER AS CAN BE SEEN FROM THE EXECUTION TIME RATIOS.

Methods	Video ID #1	Video ID #2	Video ID #3	Video ID #4	Video ID #5	Mean
		F_1 Measure Comparison				
Proposed Method (C)	0.6714	0.6589	0.7053	0.6865	0.7241	0.6892
Chasenis *et al.* [13]	0.6745	0.6412	0.7120	0.6690	0.6850	0.6763
Proposed Method (C+M)	0.6441	0.7051	0.7046	0.7340	0.7261	0.7027
Sakarya *et al.* [20]	0.6368	0.6582	0.7076	0.6877	0.6971	0.6774
		Execution Time Ratio Comparison				
Proposed Method (C)	1.1	1.6	1.3	1.7	1.4	1.4
Chasenis *et al.* [13]	3.2	4.1	6.7	7.9	5.2	5.3
Proposed Method (C+M)	3.1	4.5	2.7	5.8	1.9	3.6
Sakarya *et al.* [20]	11.2	13.1	14.2	16.3	17.1	14.3

[9] B.T. Truong and S. Venkatesh, Video abstraction: a systematic review and classification, *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 3 (1), pp. 1−37, 2007.

[10] A.G. Money and H.W. Agius, Video summarization: a conceptual framework and survey of the state of the art, *Journal of Visual Communication and Image Representation*, vol. 19 (2), pp. 121−143, 2008.

[11] Y. Zhai and M. Shah, Video Scene Segmentation Using Markov Chain Monte Carlo. *IEEE Trans. Multimedia*, vol. 8, pp. 686−697, 2006.

[12] Z. Rasheed and M. Shah, Detection and representation of scenes in videos, *IEEE Trans. Multimedia*, vol. 7, pp. 1097−1105, 2005.

[13] V.T. Chasanis, A.C. Likas and N.P. Galatsanos, Scene detection in videos using shot clustering and sequence alignment, *IEEE Trans. Multimedia*, vol. 11, pp. 89−100, 2009.

[14] M. Yeung, B.L. Yeo and B. Liu, Segmentation of video by clustering and graph analysis, *Computer Vision Image Understanding*, vol. 71, pp. 94−109,1998.

[15] Y. Zhao, T. Wang, P. Wang, W. Hu, Y. Du, Y. Zhang and G. Xu, Scene segmentation and categorization using Ncuts, *In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1−7, 2007.

[16] C. W. Ngo, Y. F. Ma, and H. J. Zhang, Video summarization and scene detection by graph modeling, *IEEE Trans. Circuits System for Video Technology*, vol. 15, pp. 296−305, 2005.

[17] U. Sakarya and Z. Telatar, Video scene detection using graph-based representations, *Signal Processing Image Communication*, vol. 25, pp. 774−783, 2010.

[18] J. M. Odobez, D. Gatica. Perez and M. Guillemot, Spectral Structuring of Home Videos, In E. Bakker, M. Lew, T. Huang, N. Sebe, and X. Zhou, editors, *Image and Video Retrieval*, volume 2728 of Lecture Notes in Computer Science, chapter 31, pp. 85−90. Springer Berlin / Heidelberg, Berlin, Heidelberg, June 2003.

[19] Z. Zhang, B. Li, H. Lu and X. Xue. Scene segmentation based on video structure and spectral methods, *In 10th International Conf. on Control, Automation, Robotics and Vision*, pp. 1093-1096, 2008.

[20] U. Sakarya, Z. Telatar and A. Aydn Alatan, Dominant sets based movie scene detection, *Signal Processing*, vol. 92, pp. 107−119, 2012.

[21] Y.-P. Tan and H. Lu, Model-based clustering and analysis of video scenes, *Proc. Int'l conf. on Image Processing*, pp. 617−620, 2000.

[22] H. Sundaram and S.F. Chang, Computable scenes and structures in films, *IEEE Trans. Multimedia*, vol. 4, pp. 482−491, 2002.

[23] Niraj Kumar, Piyush Rai, Chandrika Pulla and C V Jawahar, Video Scene Segmentation with a Semantic Similarity, *In Proc. of 5th Indian International Conference on Artificial Intelligence*, 2011.

[24] F. C. Charless Fowlkes, S. Belongie and J. Malik, Spectral grouping using Nyström method, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 214−225, 2004.

[25] H. C. Huang, Y. Y. Chuang and C. S. Chen, Multi-affinity spectral clustering, *In Proc. of International Conf. on Acoustics, Speech and Signal Processing*, pp. 2089−2092, 2012.

[26] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song and Z. H. Zhou, Multi-View Video Summarization, *IEEE Trans. Multimedia*, vol. 12, pp. 717729, 2010.

[27] Z. Cernekova, I. Pitas and C. Nikou, Information theory-based shot cut/fade detection and video summarization, *IEEE Trans. Circuits System for Video Technology*, vol. 16, pp. 82−91, 2006.

[28] G. Paschos, Perceptually uniform color spaces for color texture analysis: an empirical evaluation, *IEEE Trans. on Image Processing*, vol. 10 (6), pp. 932−937, 2001.

[29] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan and A. Yamada, MPEG-7 color and texture descriptors, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, 2000.

[30] D. Bollegala, Y. Matsuo and M. Ishizuka, A Web Search Engine Based Approach to Measure Semantic Similarity between Words, *IEEE Trans. Knowledge and Data Engineering*, vol. 23, pp. 977−990, 2007.

[31] J. Sivic and A. Zisserman, Video Google: a text retrieval approach to object matching in Videos, *In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 170−1477, 2003.

[32] N. Rasiwasia and N. M. Vasconcelos, Scene classification with low-dimensional semantic spaces and weak supervision, *In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1−6, 2008.

[33] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, vol. 60 (2), pp. 91-110, 2004.

[34] U. Von Luxburg, A tutorial on spectral clustering, *Statistics and Computing*, vol. 17, pp. 395−416, 2007.

[35] R. Baldick, Applied Optimization: Formulation and Algorithms for Engineering Systems. Cambridge University Press, 2009.

[36] A. Hanjalic, R. Lagendijk and J. Biemond, Automated high-level movie segmentation for advanced video-retrieval systems, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, pp. 580−588, 1999.

[37] http://www.imdb.com/stats.

[38] C. Siagian and L. Itti, Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 300−312, 2007.

[39] K. Huang, D. Tao, Y. Yuan, X. Li, and T. Tan, Biologically Inspired Features for Scene Classification in Video Surveillance, IEEE Trans. Syst., Man, Cybern. B, Cybern vol. 41, no. 1, pp. 307-313, Feb. 2011.

[40] A. Kowdle and T. Chen, Learning to Segment a Video to Clips Based on Scene and Camera Motion, In Europian Conference on Computer Vision, 2012.

[41] B. Wu, X. Jiang, T. Sun, S. Zhang, X. Chu, C. Shen and J. Fan, A Novel Horror Scene Detection Scheme on Revised Multiple Instance Learning Model, In Proceedings of the 17th International conference on Advances in multimedia modeling, 2011.

[42] T. Zhang, C. Xu, G. Zhu, S. Liu, H. Lu, A Generic Framework for Video Annotation via Semi-Supervised Learning, IEEE Transactions on Multimedia, 14(4), 2012.

[43] C. Liu, D. Wang, J. Zhu and B. Zhang, Learning a Contextual Multi-Thread Model for Movie/TV Scene Segmentation, IEEE Transactions on Multimedia, 15(4), 2013.

[44] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, and J. Kittler, Differential Edit Distance: A Metric for Scene Segmentation Evaluation, IEEE Transactions on Circuits and Systems for Video Technology, 22 (6), 2012.

[45] L. Baraldi, C. Grana, and R. Cucchiara, A Deep Siamese Network for Scene Detection in Broadcast Videos, In Proceedings of the 23rd ACM international conference on Multimedia, 2015.

[46] L. Baraldi, C. Grana, A. MESSINA, and R. Cucchiara, A Browsing and Retrieval System for Broadcast Videos using Scene Detection and Automatic Annotation, In Proceedings of the 24th ACM international conference on Multimedia, 2016.

[47] D. Teney, M. Brown, D. Kit, and P. Hall, Learning Similarity Metrics for Dynamic Scene Segmentation, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[48] J. Yu, Y. Rui and D. Tao Click Prediction for Web Image Reranking using Multimodal Sparse Coding, IEEE Transactions on Image Processing, 23(5): 2019-2032, 2014.

[49] J. Yu, Y. Rui, Y.Y. Tang and D. Tao, High-order Distance based Multi-view Stochastic Learning in Image Classification, IEEE Transactions on Cybernetics, 10.1109/TCYB.2014.2307862, 2014.

[50] J. Yu, M. Wang and D. Tao, Semi-supervised Multiview Distance Metric Learning for Cartoon Synthesis, IEEE Transactions on Image Processing, 21(11): 4636-4648, 2012.

[51] C. Xu, D. Tao and C. Xu, A survey on multi-view learning, arXiv preprint, 2013.

[52] C. Xu, D. Tao and C. Xu, Large-margin multi-view information bottleneck, IEEE Transactions on on Pattern Analysis and Machine Intelligence, 36(8): 1559-1772, 2014.

[53] X. Cai, F. Nie and H. Huang, Multi-View K-Means Clustering on Big Data, In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

[54] H. Wang, F. Nie and H. Huang, Multi-View Clustering and Feature Learning via Structured Sparsity , In Proceedings of 30th International Conference on Machine Learning, 2013.

[55] J. Hespanha, An efficient MATLAB Algorithm for Graph Partitioning, Technical Report, University of California, 2004.

[56] M. Anthony. Wong, and Tom Lane. A kth nearest neighbour clustering procedure, Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface. Springer US, 1981.

Rameswar Panda received the M.E degree in Electronics Telecommunication engineering from Jadavpur University, Kolkata, India in 2013. His current research interests include computer vision, multimedia analysis, and pattern recognition.

Sanjay K. Kuanar received the Ph.D and M.Tech. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India in 2015 and 2007 respectively. His current research interests include pattern recognition, multimedia analysis and computer vision.

Ananda S. Chowdhury earned his Ph.D. in Computer Science from the University of Georgia, Athens, Georgia in July 2007. From August 2007 to December 2008, he worked as a postdoctoral fellow in the department of Radiology and Imaging Sciences at the National Institutes of Health, Bethesda, Maryland. At present, he is working as an Associate Professor in the department of Electronics and Telecommunication Engineering at Jadavpur University, Kolkata, India where he leads the Imaging Vision and Pattern Recognition group. He has authored or coauthored more than forty-five papers in leading international journals and conferences, in addition to a monograph in the Springer Advances in Computer Vision and Pattern Recognition Series. His research interests include computer vision, pattern recognition, biomedical image processing, and multimedia analysis. Dr. Chowdhury is a senior member of the IEEE and the IAPR TC member of Graph-Based Representations in Pattern Recognition. He currently serves as an Associate Editor of Pattern Recognition Letters and his Erdos number is 2.