Detection and Representation of Scenes in Videos

Zeeshan Rasheed and Mubarak Shah, *Fellow, IEEE*

***Abstract*—This paper presents a method to perform a high-level segmentation of videos into scenes. A scene can be defined as a subdivision of a play in which either the setting is fixed, or when it presents continuous action in one place. We exploit this fact and propose a novel approach for clustering shots into scenes by transforming this task into a graph partitioning problem. This is achieved by constructing a weighted undirected graph called a *shot similarity graph* (SSG), where each node represents a shot and the edges between the shots are weighted by their similarity based on color and motion information. The SSG is then split into subgraphs by applying the *normalized cuts* for graph partitioning. The partitions so obtained represent individual scenes in the video. When clustering the shots, we consider the global similarities of shots rather than the individual shot pairs. We also propose a method to describe the content of each scene by selecting one representative image from the video as a scene key-frame. Recently, DVDs have become available with a chapter selection option where each chapter is represented by one image. Our algorithm automates this objective which is useful for applications such as video-on-demand, digital libraries, and the Internet. Experiments are presented with promising results on several Hollywood movies and one sitcom.**

***Index Terms*—Graph partitioning, key-frames, normalized cuts, scene, shot, video segmentation.**

I. Introduction

With digital technology becoming inexpensive and popular, there has been a tremendous increase in the availability of videos through cable and the Internet such as *video on demand*. Recently, the Berkeley "How Much Information?" project ([10]) found that 4500 motion pictures are produced annually amounting to almost 9000 hours or 16 terabytes of data *every year*. They further found that 33 000 television stations broadcast for about 16 hours a day and produce 48 million hours per year, amounting to 63 000 terabytes of data! Feasible access to this huge amount of data requires organization and efficient tools for browsing and retrieving contents of interest. Otherwise, searching a sequential tape for a specific section of a video would be time consuming as well as frustrating. Recently, DVDs options include viewing a particular scene in the movie. To obtain such a representation, a human observer is required to watch the entire video and locate the important boundaries or *scene edges*. However, manual content analysis is not feasible for a large amount of data as it is slow and expensive. This fact poses challenges to develop automatic means to locate scene boundaries and to determine a representative key-frame for each scene in the video.

In Webster's dictionary, a scene is defined as follows [20].

Definition 1: A subdivision of an act in a dramatic presentation in which the setting is fixed and the time continuous or

Definition 2: One of the subdivisions of a play; as a division of an act presenting continuous action in one place.

The first definition emphasizes the fact that shots belonging to one scene are often taken with a fixed physical setting such as inside a studio. Several cameras capture the video at different angles with repetitive structure as the background remains the same. However, this definition may not hold for all scenes; outdoor scenes may be shot with moving cameras and the background may change with time. Thus, a scene may also be defined by the continuity of ongoing actions performed by the actors. Utilizing these factors, we have developed a framework to find scene boundaries which exploits both color and motion similarities of shots. In our approach, we construct a weighted undirected graph called a *shot similarity graph* or SSG and transform the problem of scene boundary detection into a graph partitioning problem. The undirected graph consists of nodes that represent shots and are connected by weighted edges. The weights are proportional to the shot similarities and computed as a function of color and motion features of shots. The scene boundaries are detected by partitioning the SSG into subgraphs using the *normalized cuts* to maximize intra-subgraph similarities and minimize inter-subgraph similarities. Our algorithm considers *global* similarities of shots to detect boundaries rather than *local* similarities as do several related approaches (See Section II). The proposed algorithm is robust to any *local* mismatch between two shots that belong to two different scenes. We also propose a novel approach to describe the content of each detected scene by selecting one representative image from the video. The next section provides an overview of related work in this area. Section III discusses the computation of shot similarities followed by the detection of scenes using normalized cuts in Section IV. Section V discusses representing scenes with one key-frame and our experiments on several Hollywood movies and a sitcom are presented in Section VI. In this section we also compare our algorithm with a baseline algorithm proposed by Yeung *et al.* [24]. Section VII concludes our work.

II. Related Work

A large amount of work has been done to structure videos resulting in several video navigation tools for viewers. From the temporal segmentation point of view, these approaches can be loosely divided into two groups; *shot based* and *scene based*. In shot based structuring of video, the idea is to first detect shots and then represent each shot by a fixed or variable number of

Manuscript received April 19, 2003; revised September 9, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chuang Gu.

Z. Rasheed is with ObjectVideo, Inc., Reston, VA 20191 USA (e-mail: zrasheed@objectvideo.com).

M. Shah is with the School of Computer Science, University of Central Florida, Orlando, FL 32816 USA (Email: shah@cs.ucf.edu).

Digital Object Identifier 10.1109/TMM.2005.858392

frames. Zhang *et al.* [25] proposed a shot detection approach based on an image histogram which has been widely used in the research community [6], [18]. Many researchers exploited the information stored in compressed video files and achieved good accuracy at much lower computation cost, for example [11], [23]. See [9] for a comprehensive survey of shot detection algorithms.

The next, higher level of structuring videos is to cluster similar shots into *scenes*. Yeung *et al.* [24] proposed a graphical representation of videos by constructing a *scene transition graph*, STG. Each node represented a shot and the edges represented the transitions between the shots based on the visual similarity and temporal locality. They used the *complete-link* method of hierarchical clustering to split the STG into subgraphs—each representing a *scene*. Hanjalic *et al.* [7] used a similar approach for shot clustering using a graph and found *logical story units*. Linking of shots was done by defining an *inter-shot dissimilarity measure* using DCT images in MPEG compressed video. An average distance between the shots was computed in the $L^*u^*v^*$ color space and thresholded empirically. This determined whether or not two shots were part of one logical story unit. All shots in between two similar shots were merged to construct one unit. Javed *et al.* [8] proposed a framework for the segmentation of interview and game show videos. Their method automatically removed commercials and detected hosts and guests by analyzing the video structure and constructing a *shot connectivity graph*.

Ngo *et al.* [12] proposed a motion based approach to cluster similar shots to form scenes. Spatio-temporal slices of video were constructed and local orientations of pixels were computed using structure tensors. Each shot was represented either by key-frames or by constructing mosaics based on the motion pattern of slices. Finally, shots were clustered together by analyzing the shot color similarities using the color histogram intersection of key-frames or mosaics. They used motion information only to exclude moving foreground objects from mosaics and not as a cue for finding shot similarities as we have proposed. Aner *et al.* [2] proposed a similar approach, creating mosaics by registering frames of each shot and by eliminating the moving objects. They used the *rubber-sheet* matching algorithm to cluster scenes into physical settings within episodes of sitcoms as well as across episodes of the same program. However, many genres of videos, such as feature movies, are not limited to fixed physical settings, and therefore proper mosaic construction and matching is difficult to achieve.

Rui *et al.* [14] proposed the construction of a table-of-contents for videos. A time-adaptive grouping of shots was done by finding the visual similarities between them. This similarity was a function of color and activity features of the shots, weighted by their temporal locality. Shots were merged together to form groups by defining a *group threshold* and groups were merged together to form scenes by defining a *scene threshold*. Zhou *et al.* [26] exploited film editing techniques and discovered that certain regions in the video frame are more robust to noise for computing shot similarities. The clustering method was, however, similar to [7]. While [26] reported slightly better performance than [14], both approaches fail to capture the global similarities of shots in scenes. They found scene boundaries by detecting the strongly connected components in the graphs based on a one-to-one shot similarity criterion. Two major problems arise using such criterion: 1) a color match between shots of two different scenes may falsely combine the scenes, and all those in between, into one segment causing undersegmentation and 2) action scenes may be broken into many scenes for not satisfying the color matching criterion producing oversegmentation. See Fig. 1 for illustrations.

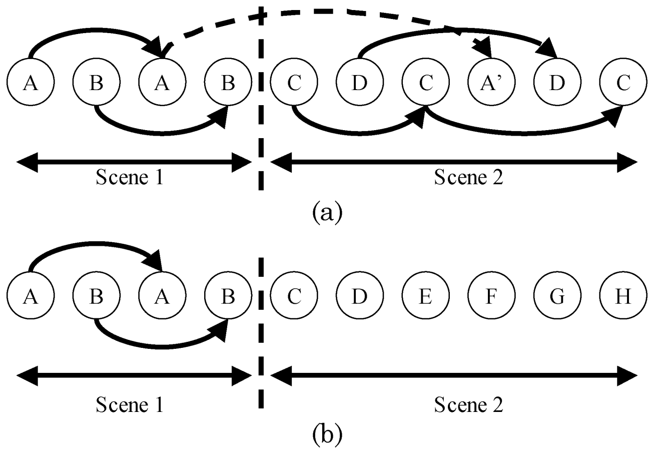

Fig. 1. Segmentation issues with existing methods. Shots are represented by circles. (a) Undersegmentation when shots of different scenes are similar. A valid scene boundary will be missed due to an erroneous shot linking indicated by the dotted link. (b) Oversegmentation due to continuously changing visual content of shots. Scene 2 is oversegmented since links cannot be established between the shots.

We believe that the detection of scenes should not be based on one-to-one shot similarity. Therefore, we consider comprehensive shot similarities rather than individual shot pairs. We incorporate the full effect of surrounding shot contexts which maximizes the intra-scene similarity between the shots of one scene and minimizes the inter-scene similarity between the shots of two different scenes. Our approach is not affected by any mismatch between shots of different scenes. It incorporates motion information together with color information of video, which makes it robust to the aforementioned problems. Recently, Obodez *et al.* [13] have also shown the use of a spectral method of grouping video shots based on their visual similarity and temporal relationship with experiments on home videos. Our approach is similar to theirs in that we also exploit the spectral structure in the affinity graph. However, it is different in that we incorporate motion similarity of shots together with color similarity which may not be consistent for a given scene of commercially created videos. We also avoid computing the eigenvectors. Instead, we use a recursive bipartitioning technique using normalized cuts. Although there are an exponential number of possible partitions of a graph with N nodes, only $N-1$ partitions are permissible as a scene must consist of temporally connected shots. This ensures that a solution is achieved in linear time. Odobez *et al.* also approximate the number of clusters before segmenting the video, however, we do not put such a constraint in our approach.

III. Detecting Shots and Finding Shot Similarities

We first divide the video into shots using a modified version of the color histogram intersection method proposed by Haering [5]. We use a 16-bin HSV normalized color histogram for each frame with eight bins for *hue* and 4 bins each for *saturation* and *value*. Shots shorter than ten frames are merged with the following shots as being erroneous. Thus, the z^{th} shot in the video can be expressed as a set of frames, that is

$$S_z = \{f^a, f^{a+1}, \ldots, f^b\}$$

where a and b are the indices of the first and last frames of the shot respectively.

A. Shot Key-Frame Detection

The next step is to extract key-frames to represent the shot's content. Choosing an appropriate number of key-frames is important because only they will be used in scene detection rather than all the frames in the video. We consider a variety of videos; selecting the middle frame may represent a static shot with little actor or camera motion well, however, a dynamic shot with higher actor or camera motion may not be represented adequately. Some existing approaches are [4], [21], and [27]. We use a variant of the algorithm presented by Zhuang *et al.* [27] which detects multiple frames based on the visual variations in shots. Let a shot, S_z, be represented by a set of key-frames, K_z. Initially, the middle frame of the shot is selected and added to an empty set K_z as the first key-frame (where the index of the middle frame of the shot is $\lfloor (a+b)/2 \rfloor$). Next, each frame within the shot is compared with frames in K_z. If it differs from all previously chosen key-frames by a fixed threshold, it is added in K_z. This algorithm can be summarized as follows.

```
STEP 1: Select middle frame as the
first key-frame
  $K_z \leftarrow \{f^{\lfloor (a+b)/2 \rfloor}\}$
STEP 2: for $r = a$ to $b$
  if $\max(ColSim(r,k)) < Th \quad \forall f^k \in K_z$
    Then $K_z \leftarrow K_z \cup \{f^r\}$
```

where Th is the minimum frame similarity threshold that declares two frames to be similar (generally in range 0.8–0.9). The $ColSim$ is defined as the color similarity between two image frames, that is

$$ColSim(x,y) = \sum_{h \in \text{bins}} \min\left(H_x(h), H_y(h)\right) \tag{1}$$

where H_x and H_y are the HSV color histogram of frames x and y respectively and $ColSim(x,y) \in [0,1]$. This method assures that every key-frame is distinct and, therefore, prevents redundancy. Fig. 2(a) and (b) shows key-frames selected for shots from a sitcom and a feature movie, respectively.

B. Shot Motion Content Feature

We associate a feature, Mot_z with each shot z, which is the motion content of the shot normalized by the number of frames in it; that is

$$Mot_z = \frac{1}{b-a} \sum_{f=a}^{b-1} \left(1 - ColSim(f, f+1)\right) \tag{2}$$

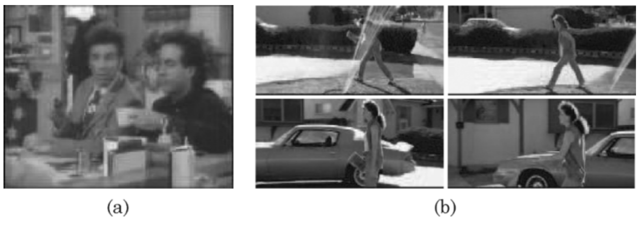
Fig. 2. Key-frame detection. (a) One frame selected for a shot in a sitcom. (b) Four frames selected for a shot from the movie "Terminator II".

where $Mot_z \in [0,1]$. Due to the temporally changing visual content in an action scene, the motion content value of the shots is much higher than shots of a dialogue scene. Therefore, it can be used as a suitable feature for shot similarity estimation. Sometimes, the motion is due to the camera, for example a pan or a tilt in a dialogue scene. In such cases, this quantity may be misleading. However, the camera motion is generally slow and smooth for dialogue scenes. Since (2) exploits the similarities of consecutive frames, Mot_Z is very small for steady camera motion and its overall estimation is not affected much by motion outliers.

C. Color and Motion Similarities Between Shots

As explained in Section I, shots that belong to a scene often have similar visual (color) and/or action (motion) contents. Generally, dialogue shots span many frames and are filmed within a fixed physical setting. The repetitive transitions between the fixed camera views result in higher visual correlation of shots. In contrast, shots in fight and chase scenes change rapidly and last only a few frames [3]. In a similar fashion, the motion content of shots also depends on the nature of the scene. Dialogue shots are relatively calm (neither actors nor the camera exhibit large motion). In fight and chase shots, the camera motion is jerky and haphazard with larger movements of actors. For a given scene, these two attributes are kept consistent over time to maintain the pace of the movie. Thus, we compute the similarities between shots as a function of their visual and motion content features. That is, the similarity between shots i and j will be

$$ShotSim(i,j) = \alpha \cdot VisSim(i,j) + \beta \cdot MotSim(i,j) \tag{3}$$

where α and β are the weights given to each shot feature such that $\alpha + \beta = 1$. We used $\alpha = \beta = 0.5$ in our experiments which provided satisfactory results. The $VisSim$ between any arbitrary pair of shots i and j is the maximum $ColSim$ of all possible pairs of their key-frames, i.e.,

$$VisSim(i,j) = max_{p \in K_i, q \in K_j}\left(ColSim(p,q)\right) \tag{4}$$

where p and q are the key-frames of shot i and j, respectively. We compute the motion similarity, $MotSim$, between two shots as follows:

$$MotSim(i,j) = \frac{2 \times \min(Mot_i, Mot_j)}{Mot_i + Mot_j}. \tag{5}$$

Thus, if two shots have similar motion content, their $MotSim$ will have a higher value. Note that both $VisSim$ and $MotSim$ are in the range of 0–1.

IV. Scene Detection Using Graph Cuts

Graph partitioning techniques are known for their effective perceptual grouping. Several algorithms have been proposed for segmenting images based on pixel proximity and color intensity similarity, for example, [16], [22] and [15]. See [17] for a recent study of different graph partitioning techniques. In general, a good partition is one that maximizes the total dissimilarity between the partitions and maximizes the total similarity within the partitions. In this section, we first describe the construction of a graph in which each node represents a shot in the video and the edges are weighted by their similarity or affinity. Later, we demonstrate the use of the normalized cuts to detect scenes in the video.

A. Construction of Shot Similarity Graph, SSG

Given N shots, we construct a weighted undirected graph called a shot similarity graph, $G=(V,E)$, such that each shot i is represented by a node, v_i and an edge exists between all possible pairs of nodes. Let $e(i,j) \in E$ be the edge between nodes i and j with an associated weight $W(i,j)$ which reflects the likelihood that the two shots belong to one scene. It is less likely that two shots farther apart in time will belong to one scene. Therefore, the weight $W(i,j)$ is made proportional to the $ShotSim(i,j)$, as well as temporal proximity of the shots. This is formulated as

$$W(i,j) = w(i,j) \times ShotSim(i,j) \tag{6}$$

where $w(i,j)$ is a decreasing function of the temporal distance between the shots. We chose an exponential weighting function over a step function as in [24] for its relative simplicity and neutrality. It can be considered a memory parameter in that the ability to recall a shot gradually decreases with time. Thus the weight $w(i,j)$ decays with the temporal distance between the middle frames of the two shots under consideration, that is

$$w(i,j) = e^{-\frac{1}{d}\cdot\left|\frac{m_i-m_j}{\sigma}\right|^2} \tag{7}$$

where σ is the standard deviation of shot durations in the entire video. Fig. 3 shows w against the temporal distance between shots for different values of d. d influences the number of final clusters or scenes. A large value would result in higher similarity between shot pairs even if they are temporally far apart. With smaller values, shots will be *forgotten* quickly, thus forming numerous small clusters with fewer shots. One way of selecting d is to obtain a reasonable tradeoff between recall and precision values. A segmentation with a large number of scenes is likely to have higher recall and relatively poor precision and vice versa. Fig. 4 provides recall/precision curves with different values of d for two videos. We chose $d = 20$ which provided promising results. Fig. 5(a) shows the shot similarity graph constructed for 36 min of the movie "A Beautiful Mind". There are 219 shots in the video. The similarities between the nodes are represented by pixel intensities (lower intensity means higher similarity). The diagonal pixels represent the self similarity of shots. The magnified section of SSG shows that shots belonging to a scene appear in the form of clusters. Fig. 5(b) shows a similar graph for one episode of the sitcom "Seinfeld".

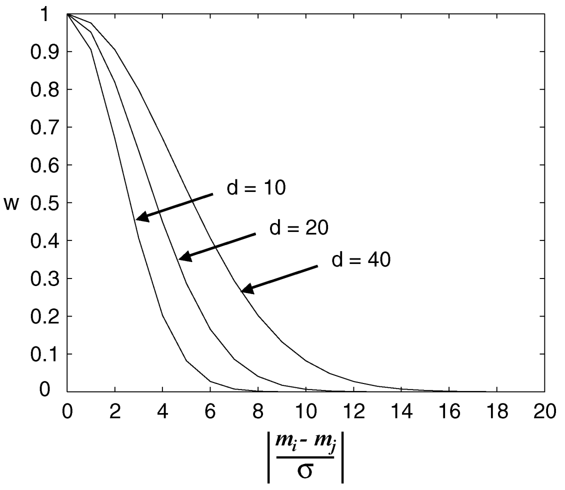

Fig. 3. Temporal proximity weight, $w(i,j)$, is an exponentially decreasing function of temporal distance between two arbitrary shots, i and j. Note that it decreases at a slower rate for larger values of d.

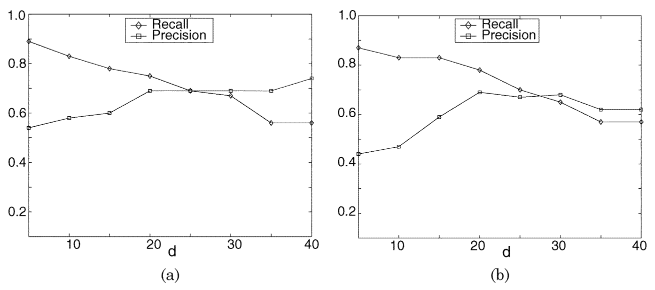

Fig. 4. Recall and precision with regard to the decay parameter d for (a) "Terminator II" and (b) "Top Gun".

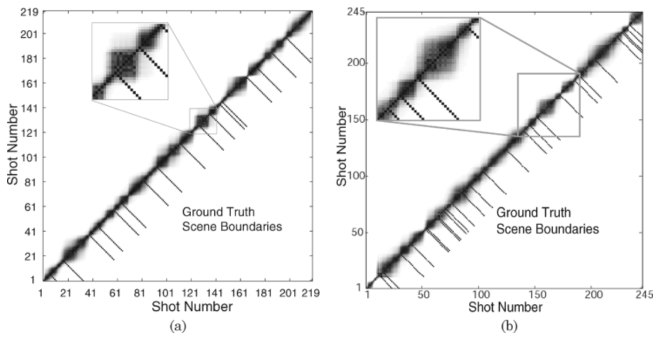

Fig. 5. Shot similarity graph for (a) 36 min of the movie "A Beautiful Mind" and (b) 18 min of the sitcom "Seinfeld". Higher similarity is represented by darker pixels. The ground truth scene boundaries are indicated with lines on the lower right side of the diagonal. Note that shots that belong to a particular scene form distinct clusters, as seen in the magnified section of the SSG.

B. Recursive Bipartitioning of SSG Using Normalized Cuts

We employ the graph partitioning technique proposed by Shi and Malik [16] called normalized cuts. Starting with an initial SSG, $G=(V,E)$, we seek to partition into two disjoint subgraphs, $G'=(V',E')$ and $G''=(V'',E'')$ such that $V' \cup V'' = V$ and $V' \cap V'' = \emptyset$. Such a partition is achieved by removing the edges connecting subgraphs G' and G''. In graph theory literature, the summation of weights associated with the edges being

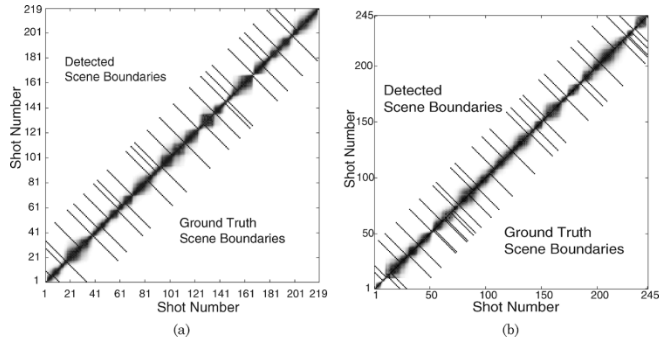

Fig. 6. Scene detection for (a) "A Beautiful Mind" and (b) "Seinfeld". Detected scene boundaries are indicated with lines in the upper-left side of the diagonal. Out of 18 scene boundaries, 15 scene boundaries are identified correctly in (a). In (b), 23 scene boundaries are correctly identified out of 28 scene boundaries in the ground truth.

removed is called a *cut* and it reflects the degree of dissimilarity between the two parts, that is

$$cut(V', V'') = \sum_{i \in V', j \in V''} W(i,j). \tag{8}$$

A partition that minimizes the *cut* value is considered an optimal bipartitioning of the graph. However, minimizing the *cut* value does not always provide a globally optimal solution as it favors cutting small sets of isolated nodes in the graph. To overcome this shortcoming, [16] also considered the *degree of association* of the partitions being created with regard to the parent graph. Thus, the association of the graph is computed as follows:

$$assoc(X, V) = \sum_{c \in X, d \in V} W(c,d) \tag{9}$$

where $assoc(X, V)$ is the association measure of the graph and equals the total connection from all nodes in X to all nodes in V. The new degree of disassociation is the cut cost as a fraction of the total edge connections to all the nodes in the graph and called normalized cuts or $Ncut$

$$Ncut(V', V'') = \frac{cut(V', V'')}{assoc(V', V)} + \frac{cut(V', V'')}{assoc(V'', V)}. \tag{10}$$

A partition is created for which Ncut is the minimum thus satisfying both criteria: minimization of the disassociation between the groups and maximization of the association within the groups. Fig. 6(a) shows the scene detection for the movie "A Beautiful Mind". Our experiments provided results with high recall/precision values even when applied to very different genres of feature movies.

V. Scene Representation

A scene representation using one or multiple images is crucial for building an interactive tool for video browsing. The purpose of this image is to give viewers a hint of the height of drama, suspense and/or action of the scene. Chapter selection menus on DVDs represent each chapter by one key-frame. The creators, who have complete access to the script of the movies, manually pick a frame that adequately reflects the scenario. We present a method to automate this task. Since this is a subjective process, the choice of image may vary from one individual to another. Although a lot of work has been done on detecting key-frames for a single shot, detection of a single key-frame for a scene with multiple shots has not yet been addressed.

We have observed that DVD chapter images have one, or often multiple faces. One reason for this is to introduce as many characters of the scene as possible. Rarely, an image of a building or landscape is preferred to highlight the location of the scene. Also, shots with less action are preferred. Thus, the criteria for a good representative shot can be summarized as:

- the shot is shown several times (higher visual similarity with other shots);
- the shot spans a longer period of time (longer shot length);
- the shot has minimal action content (smaller shot motion content);
- the shot has multiple people.

These criteria are intuitive but require further justification. Since computers lack the power of high-level human perception, we are left with only low-level image and video features. We need to numerically combine these features in a way that maximizes some desired measure for qualified shots. We call this measure *shot goodness* and compute it as a function of shot visual similarity, shot length and shot activity. For each scene, a fixed number of shots with the highest shot goodness value are selected as candidates. Of these, the shot with the maximum number of faces is selected as representative for the corresponding scene.

A. Measuring Shot Goodness

Shot goodness is computed by analyzing three properties of every shot; visual similarity, shot length and shot motion content. A quantity $C(i)$ is first computed for every shot i in the scene which is the summation of $VisSim$ [see (4)] with regard to every other shot in the scene, that is

$$C(i) = \sum_{j \in Scene} VisSim(i,j). \tag{11}$$

Note that $C(i)$ is large if the shot is shown several times in a scene. The shot goodness Γ is then computed as

$$\Gamma(i) = \frac{C^2(i) \times L_i}{log(Mot_i + \Theta)} \tag{12}$$

where L_i is the shot length of i^{th} shot and Θ is a small positive constant used to prevent division by zero. Note that (12) is derived intuitively from the criteria mentioned above. The squared value of $C(i)$ is used in accordance with the first criterion to give more emphasis to repetitive shots. Γ is also proportional to the length of shots which satisfies the second criterion. The shot motion content in the denominator satisfies the third criterion. The log term for shot motion content is incorporated to reduce its effect on shot goodness.

B. Detection of Faces

We used the face detector program provided by OpenCV library [1] which is based on the idea of object detection initially proposed by Viola *et al.* [19]. A classifier which is a cascade of boosted classifiers working with haar-like features, is trained with 24 × 24 pixel sample-views of human faces. We run it

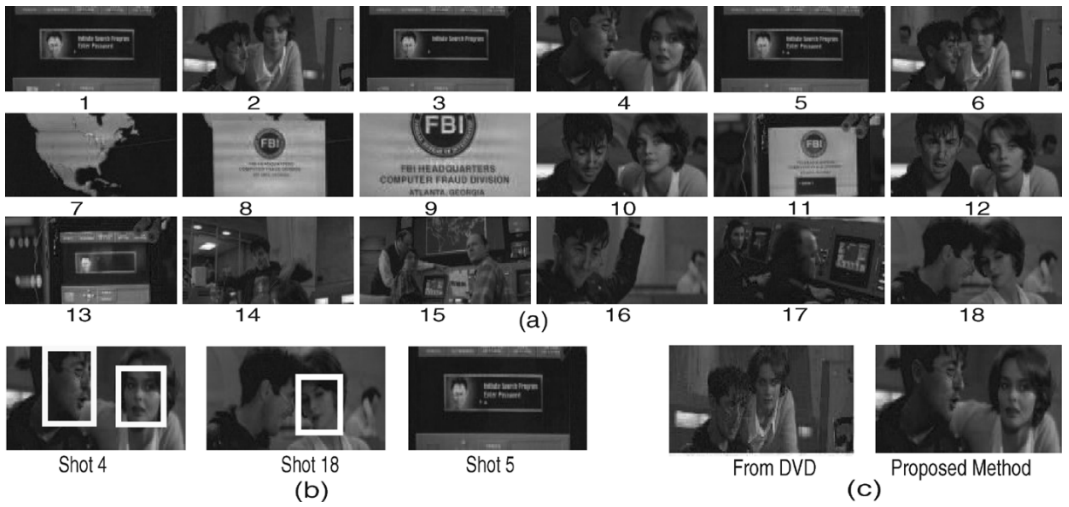

Fig. 7. Best key-frame detection in a scene from the movie "Golden Eye" with 18 shots. (a) First key-frame of each shot is shown with the shot number. (b) Three key-frames with the highest shot goodness values. Rectangles show the detected faces in the images. (c) Compare the key-frame selected by our algorithm (right) with the image obtained from the DVD (left).

with three scales to detect human faces. Although, the classifier performs better on frontal faces, it is robust to lighting and color variations. In the case of a tie or when no face is detected, the key-frame of the shot with the highest goodness value is selected. Fig. 7 shows key-frame detection for a scene from the movie "Golden Eye". More results are shown in Fig. 10.

VI. Experiments

We performed experiments on five Hollywood movies: "A Beautiful Mind(BM)", "Terminator II(T-II)", "Top Gun(TG)", "Gone in 60 seconds(G-60)", and "Golden Eye(GE)". Each sample was taken from the middle of the movie, 35–60 min long, and digitized at 29.97 fps. We also experimented with 18 min of a sitcom, "Seinfeld(SF)" (without commercials). The data set represent a variety of film genres such as action and drama as well as a TV sitcom which has a different shooting style from feature films. The experiments show that the algorithm is robust regardless of the film genre. For each video, two human observers identified the scene boundaries and the intersection of their segmentation is used as the ground truth. Chapter information from the DVDs is also incorporated to evaluate the results. DVD chapters appeared to be the supersets of scenes identified by human observers (see Table II). We evaluate the performance of the proposed method with regard to both sets of ground truth. No DVD ground truth was available for the sitcom "Seinfeld".

Table I(a) summarizes the data set and the results obtained by the proposed method with regard to the human-generated ground truth and also shows the number of false-positive and false-negative scenes. Generally, the *recall* and *precision* values provide a good assessment of any data-retrieval system. We define recall as the ratio of the number of correctly identified scene boundaries to the total number of scenes in the ground truth. Precision is the ratio of the number of correctly identified scenes to the number of scenes detected by the algorithm. To determine if a detected scene is correct, we use the *best match* method with a sliding window of τ seconds as the tolerance factor. Table I(b) provides the result statistics with regard to the ground truth obtained from the DVDs of each movie. While our final number of

TABLE I
SUMMARY OF DATA SET AND EXPERIMENTAL RESULTS WITH GROUND TRUTH GENERATED BY A HUMAN OBSERVER AND THE DVD CHAPTERS

	Title					
	BM	T-II	TG	G-60	GE	SF
Duration (min)	36	55	50	58	60	18
Shots	219	994	754	1332	879	245
RESULTS WITH THE HUMAN GENERATED GROUND TRUTH						
Scenes in ground truth	18	36	23	39	25	28
Scenes detected	28	39	26	57	44	27
Correct detection	15	27	18	29	22	23
False negative	3	9	5	10	3	5
False positive	13	12	8	28	22	4
Recall	83%	75%	78%	74%	88%	82%
Precision	54%	69%	69%	51%	50%	85%
RESULTS WITH THE DVD CHAPTERS						
Scenes in ground truth	7	36	13	17	19	N/A
Scenes detected	28	39	26	57	44	N/A
Correct detection	7	27	9	15	16	N/A
False negative	0	9	4	2	3	N/A
False positive	21	12	17	42	28	N/A
Recall	100%	75%	69%	88%	84%	N/A
Precision	25%	69%	35%	26%	36%	N/A

TABLE II
SCENE BOUNDARY DETECTION FOR THE MOVIE "A BEAUTIFUL MIND". CORRECTLY IDENTIFIED SCENES ARE MARKED WITH A CHECK✓

No.	DVD Chapter	Human Observer	Scene Det.?
1	Mathematicians	Mathematicians	✓
2		Reflections	✓
3		Princeton Dorm	✓
4		Drinking	✓
5	A Challenge	A Challenge	✓
6	The Need to Focus	The Need to Focus	✓
7		The Bar	×
8		Princeton	✓
9		Dorm Room	✓
10	Governing Dynamics	Governing Dynamics	✓
11		Research	×
12		With the Principle	✓
13		Celebrations	✓
14	The Pentagon	The Pentagon	✓
15		Wheelers Def. Labs	×
16	Teacher and Students	Teacher & Students	✓
17	Code Breaker	Code Breaker	✓
18		Laboratory	✓

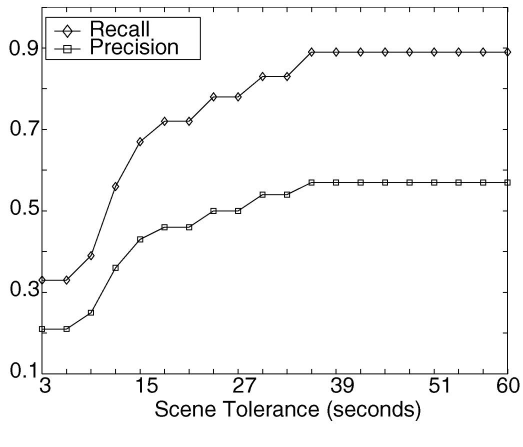

Fig. 8. Recall and precision of scene detection against the scene tolerance threshold for the movie "A Beautiful Mind".

scene boundaries is more than the ground truth, we believe that a slight oversegmentation is preferable over undersegmentation because split scenes can be combined by further analysis. While browsing a video, it is preferable to have two segments of one scene rather than one segment consisting of two scenes.

Fig. 8 provides scene detection accuracy against the temporal tolerance factor τ for the movie "A Beautiful Mind". In this movie, the mean and standard deviation of scene duration are $\mu = 120.71$ s and $\sigma = 45.44$ s. Note that recall and precision drastically change within a 3–18 s range and become steady after 36 s. In other words, detected scenes are likely to be 20–30 s from the real boundary. We chose $\tau = 30$ s in evaluating our system performance.

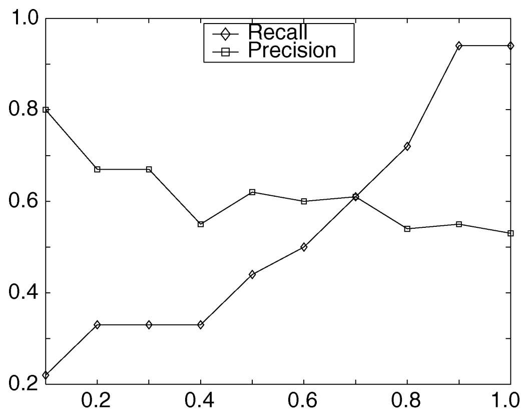

Fig. 9. Recall and precision with regard to the stopping threshold for 36 min of the movie "A Beautiful Mind". Our experiments show that good segmentation is achieved when λ is set between 0.6–0.9.

A. Computational Complexity

Minimizing normalized cut is an NP-complete problem. Shi and Malik [16] proposed the use of eigenvectors for both recursive partitioning and simultaneous K-way cut. Scenes, as defined in Section I, are composed of shots which are time continuous. That is, no two scenes have an overlap in physical time. We exploit this fact and reject all such partitions that do not satisfy the following constrain:

$$(i < j \text{ or } i > j) \ \ for\ all \ \ v_i \in V', \quad v_j \in V''.$$

Thus, a cut can only be made along the temporal axis (along the diagonal of Fig. 5). We apply a recursive algorithm of graph partitioning so that at each step the graph is divided into two subgraphs. The recursion is done as long as the Ncut value is below some stopping threshold, say λ. With a large λ, more scenes are obtained with a likelihood of large recall and small precision and vise versa (see Fig. 9). In the first recursion, there will be N computations to find the minimum Ncut value. Let there be M and $N - M$ shots present in each segment. This will require $M + (N - M) = N$ computations in the next recursion. Thus, the overall complexity will be $O(cN)$ where c is the number of scenes in the video. Generally, $c \ll N$.

B. Scene Representation Results

Fig. 10 compares our scene representation results using one key-frame with those in the DVD menu. We observed that the algorithm performs better for scenes with longer shots, such as a dialogue scene. We acknowledge the fact that the selection of one key-frame from a scene largely depends on discretion and personal choice, therefore, automatic key-frame detection is an ill-posed problem. A good example of this is in the movie "Terminator II". The frame chosen for the chapter "Meet John Conner" was a close shot of two actors, whereas our algorithm selected a frame with three actors. This may be preferable over the original one, as it provides more detail about the scene. Similarly, in the chapter "St. Petersburg, Russia" of "Golden Eye",

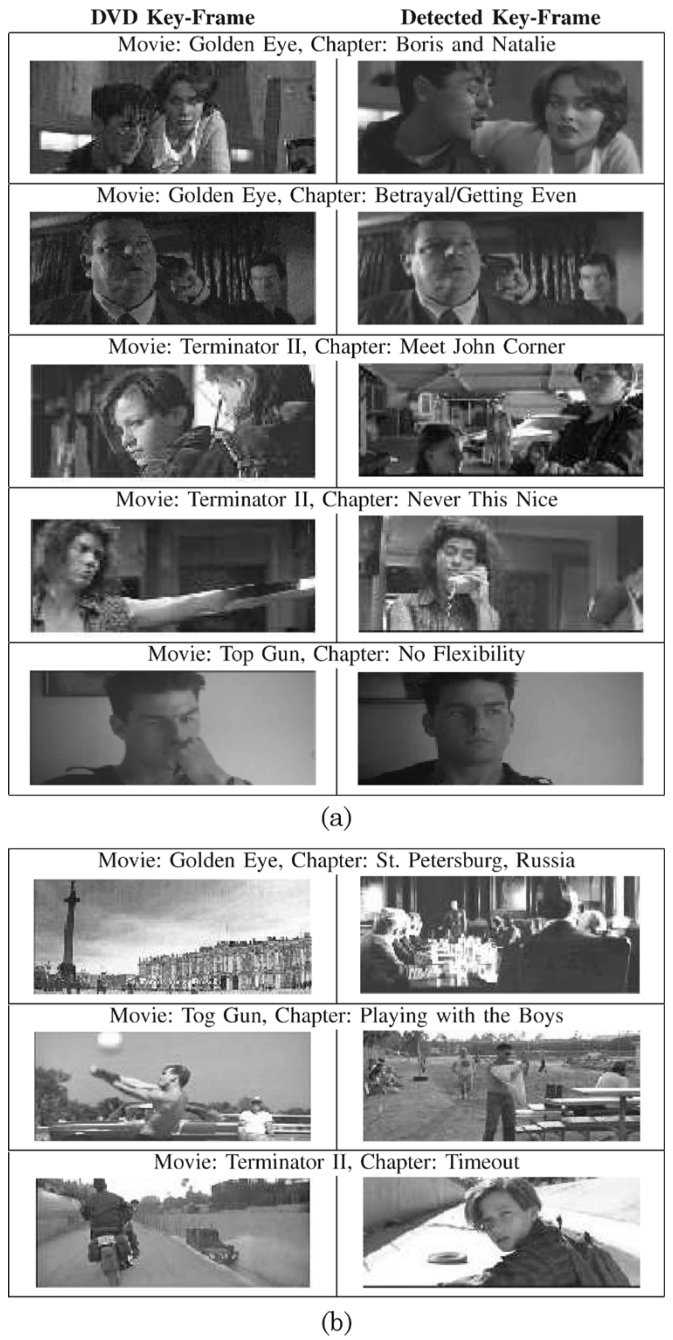
DVD Key-Frame
Detected Key-Frame
Movie: Golden Eye, Chapter: Boris and Natalie
Movie: Golden Eye, Chapter: Betrayal/Getting Even
Movie: Terminator II, Chapter: Meet John Corner
Movie: Terminator II, Chapter: Never This Nice
Movie: Top Gun, Chapter: No Flexibility
(a)
Movie: Golden Eye, Chapter: St. Petersburg, Russia
Movie: Tog Gun, Chapter: Playing with the Boys
Movie: Terminator II, Chapter: Timeout
(b)

Fig. 10. Scene content representation by scene key-frames. Left: frames obtained from DVDs. Right: automatically detected frames. (a) Detected key-frames which are very similar to DVD frames. (b) Detected key-frames which are not very similar to DVD frames.

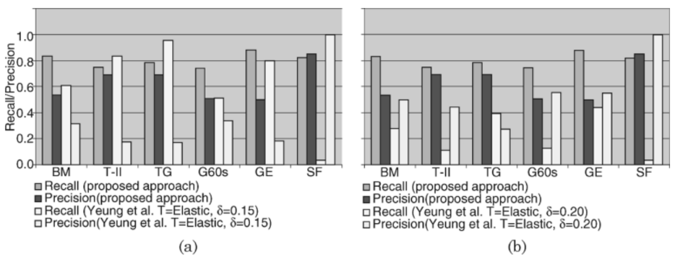

Fig. 11. Comparison of proposed algorithm with the baseline (Yeung *et al.*).

a picture of a building is chosen to reveal the location, whereas our algorithm detected a frame with people meeting in a room. This may be a better representation of the context of the scene.

C. Comparison With a Baseline Algorithm

In this section, we compare our algorithm with Yeung *et al.*, [24]. In their approach, shots are grouped into story units by constructing a *shot transition graph*. The segmentation is further refined by dynamically adjusting the temporal threshold T (called T *elastic*) as explained in [24, Sec. 7]. The experiments were performed with two different values of visual dissimilarity threshold, $\delta = 0.15$ and 0.20. We found that between these two values, the system gave the best results. Their output suffered from both over/under-segmentation problems. Fig. 11 shows a comparison of both approaches.

VII. Conclusion

We presented a novel method for high-level segmentation of video into scenes. We exploited the fact that shots that belong to one scene often have similar visual (color) and action (motion) attributes and transformed the scene segmentation task into a graph partitioning problem. The proposed method is superior to other graph-based approaches in that it considers all shots and captures the global similarities rather than merely the local similarities. Our approach is robust to local mismatches and produces semantically meaningful scenes. We also proposed a method to represent the scene content using one key-frame. We presented extensive experimental evaluation on several Hollywood movies and a TV sitcom, which validate the proposed approach. Thus, we provided a complete system to organize huge amounts of video without human intervention, which can offer browsing facilities to viewers in electronic form.

References

[1] Open Computer Vision Library.. [Online] Available: http://sourceforge.net/projects/opencvlibrary

[2] A. Aner and J. R. Kender, "Video summaries through mosaic-based shot and scene clustering," in *Proc. European Conf. Computer Vision*, 2002, pp. 388–402.

[3] D. Arijon, *Grammar of the Film Language*. New York: Hasting House, 1976.

[4] P. O. Gresele and T. S. Huang, "Gisting of video documents: a key frames selection algorithm using relative activity measure," in *Proc. Int. Conf. Visual Information Systems*, San Diego, CA, Dec. 1997.

[5] N. Haering, "A Framework for the Design of Event Detections," Ph.D. dissertation, Sch. Comput. Sci., Univ. Central Florida, Gainesville, 1999.

[6] N. Haering, R. J. Qian, and M. I. Sezan, "A semantic event-detection approach and its application to detecting hunts in wildlife video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 857–868, Sep. 2000.

[7] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 580–588, Jun. 1999.

[8] O. Javed, Z. Rasheed, and M. Shah, "A framework for segmentation of talk and game shows," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 532–537.

[9] I. Koprinska and S. Carrato, "Temporal video segmentation: a survey," *Signal Process.: Image Commun.*, vol. 16, no. 5, pp. 477–500, Jan. 2001.

[10] P. Lyman and H. R. Varian. (2000) How Much Information?. Sch. Inform. Manag. and Syst., Univ. California at Berkeley. [Online] Available: http://www.sims.berkeley.edu/research/projects/how-much-info/

[11] J. Meng, Y. Juan, and S.-F. Chang, Scene change detection in a MPEG compressed video sequence, in *SPIE Algorithms and Technologies*, pp. 14–25, 1995.

[12] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, "Motion-based video representation for scene change detection," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 127–142, Nov. 2002.

[13] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot, "Spectral structuring of home videos," in *Proc. Int. Conf. Image and Video Retrieval*, 2003, pp. 310–320.

[14] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *ACM Multimedia Syst. J., Special Issue Multimedia Systems on Video Libraries*, vol. 7, no. 5, pp. 359–368, Sep. 1999.
[15] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization: graph spectral partitioning and learning automata," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 504–525, May 2000.
[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[17] P. Soundararajan and S. S. Sarkar, "Analysis of MinCut, average cut and normalized cut measures," in *Workshop on Perceptual Organization in Computer Vision*, 2001.
[18] D. Swanberg, C.-F. Shu, and R. Jain, Knowledge-guided parsing in video databases, in *SPIE Storage and Retrieval for Image and Video Databases*, pp. 13–24, 1993.
[19] P. Viola and M. J. Jones, Rapid object detection using a boosted cascade of simple features, in *IEEE Comput. Vis. and Pattern Recognit.*, pp. 511–518, 2001.
[20] Webster. Webster Dictionary. [Online] Available: http://www.m-w.com
[21] W. Wolf, "Key frame selection by motion analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1996, pp. 1228–1231.
[22] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1101–1113, Nov. 1993.
[23] B.-L. Yeo and B. Liu, "Rapid scene change detection on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 6, pp. 533–544, Dec. 1995.
[24] M. Yeung, B.-L. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Comput. Vis. Image Understanding*, vol. 71, no. 1, pp. 94–109, Jul. 1998.
[25] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video ," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.
[26] J. Zhou and W. Tavanapong, "ShotWeave: a shot clustering technique for story browsing for large video databases," in *Proc. Int. Workshop Multimedia Data Document Engineering*, Mar. 2002, pp. 299–317.
[27] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. IEEE Int. Conf. Image Processing*, 1998, pp. 866–870.

Zeeshan Rasheed received the Bachelor's degree in electrical engineering from NED University of Engineering and Technology, Pakistan, in 1998 and the Ph.D. degree in computer science (computer vision focus) from the University of Central Florida, Orlando, in 2003.

He is currently a Video Scientist with ObjectVideo, Reston, VA. His research interests include video understanding and categorization, multimedia, human tracking, and real-time visual surveillance.

Dr. Rasheed was awarded the Hillman Fellowship in 2001 for excellence in research in the Computer Science Ph.D. program.

Mubarak Shah (F'03) is a Professor of computer science and the Founding Director of the Computer Visions Laboratory, University of Central Florida, Orlando. His research involves the areas of computer vision, video computing, and video surveillance and monitoring. He has supervised several Ph.D., M.S., and B.S. students to completion and has published close to 100 articles in leading journals and conferences. He is co-author *Motion-Based Recognition* (Norwell, MA: Kluwer, 1997) and *Video Registration* (Kluwer, 2003), and is an editor of the Kluwer international book series "Video Computing".

Dr. Shah was an IEEE Distinguished Visitor speaker during 1997–2000. He is an Associate Editor of the *Pattern Recognition and Machine Vision and Applications* journals. He was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (1998–2002), and a Guest Editor of the special issue of the *International Journal of Computer Vision on Video Computing*. He received the Harris Corporation Engineering Achievement Award in 1999, the Research Incentive Award in 2003, and the IEEE Outstanding Engineering Educator Award in 1997.