# Video Scene Segmentation Using Markov Chain Monte Carlo

Yun Zhai, *Student Member, IEEE*, and Mubarak Shah, *Fellow, IEEE*

***Abstract*—Videos are composed of many shots that are caused by different camera operations, e.g., on/off operations and switching between cameras. One important goal in video analysis is to group the shots into temporal scenes, such that all the shots in a single scene are related to the same subject, which could be a particular physical setting, an ongoing action or a theme. In this paper, we present a general framework for temporal scene segmentation in various video domains. The proposed method is formulated in a statistical fashion and uses the Markov chain Monte Carlo (MCMC) technique to determine the boundaries between video scenes. In this approach, a set of arbitrary scene boundaries are initialized at random locations and are automatically updated using two types of updates: diffusion and jumps. Diffusion is the process of updating the boundaries between adjacent scenes. Jumps consist of two reversible operations: the merging of two scenes and the splitting of an existing scene. The posterior probability of the target distribution of the number of scenes and their corresponding boundary locations is computed based on the model priors and the data likelihood. The updates of the model parameters are controlled by the hypothesis ratio test in the MCMC process, and the samples are collected to generate the final scene boundaries. The major advantage of the proposed framework is two-fold: 1) it is able to find the weak boundaries as well as the strong boundaries, i.e., it does not rely on the fixed threshold; 2) it can be applied to different video domains. We have tested the proposed method on two video domains: home videos and feature films, and accurate results have been obtained.**

***Index Terms*—Markov chain Monte Carlo, video scene segmentation.**

## I. Introduction

VIDEOS are often constructed in the hierarchical fashion: $[\text{Frame}] \rightarrow [\text{Shot}] \rightarrow [\text{Scene}] \rightarrow [\text{Video}]$. The lowest level contains the individual frames. A series of continuous frames with consistent background settings constitute a shot. A scene or a story is a group of semantically related shots, which are coherent to a certain subject or theme. At the highest level, the entire video is composed of multiple scenes, which result in the complete storyline. Scenes are the semantic units of the video, and temporal scene segmentation is defined as a process of clustering video shots into temporal groups, such that the shots within each group are related to each other with respect to certain aspects. This is an important and fundamental problem in video processing and understanding, and it provides more meaningful and complete information for the video content understanding compared to the shot-level analysis. Scene segmentation has many applications in various domains. For example, in the feature films, scene segmentation provides the chapters that correspond to the different subthemes of the movies. In television videos, segmentation can be used to separate the commercials from the regular programs. In news broadcast programs, segmentation can be used to identify different news stories. In home videos, scene segmentation may help the consumers to logically organize the videos related to the different events [(e.g., birthday, graduation, weddings, or vacation (e.g. city tours, sightseeing)].

Scenes are composed of the video shots. The video shots are caused by different camera operations, e.g., turning the camera on/off, the switching between cameras, and other video editing techniques. Consider this: a tourist is recording a video around a monument. He wants to have different views of the monument. Therefore, he takes one sequence from the frontal view and shuts the camera off. Then, he walks to the side of the monument and records another sequence. In this case, the entire scene is composed of two shots, which are generated by the operations (on/off) of a single camera. On the other hand, in movies or TV programs, the shots are generated from different cameras and are appended one after another to constitute the story lines. A scene sometimes can be composed of a single shot. For instance, in the example described above, the tourist could have the camera on all the time and keeps recording the video. In this case, the scene and the shot are the same. However, more often, scenes are composed of multiple shots, such as movies or TV programs. Hence, a single shot is insufficient to reveal the semantic meaning of the video content. For example, in feature films, how could one answer a query related to a suspense scene based only on the content of a single shot? These types of scenes can only be identified with multiple shots showing the increasing tension in the video. In other domains, more often the semantic concepts are difficult to be determined by using only a single shot, since they are introduced to viewers over time. Thus, a meaningful result can only be achieved by exploiting the video scenes, which are the interconnections of the shot contents.

### A. Related Work

Several temporal segmentation methods have been developed for different types of videos. Hanjalic *et al.* [11] proposed a method for detecting boundaries of the logical story units in movies. In their work, inter-shot similarity is computed based on the block matching of the keyframes. Similar shots are linked, and the segmentation process is performed by connecting the overlapping links. Rasheed *et al.* [23] proposed a two-pass algorithm for scene segmentation in feature films and TV shows.

Manuscript received January 13, 2005; revised September 20, 2005. This work was supported in part by the U.S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government. The associate editor coordinating the review of this manuscript and apporving it for publication was Dr. Lap-Pui Chau.

The authors are with the School of Computer Science, University of Central Florida, Orlando, FL 32826 USA (e-mail: yzhai@cs.ucf.edu; shah@cs.ucf.edu).

Digital Object Identifier 10.1109/TMM.2006.876299

In the first pass, potential scene boundaries of the video are initially detected based on the color similarity constraint, *backward shot coherence* (BSC). Oversegmented scenes from the first pass are then merged in the second pass, based on the analysis of the motion content in the scenes. Sundaram *et al.* [25] used the audiovisual features of the video in the movie scene segmentation. First, two types of scenes, audio scenes and video scenes, are detected separately. Then, the correspondences between these two sets of scenes are determined using a time-constrained nearest-neighbor algorithm. Adams *et al.* [1] proposed the "tempo" for the segmentation of the movies. The "tempo" of a shot is a combination of the shot length and the motion content of shot. The dramatic story sections or events in the movie are detected by finding the zero-crossings of the "tempo" plot. Yeung *et al.* [28] proposed a graph-based representation of the video data by constructing a shot connectivity graph. The graph is split into several subportions using the complete-link method of hierarchical clustering such that each subgraph satisfies a color similarity constraint. These methods are based on the "film grammar," which is a set of production rules of how the movies or TV shows should be composed. For instance, in action scenes, the shots are generally short, and their motion content is high. On the other hand, the shots are long and the visual appearance is smooth in drama scenes. However, these heuristics are not applicable to the other types of videos. For instance, home videos are recorded in a completely "free" style. Shooters are not trained with recording techniques, and often no obvious format or pattern exists in the video. Furthermore, since the rules in the production of films and TV shows are different, the methods for these two domains of videos cannot be used interchangeably.

There is a particular interest in the story segmentation of the news broadcast videos. Hoashi *et al.* [13] has proposed an SVM-based news segmentation method. The segmentation process involves the detection of the general story boundaries, in addition to the special type of stories, e.g., finance report and sports news. Finally, the anchor shots are further analyzed based on the audio silence. Hsu *et al.* [14] proposed a statistical approach based on the discriminative models. The authors have developed the *BoostME*, which uses the maximum entropy classifiers and the associated confidence scores in each boosting iteration. Chaisorn *et al.* [4] used hidden Markov models (HMM) to find the story boundaries. The video shots are first classified into different categories. The HMM contains four states and is trained on three features: type of the shot, whether the location changes (true or false), and whether the speaker changes (true or fase). These methods were developed based on the unique characteristics of the news video. The video shots are commonly classified into news program related categories, e.g., anchor person, weather, commercials and lead-in/out shots. These categories are not available in other domains of videos, such as home videos or feature films. Furthermore, the news segmentation methods usually involve the special treatment on the anchor person shots, which exist only in news videos.

### B. Proposed Approach

In this paper, we propose a general framework for the temporal video segmentation by using the Markov chain Monte Carlo (MCMC) technique. Many of the previously developed methods are based on the fixed global thresholds, which are not desirable in many cases. Moreover, due to the fixed thresholds, these methods are likely to generate either oversegmentation or undersegmentation. Also, these methods may use some special knowledge about a particular domain, which may not be appropriate for other domains. For example, there is no obvious video structure in home videos. Due to that, it is not easy to generalize these methods to other domains. In contrast, we do not use any fixed threshold or utilize any structure information of the video. Instead, we have developed an iterative method to evaluate the segmentation parameters, including the number of the scene segments and their corresponding locations. In our formulation, if the number of the segments changes, the dimension of the vector containing the boundary locations also changes. The solution space for these two parameters is too complex for the direct analytical computation. Therefore, these two parameters are estimated in a statistical fashion using the MCMC technique.

The MCMC technique has been used in several applications in the fields of image processing, video content analysis and computer vision in the past few years. Geman *et al.* [6] were the first ones to apply the MCMC technique in the image analysis using the Gibbs sampler. The MCMC technique involving the jump and diffusion method was introduced by Grenander *et al.* [8], and Green [7] further proposed the reversible jumps. It has been applied in sampling and learning by Zhu *et al.* [33]. For one-dimensional (1-D) signal segmentation problems, Phillips *et al.* has discussed the changepoint problem in [22]. Dellaert *et al.* [5] proposed an EM-based technique for solving the structure-from-motion (SFM) problem without known correspondences. The MCMC algorithm [12] with symmetric transition probabilities was used to generate the samples of the assignment vectors for the feature points in each frame. Senegas [24] proposed a method for solving the disparity problem in stereo vision. The MCMC sampling process was applied to estimate the posterior distribution of the disparity. Tu *et al.* [27] and Han *et al.* [10] have applied the data-driven MCMC (DDMCMC) to the optical and range image segmentations.

Our proposed Markov chain contains three types of updates: shifting of boundaries, merging of two adjacent scenes and the splitting of one scene into two scenes. Due to these updates, the solution can jump between different parameters spaces, (the dimension of the parameter vector can change), as well as diffuse inside the same space, (the elements in the parameter vector are changed without changing the vector dimension). We assume that each shot in the video has a likelihood of being declared as the scene boundary. Shots with higher likelihoods coincide more with the true boundaries. Initially, two segments are assumed, and they are separated by a randomly selected shot. Then, in each iteration of the updates in the MCMC process, several shots are declared as the scene boundaries. Their likelihoods are accumulated, while the likelihoods of other shots are kept the same. Several Markov chains are executed independently to avoid the possible misdetections caused by a single chain, and the samples from all of the chains are collected for the computation of the shot likelihoods. Finally, the shots with the highest likelihoods in their neighborhoods are declared as the scene boundary locations. One advantage of using the sampling technique is that both the weak and strong boundaries can be detected without defining any specific threshold. We have tested the proposed framework on two video domains, home videos and feature films, and very accurate and competitive results have been obtained.

The rest of this paper is organized as follows. Section II proposes the MCMC algorithm and presents the computations of the transition probabilities and the posterior probability. Sections III-A and III-B deal with the applications of the general framework on the segmentations of the home videos and the feature films, respectively. Section IV presents the discussions of the proposed work on other video domains. Finally, Section V provides the conclusion and discussions of the proposed framework.

## II. Proposed Framework

By the problem definition, given the shots in the video, scene segmentation of the video is a process of grouping the related shots into clusters. In each scene, the shots are related to each other in terms of the corresponding *central concept*. The *central concepts* are different in various contexts. For instance, in home videos, the *central concept* sometimes refers to the same physical environmental setting, e.g., shots related to the same historical monument, or sometimes it refers to the same event, e.g., shots related to a birthday party or a wedding ceremony. In the news programs, the *central concept* refers to a specific story topic, e.g., shots related to a political reporting, a weather forecast or a sports reporting. In the feature films, *central concept* refers to the same subthemes of the story line, e.g., shots related to an action scene or a suspense scene. Different scenes are distinguished by their differences with respect to the *central concept*, and the scene boundaries are the locations where the intrinsic properties of the *central concept* change.

Based on this, we propose a statistical solution for the two model parameters, the number of the scenes and their corresponding boundary locations. The boundary locations are considered as the changepoints of the *central concept*, and the problem is formulated as a changepoint problem. In a typical changepoint problem, the random process has different controlling parameters over time. The goal is to find the points where these parameters change. A simple example of a changepoint problem is shown in Fig. 1. In this example, 600 observations are generated from five different uniform distributions. The changepoints are the locations where the distribution mean changes (the steps in the plot). In our application of the temporal scene segmentation, the controlling parameters become the *central concept*, and the steps in the posterior mean plot become the scene boundaries in the video. To estimate the boundaries locations, the MCMC technique is used. In the iterative process of MCMC, the posterior probability of the model parameters is computed based on the model priors and the data likelihood of the video. The samples are collected based on the ratio tests involving the posterior probabilities and the transition probabilities. In the rest of this section, we first introduce the overall MCMC algorithm. Then, a detailed description of the different types of update proposals is presented. Finally, we describe the computation of the posterior probability.

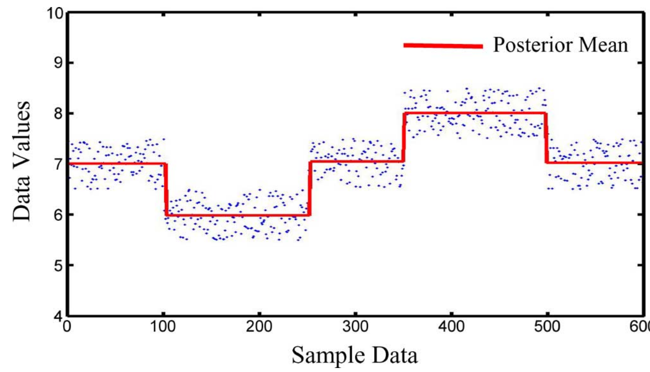


Fig. 1. Example of the changepoint problem. There are five segments containing over the 600 observations that are generated by the uniform distributions with different parameters. The red plot is the posterior mean of the segments, and the locations of the steps are the changepoints in the data, i.e., the places where the mean changes. (Color version available online at http://ieeexplore.ieee.org.)

### A. General MCMC Algorithm

We use a hierarchical Bayesian model in the MCMC process. We assume that the model set $\{M_k, k \in \Phi\}$ is a countable set, where $k$ is the number of the detected scenes, and $\Phi = \{1, 2, \cdots\}$ is a set of all the possible scene numbers. Model $M_k$ has a parameter vector $\theta_k$, which contains the $k-1$ scene boundary locations (note: since the first scene always takes the first shot as its starting boundary, it is ignored in our estimation process). Let $y$ denote the video features selected for the data likelihood computation. Based on the Bayes rule, the posterior probability of the parameter $k$ and $\theta_k$ given $y$ is

$$p(k, \theta_k|y) \propto p(y|k, \theta_k)p(\theta_k|k)p(k) \quad (1)$$

where $p(k)$ is the prior probability for the number of scenes, $p(\theta_k|k)$ is the conditional prior for the boundary locations $\theta_k$ given $k$, and $p(y|k, \theta_k)$ is the likelihood of the data given the parameters $k$ and $\theta_k$. Since the boundary vector, $\theta_k$, implicitly determines $k$, the above equation can be further simplified as

$$p(k, \theta_k|y) \propto p(y|\theta_k)p(\theta_k|k)p(k). \quad (2)$$

In the rest of this paper, we use the shorter term $\pi(x) = p(k, \theta_k|y)$ to denote this target posterior, with $x = \{k, \theta_k\}$ considered as a combined parameter vector of $k$ and $\theta_k$.

The general Metropolis-Hasting-Green algorithm [7] is well suited for our task, where the dimension of the parameter vector, $x$, may change during the updates. It is described as follows.

- Initialize the model parameters $x_0$.
- At each iteration $i$, perform the following actions.
  1) Generate $Th_\alpha$ from $Uni[0, 1]$.
  2) Create a new parameter $x'_{i-1}$ from some trial distribution based only on $x_{i-1}$ with a proposal transition (diffusion or jump).
  3) Calculate the ratio $\alpha(x_{i-1}, x'_{i-1})$ as

$$\alpha\left(x_{i-1}, x'_{i-1}\right) = min\left\{1, \frac{\pi\left(x'_{i-1}\right) q\left(x'_{i-1}, x_{i-1}\right)}{\pi(x_{i-1}) q\left(x_{i-1}, x'_{i-1}\right)}\right\}. \quad (3)$$

  4) Update $x_i = x'_{i-1}$, if $\alpha > Th_\alpha$. Otherwise, set $x_i = x_{i-1}$.

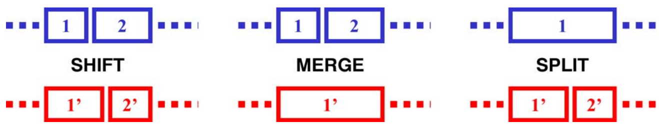


Fig. 2. Graphical representation of three types of the updates. The top row shows the scenes before updates, and the bottom row shows the update results. (Color version available online at http://ieeexplore.ieee.org.)

In this algorithm, $q(x, x')$ is the transition probability from $x$ to $x'$. The transition probability from one state to another depends on the type of the updates. It should satisfy the reversibility property. Therefore, the proposed updates should also be reversible to ensure this property.

Before going into the detailed description of the updating process, we first present the notations for the variables. Let $k$ be the current number of detected scenes, $T$ be the total number of shots in the video, $S_m$ be the $m$-th scene with shots $\{s_m^1, s_m^2, \cdots, s_m^{n_m}\}$, where $n_m$ is the number of shots in scene $S_m$, $S'_m$ be the $m$-th scene after update, $\mathbb{L}(y|\theta_k)$ be the data likelihood of the entire video, $\mathbb{L}(y_m|f_m)$ be the likelihood of scene $S_m$ given the corresponding features $f_m$. Finally, $k_{max}$ is the maximum number of the scenes allowed.

The proposal updates contain two parts, diffusion and jumps. Diffusion is defined as the update without changing the structure of the parameter vector $x$. It traverses within the same subspace. On the other hand, jumps do change the structure and traverse across different subspaces. In our case, the diffusion is the shifting of the boundaries between the adjacent scenes. There are two types of jumps: the merging of two adjacent scenes and the splitting of an existing scene. Fig. 2 shows the graphical representations of the updates. In many applications ([7], [10], [27]), two more updates were proposed: diffusion on the segment model parameter(s) and the change of the segment models. The segment model parameters are the ones that control the generation of the sample data, e.g., posterior means in Fig. 1. In our application of the video scene segmentation, based on the underlying assumption that each segment is coherent to its *central concept*, there is often only one scene model for a single video domain. Thus, changing between models is not needed in this case. Furthermore, in some cases like home videos, the data size (number of shots in our case) is small. The maximum likelihood estimator is adequately effective to compute the parameter(s). Therefore, the model parameter diffusion steps can also be dropped.

Let $\eta_k$, $b_k$ and $d_k$ denote the probabilities of choosing shifting, merging and splitting, respectively. They satisfy $\eta_k + b_k + d_k = 1$. Naturally, $\eta_1 = b_1 = 0$ and $d_{k_{max}} = 0$. We use the similar computation proposed in [7], where $b_{k+1} = c \cdot min\{1, p(k)/p(k+1)\}$ and $d_k = c \cdot min\{1, p(k+1)/p(k)\}$, with constant $c$ such that $b_k + d_k \leq C, \forall k = 1, \cdots, k_{max}$. This results in $b_{k+1}p(k+1) = d_k p(k)$.

### B. Stochastic Diffusions

The diffusions involve the shifts of the scene boundaries between adjacent video scenes. The update is carried out as follows.

- A number $m$ is randomly drawn from the discrete uniform distribution $[1, k-1]$, such that the boundary between $S_m$ and $S_{m+1}$ is updated.
- The new boundary $s^t$ is drawn from a 1-D normal distribution with the mean at the original boundary $s_{m+1}^1$ in the range of $[s_m^1, s_{m+1}^{n_{m+1}}]$. The updated scene $S'_m$ contains shots of $\{s_m^1, \cdots, s^{t-1}\}$, and the updated scene $S'_{m+1}$ contains $\{s^t, \cdots, s_{m+1}^{n_{m+1}}\}$.

Assume the number of the current scenes is $k$ and the current parameter vector is $x = \{k, \theta_k\}$. Then, the probability for selecting scene $S_m$ is $1/(k-1)$. Since the potential shift is drawn from a normal distribution around the original scene boundary $\hat{t}$, this drawing probability for the new boundary $t$ is computed as

$$p(t) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{\Delta t^2}{2\sigma^2}\right)\left(I_{[s_m^1, s_{m+1}^{n_{m+1}}]}(t)\right) \qquad (4)$$

where $\Delta t = t - \hat{t}$, and $\sigma$ is the standard deviation of the movement (in our experiment, $\sigma = 2$). The indicator function $I(t)$ controls the shift, such that the new boundary is within the correct range. The normal distribution is assumed since the new boundary is not expected to deviate from the old boundary too far. In summary, the forward transition probability for the shift update is $q(x, x') = (1/(k-1))p(t)$.

During this entire update, the total number of scenes, $k$, is not changed, and the new boundary remains in the original range $[s_m^1, s_{m+1}^{n_{m+1}}]$. The reverse transition is the process of shifting from the new boundary $t$ back to the original boundary $\hat{t}$. Thus, the relationship between $q(x, x')$ and its reverse version $q(x', x)$ is equal due to the symmetrical property of the normal distribution.

### C. Reversible Jumps: Merge and Split

For the jump updates, the transition during a merge is related to the transition of a split, since merge and split are a pair of reversed updates. Let us consider the splits first. The number of scenes is increased by 1 by splitting a scene $S_m = \{s_m^1, \cdots, s_m^{n_m}\}$ into two new scenes $S'_m = \{s_m^1, \cdots, t-1\}$ and $S'_{m+1} = \{t, \cdots, s_m^{n_m}\}$, where $t$ is the new boundary. The process contains two portions: selecting a scene $S_m$ and selecting a new boundary between its old boundaries. The selection of the new boundary in the split process can be performed assuming the uniform distributions [7]. However, to achieve better performance, the data-driven technique is often used ([10] and [27]) to propose the jump transitions. We assume uniform probability for selecting scene $S_m$. The new boundary $t$ is chosen, such that it provides the maximum likelihoods for the two new scenes:

$$t = arg\ max\left(\mathbb{L}\left(S'_m|f'_m\right) + \mathbb{L}\left(S'_{m+1}|f'_{m+1}\right)\right) \qquad (5)$$

where $\mathbb{L}(S'_m|f'_m)$ and $\mathbb{L}(S'_{m+1}|f'_{m+1})$ are the likelihoods of the new scenes $S'_m$ and $S'_{m+1}$ given their corresponding features. If we consider the video scenes are independent events in the time series, the proposal probability for a split can be expressed in the following form:

$$q(x, x') = \frac{1}{k}\mathbb{L}\left(S'_m|f'_m\right)\mathbb{L}\left(S'_{m+1}|f'_{m+1}\right). \qquad (6)$$

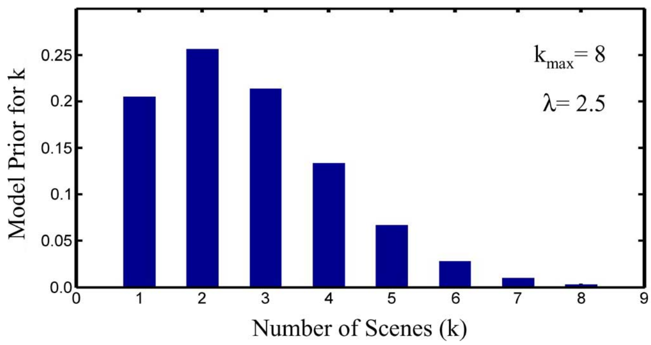


Fig. 3. Prior distribution (Poisson) of the model parameter $k$, the number of scenes in the video. The mean of the distribution, $\lambda$, is preassigned as 2.5, and $k_{max}$ is 8. (Color version available online at http://ieeexplore.ieee.org.)

The reversed update of the split is the merging of two scenes into one. The construction of the proposal probability for the merge can be carried out similarly to the one for the split. Again, we assume the uniform distribution for selecting scene $S_m$, such that scenes $S_m$ and $S_{m+1}$ are merged into $S'_m$. The proposal probability for the merge transition is constructed as follows:

$$q(x, x') = \frac{1}{k-1} \mathbb{L}\left(S'_m | f'_m\right). \tag{7}$$

## D. Posterior Probability

Since Poisson distribution models the number of incidents happening in a unit time interval, we assume the number of scenes, $k$, is drawn from a such distribution with mean $\lambda$. The model prior on $k$ is computed as

$$p(k) = e^{-\lambda} \frac{\lambda^k}{k!} \cdot I_{[1,k_{max}]}(k) \tag{8}$$

where $I_{[1,k_{max}]}(k)$ is an indicator function. $I_k = 1$, if $1 \leq k \leq k_{max}$; $I_k = 0$ otherwise. A plot of the prior distribution is shown in Fig. 3.

If there are $k$ segments (scenes) in the video, then there are $k-1$ scene boundaries, since the boundary for the first scene is always the beginning of the video. The probability of $p(\theta_k|k)$ is the same as the probability of selecting a subset with size $k-1$ from the remaining $T-1$ shots. Therefore, the conditional prior can be defined in terms of the combinations

$$p(\theta_k | k) = \frac{1}{C_{k-1}^{T-1}} = \frac{(k-1)!(T-k)!}{(T-1)!}. \tag{9}$$

The last term to be computed is the likelihood. Let $\mathbb{L}(y|\theta_k) = p(y|\theta_k)$ denote the global likelihood of the video data $y$ given the parameter vector $\theta_k$. As discussed in Section II, each scene possesses a different *central concept*. It is meaningful to make an assumption that each scene is independently recorded from others. Therefore, the overall likelihood can be expressed as

$$\mathbb{L}(y|\theta_k) = \left( \prod_{m=1}^{L} \mathbb{L}(y_m | f_m) \right)^{\frac{1}{L}} \tag{10}$$

where $\mathbb{L}(y_m|f_m)$ is the individual likelihood of data $y_m$ in scene $S_m$, based on the feature values $f_m$. The geometric mean of the individual likelihoods is considered for the normalization purpose. In order to make the ratio test meaningful, the likelihood should be scaled to the same level during each iteration. The definition of the *central concept* is different across domains. Therefore, the features selected to compute the likelihoods are different for the different types of videos. Here, $\mathbb{L}(y|\theta_k)$ is a general representation of the likelihood rather than a specific computation.

The target posterior probability is proportional to the product of the model prior $p(k)$, the conditional prior $p(\theta_k|k)$, and the data likelihood $\mathbb{L}(y|\theta_k)$,

$$\pi(x) \propto \mathbb{L}(y|\theta_k) p(\theta_k|k) p(k). \tag{11}$$

To determine if the proposed update in the parameter space is accepted or rejected, we compute the ratio of the two terms: $\pi(x')q(x', x)$ and $\pi(x)q(x, x')$. If the ratio, $\alpha(x, x')$, satisfies the stochastically generated threshold, the proposed update is accepted; otherwise, the model parameters are kept the same as in the previous iteration.

# III. Applications and Discussions

In this section, we demonstrate the proposed scene segmentation method on two video domains. If we examine the generation process of the videos, we can classify them into two categories.

- *Produced Videos*: This group contains the feature films, television news programs and other television talk or game shows. They are initially recorded in the raw format and are later modified to produce the carefully organized video programs with accordance to the certain video production rules.
- *Raw Videos*: Compared to the previous group, this category involves little post-modification and contains videos that are mostly in the forms in which they were originally recorded.

Common domains in this category are home, surveillance and meeting videos.

Due to the large variety of video domains, we have selected two representative domains to demonstrate the effectiveness and the generality of the proposed method, with one domain from each of the categories described above. The home video domain is chosen as the representative domain of the *Raw Video* category, and the feature film domain is selected for the *Produced Videos* category. In this paper, we assume the video shots are available. In the experiment, we used a multi-resolution method provided in [30] to detect and classify the video shot boundaries in both home videos and feature films.

## A. Home Videos

Home video is a broad term that refers to the videos composed with a "free-style," e.g., family videos, tour videos, wedding tapes or ground reconnaissance videos (GRV). They are recorded from hand-held cameras, spy cameras, cameras mounted on ground vehicles, etc., and come in different forms.

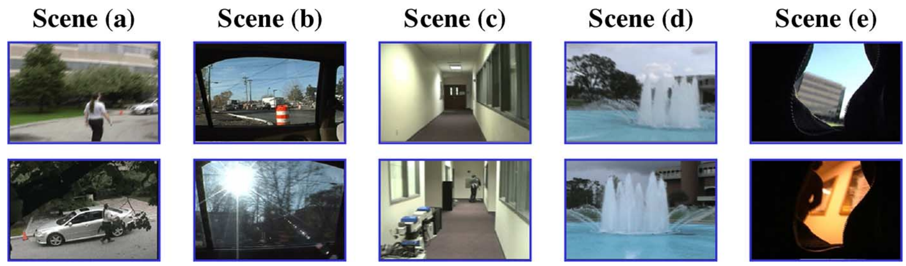


Fig. 4. Five example home video scenes with their keyframes. Some of them are (c) the indoor scene; some are (a), (b), (d), (e) the outdoor scenes. Scenes (a), (b) were taken by the cameras mounted on the ground vehicles, (e) was taken by a spy camera in a bag, and (c), (d) were taken by hand-held cameras. (Color version available online at http://ieeexplore.ieee.org.)

Some are with high resolutions, while some others have low quality. Some have full field of view, and some may be recorded by cameras hidden in the bags (GRV), so part of their field of view is blocked by the carrier. Some example keyframes are shown in Fig. 4. Temporal scene segmentation of home videos provides the logical units related to the interesting locations or events, and the output segments can be used for the further analysis and processing of the videos, e.g., indexing, storage, retrieval of the video and action recognition. Since there is no grammar involved in the production process of the home videos, the temporal segmentation emphasizes more on the analysis of the features derived from the video than on the video structure. As mentioned in Section I, this type of analysis could be threshold-based, zero-crossing based, etc., with or without the training of the features. Home videos are not well-controlled as other domains like television programs. The scene boundaries sometimes are clearly identifiable (strong boundaries), but many times they are difficult to be determined using the same criteria for the strong boundary detection. Due to this uncertainty in the home videos, it is likely to create either undersegmentation or oversegmentation using any fixed threshold, and it is not practical to train the system for the threshold selection. On the other hand, the proposed approach finds the boundary locations by detecting the local peaks in the likelihood plot of the video shots, and therefore, avoids the previously mentioned problems.

*1) Feature Selection:* In the context of temporal scene segmentation, a variety of features have been exploited. The commonly used features include color, motion content, shot length, etc. Since the home videos are taken in a "free style," the patterns for the motion content and the shot length are not distinctive across different scenes. Usually the shots in the same temporal scene are coherent with respect to the same environment. There are visual similarities that exist among these shots. On the other hand, the shots from different scenes should be visually distinctive. Therefore, we have focused our efforts on the analysis of the color information in the shots. We use the histograms to represent the color information of the video frames. The color histogram for each frame is the three-dimensional histogram in the RGB space with eight bins in each dimension. Let $h_i$ be the histogram for frame $f_i$. Furthermore, we define the histogram intersection between frames $f_i$ and $f_j$ as

$$HistInter(f_i, f_j) = \sum_{b \in Allbins} min\left(h_i^b,\ h_j^b\right) \qquad (12)$$

where $b$ is the individual bin in the histogram.

Instead of using all the frames in the shot, we extract the keyframes as the representation of the shot, and further analysis is performed based on the keyframes only. It is common to select a single keyframe for each shot. However, for the shots with long durations and with high activity content, multiple keyframes form better representation. Several keyframe selection approaches have been proposed in the past few years ([9], [11], [23], [32]). In this paper, we use the method proposed in [23]. Assume there are a total of $n$ frames in shot $s$, the procedure for selecting the keyframes is described as follows.

- Include the middle frame into the keyframe set $\mathbb{K}_s$ as the first keyframe $\kappa_s^1$.
- For $i = 1 : n$, do
  If $max(HistInter(f_i, \kappa_s^j)) < Th, \forall \kappa_s^j \in \mathbb{K}_s$
  Include $f_i$ into $\mathbb{K}_s$ as a new keyframe.

In this algorithm, $Th$ is the threshold for selecting a new keyframe, and we use the histograms of the keyframes as their representation.

*2) Likelihood Computation:* We define the visual similarity between two shots in terms of the Bhattacharya distance. The Bhattacharya distance between two histograms $h_1$ and $h_2$ is defined as $d_B(h_1, h_2) = -ln(\sum_{b \in allbins} \sqrt{h_1^b h_2^b})$. The visual similarity between shots $s_i$ and $s_j$ is as follows:

$$Sim(s_i, s_j) = max\left(\mathbb{C} - d_B\left(\kappa_{s_i}^m, \kappa_{s_j}^n\right)\right) \qquad (13)$$

where $\kappa_{s_i}^m \in \mathbb{K}_{s_i}$, $\kappa_{s_j}^n \in \mathbb{K}_{s_j}$, and $\mathbb{C}$ is a constant. After computing the visual similarity between all pairs of shots in the video, a similarity map is generated. One such map is shown in Fig. 5. In this map, the brighter cell represents higher similarity value. The shots that are in the same temporal scene form a bright block along the diagonal in the similarity map. If the shots $[s_a, \cdots, s_b]$ are clustered into scene $S_m$, the likelihood for this scene is computed as

$$\mathbb{L}(y_m | f_m) = avg\left(\mathbb{M}(a : b, a : b)\right) \qquad (14)$$

which is the average similarity value of the subblock in the similarity map $\mathbb{M}$ starting from row $a$ to row $b$. It is intuitive that the correct segmentation of the video gives the diagonal blocks to reach the maximum likelihood. To compute the overall likelihood, substitute (14) into (10). Up to this point, the overall likelihood $\mathbb{L}(y|\theta_k)$, the conditional prior $p(\theta_k|k)$ and the model prior

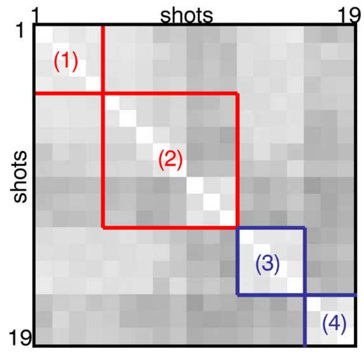


Fig. 5. Visual similarity map of the shots in a testing video. The brighter cell represents the higher similarity. The shots in the same scene possess higher similarity comparing across scenes. The bright blocks on the diagonal gives ideas of the temporal scenes. The figure shows the intermediate results for one iteration, where the red scenes (1 and 2) are not matched with correct boundaries, and the blue scenes (3 and 4) are the correct detections. (Color version available online at http://ieeexplore.ieee.org.)

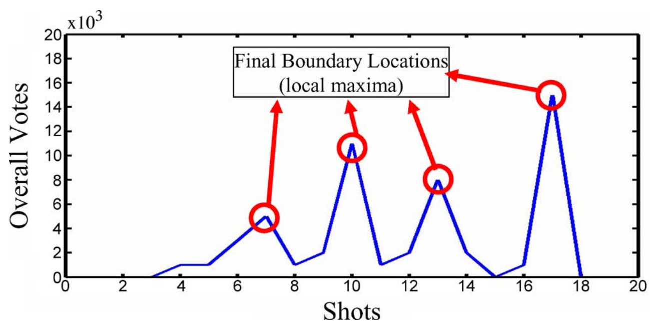


Fig. 6. Overall votes of the shots declared as the scene boundaries from multiple independent Markov chains. The red circles represent the shots that are declared as the final scene boundary locations, which correspond to the local maxima in the overall vote plot. (Color version available online at http://ieeexplore.ieee.org.)

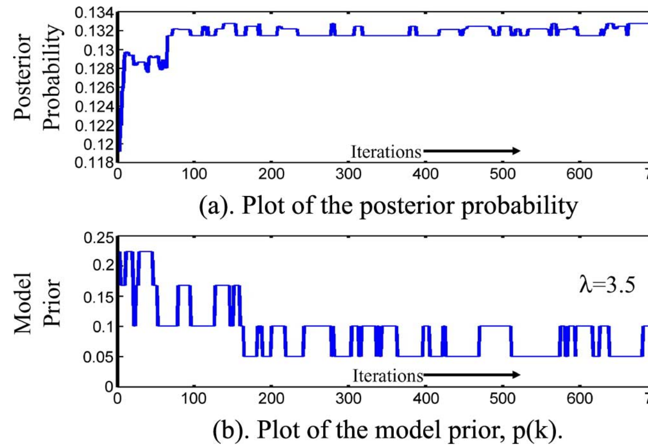


Fig. 7. (a) Plot of the posterior probability of the parameter estimation during a single Markov chain (run). As demonstrated in the figure, after certain iterations, the posterior reaches to a "confidence" level and stays there with minor fluctuations. It should be noted that if the data size (number of shots in our application) is small, the process reaches this level quickly. (b) Plot of the model prior for the number of the scenes, $k$, where the model mean, $\lambda$, is set to be 3.5. The horizontal axis in both plots represents the number of iterations. At the end of the process, plot (a) gives the posterior probability of the parameters given the video data, and plot (b) gives the information on the number of scenes, $k$. (Color version available online at http://ieeexplore.ieee.org.)

$p(k)$ are determined. Therefore, the acceptance for the proposal updates are decided by the ratio test described in the MCMC algorithm.

*3) System Performance:* The proposed method has been tested on four home videos with 23 scenes. These scenes were recorded with various environmental settings. Each scene is composed of multiple video shots. Some of them are indoor scenes (Scenes (c) and (e) in Fig. 4), while others are outdoor scenes (Scenes (a), (b), and (d) in Fig. 4). Furthermore, the videos were taken in different styles. Some scenes were recorded from the hand-held cameras (Scenes (a), (c), and (d) in Fig. 4). Some were recorded by spy camera hidden in the bag (Scene (e) in Fig. 4), and others were recorded by the camera mounted on the ground vehicles (Scene (b) in Fig. 4).

It is well known that samples generated from a single Markov chain may not result in the accurate solution. Rather, the solution generated from a single chain may be in the neighborhood of the true solution. To overcome this problem, we independently execute multiple Markov chains. The results from each individual chain provide the votes for the shots that have been declared as the scene boundaries. After certain runs, the shots with the locally highest votes represent the final scene boundaries. Fig. 6 shows the overall votes of the scene shots being declared as scene boundaries from all runs, and the red circles represent the local maxima, which correspond to the true boundaries. Even though one single chain may not provide the correct result, there is an issue of the posterior probability reaching the "confidence" level. This is referred as the "burn-in" period. As shown in Fig. 7, after certain iterations, the posterior probability reaches a level and stays there with only minor fluctuations. For this particular testing video, the "burn-in" time is short due to the small size of the data (number of shots). A simplified version of the iteration process is shown in Fig. 8.

The matches between the ground truth data and the segmented scenes are based on the matching of their starting boundaries. For a given home video with $n$ scenes, let $\{t_1, t_2, \dots, t_n\}$ denote the starting shots of the reference scenes and $\{s_1, s_2, \dots, s_k\}$ denote the starting shots of the detected scenes. Scene $t_i$ is declared as matched if one of the detected scenes $s_j$ has the same starting shot.

Two accuracy measures are used to measure the system performance: precision and recall

$$Precision = \frac{X}{A}, \quad Recall = \frac{X}{B} \tag{15}$$

where $X$ is the number of the correct matches between the system detections and the ground truth scenes; $A$ is the total number of the system detections; and $B$ is the total number of the ground truth references. The detailed precision/recall measures are shown in Table I. If the matches in all of the videos are treated equally important, the overall precision and recall are 0.840 and 0.913, respectively.

To further demonstrate the effectiveness of the proposed method, we also compare our system output with the results generated by one of the previously developed methods. As the most relevant technique to our scenario, we choose the *Backward Shot Coherence (BSC)* approach proposed in [23]. The BSC approach is a two-pass algorithm, which first segments

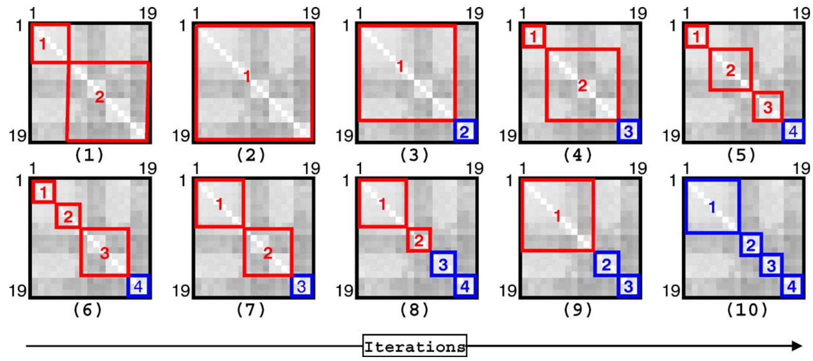


Fig. 8. Demonstration of a simplified MCMC iteration process. We show ten updates during a single run. The red boxes represent the detected scenes that do not match with the true boundaries, while the blue boxes show the detected scenes matched with the ground truth. The sample video contains 19 shots, which are initially split into two arbitrary scenes (1). After a series of updates, including shift (6), merge (2), (7), (9) and split (3), (4), (5), (8), (10), the final detected scenes (10) match with the true boundary locations. As illustrated in the figure, the scenes are eventually "locked" with the bright diagonal blocks in the similarity map. (Color version available online at http://ieeexplore.ieee.org.)

TABLE I
ACCURACY MEASURES OF FOUR HOME VIDEOS. INSERTION IS THE NUMBER OF THE OVER SEGMENTATION (FALSE POSITIVES), AND DELETION IS THE NUMBER OF THE MISDETECTIONS (FALSE NEGATIVES)

| ***Measures*** | ***clip1*** | ***clip2*** | ***clip3*** | ***clip4*** |
|---|---|---|---|---|
| Length | 12:42 | 06:53 | 07:31 | 17:53 |
| Num. of Shot | 47 | 16 | 19 | 25 |
| Num. of Scenes | 8 | 5 | 5 | 5 |
| Detected Scenes | 8 | 5 | 5 | 7 |
| Match | 7 | 5 | 5 | 4 |
| Insertion | 1 | 0 | 0 | 3 |
| Deletion | 1 | 0 | 0 | 1 |
| Precision | 0.875 | 1.000 | 1.000 | 0.571 |
| Recall | 0.875 | 1.000 | 1.000 | 0.800 |

the video into initial scenes using the color consistency and then merges them based on the similarity between their motion contents. In the home videos, the same recorder often exhibits similar motion of the camera. Furthermore, unlike other domains, motion content in home videos is less meaningful and not distinctive across the scenes. Based on the experimental observations, results obtained using both passes in the BSC algorithm are the same as the results obtained using only its first pass, which generates the scene segments using the color information. Since only the visual information is useful in our application, we compare the system performance between the results generated by the proposed MCMC method and the BSC method for the sake of fairness. The comparison results are shown in Table II.

### *B. Feature Films*

To demonstrate the generality of the proposed framework, we have also tested the proposed system on three feature films: *Gone in 60 Seconds*, *Dr. No - 007*, and *Mummy Returns*.

*1) Feature Selection:* Based on the definition provided by the Webster dictionary [15], a movie scene is one of the subdivisions of a play, or it presents continuous actions in one place. The movie scenes are composed according to the *film grammar*, which is a set of rules about how the movies are produced.

TABLE II
COMPARISON BETWEEN THE PROPOSED MARKOV CHAIN MONTE CARLO (MCMC) METHOD AND THE *Backward Shot Coherence (BSC)* [23]. THE OVERALL PRECISION AND RECALL ARE COMPUTED AS EVERY SCENE IN ALL VIDEOS IS EQUALLY IMPORTANT. THE LAST COLUMN SHOWS THE NUMBER OF THE REFERENCE SCENES IN EACH CLIP

| ***Measures*** | ***MCMC*** | ***BSC*** | ***Reference*** |
|---|---|---|---|
| Clip1 Detection | 8 | 7 | 8 |
| Clip1 Match | 7 | 4 | - |
| Clip2 Detection | 5 | 4 | 5 |
| Clip2 Match | 5 | 4 | - |
| Clip3 Detection | 5 | 6 | 5 |
| Clip3 Match | 5 | 4 | - |
| Clip4 Detection | 7 | 7 | 5 |
| Clip4 Match | 4 | 4 | - |
| Total Detection | 25 | 24 | - |
| Total Match | 21 | 16 | - |
| Total Insertion | 4 | 8 | - |
| Total Deletion | 2 | 7 | - |
| Overall Precision | 0.840 | 0.667 | - |
| Overall Recall | 0.913 | 0.696 | - |

In a scene, the shots often exhibit similar patterns, which can be reflected by the low-level features. For example, in action scenes, the shots are generally short in length, and the visual content, which indicates the activity level of the scene, changes rapidly. On the other hand, in drama scenes, the shots are much longer, and the visual content is relatively consistent. For feature films, we use these two features computed from the movies, shot length and visual content, to group the semantically coherent shots into scenes. Let $l_s$ denote the length of shot $s$ and $v_s$ be the visual content in that shot. The shot length represents the pace of the movie, and the visual content shows how much is going on in the shot. The visual content is defined as

$$v_s = \frac{1}{N_s} \sum_{i=1}^{N_s} (1 - HistInter(f_i, f_{i+1})) \qquad (16)$$

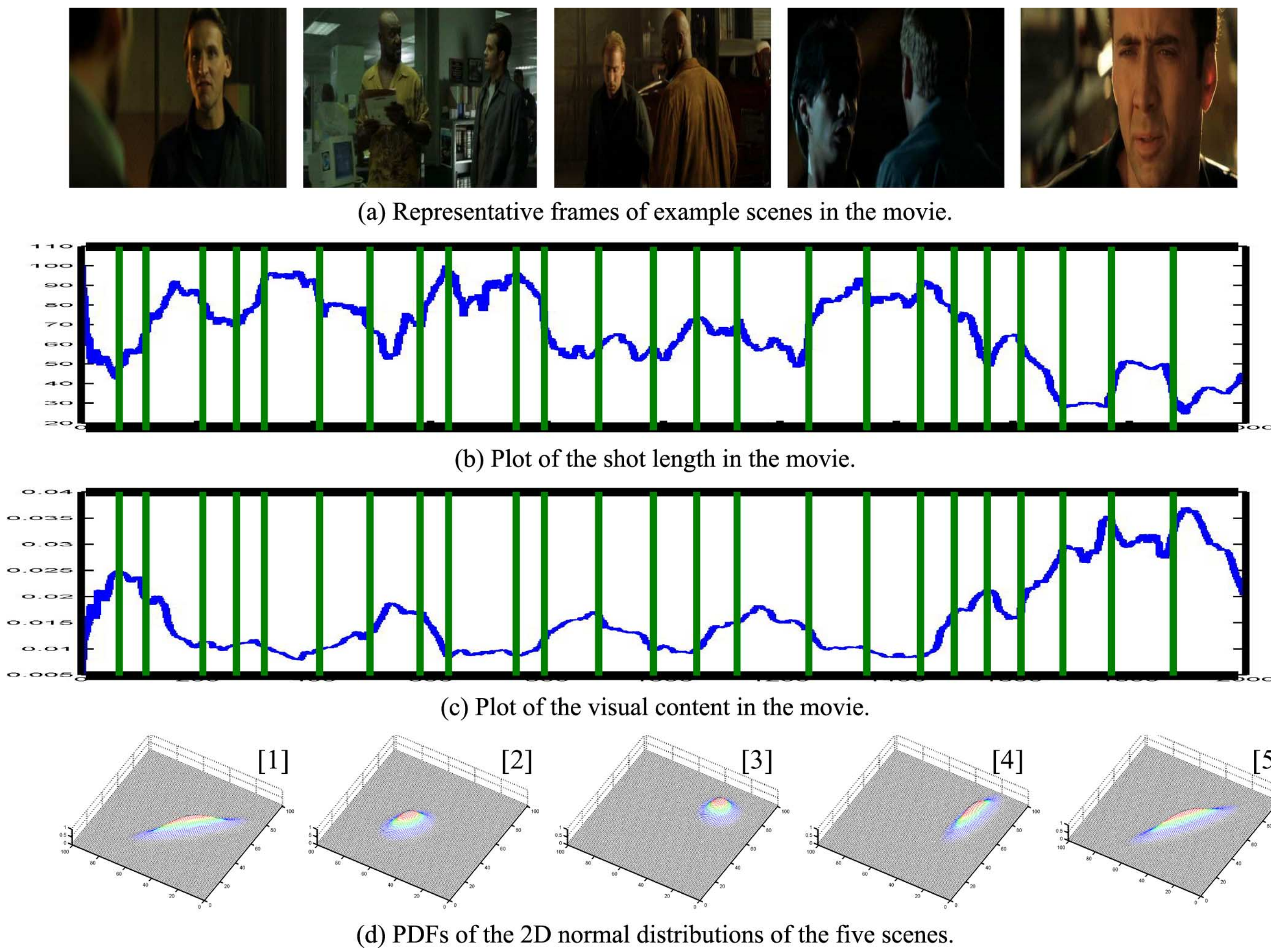


Fig. 9. (a) Representative frames of some example scenes in the movie *Gone In 60 Seconds*. (b) Plot of the shot length variable. (c). Plot of the visual disturbance feature. Usually, the shots with shorter length are accompanied by higher level of visual disturbance. The green bars represent the scene boundaries in the movie, which were detected by the proposed method; (d) PDF plots on the 2-D normal distribution of the first five scenes in the movie. The distribution parameters, mean and covariance, are different across the scenes. (Color version available online at http://ieeexplore.ieee.org.)

where $HistInter(f_i, f_{i+1})$ is the color histogram intersection between the $i$th and $(i+1)$th frames, and $N_s$ is the number of frames in shot $s$. The plots of the shot length and the visual content are shown in Fig. 9. These two features are used in the construction of the data likelihood.

*2) Likelihood Computation:* In the film production, the patterns for different features are related to each other. For instance, in action scenes, the short shots are accompanied by high degree of visual content. Therefore, the features $l_s$ and $v_s$ should not be considered independent of each other. We use a two-dimensional (2-D) normal distribution to model the features in a scene $S_m$,

$$N(g_s, m) = \frac{1}{\sqrt{2\pi S}} exp\left(-\frac{(g_s - \hat{g}_m)^T G^{-1}(g_s - \hat{g}_m)}{2}\right) \tag{17}$$

where $g_s$ is the feature vector $[l_s\ v_s]^T$. The vector $\hat{g}_m$ is computed as the sample means for the entire scene $S_m$, and $G$ is the covariance matrix with determinant $S$. Again, by considering the shots to be recorded independently, the likelihood in each scene $S_m$ is,

$$\mathbb{L}(y_m|f_m) = \left(\prod_{s=1}^{n_m} N(g_s, m)\right)^{\frac{1}{n_m}}. \tag{18}$$

We substitute (18) in (10), and perform the ratio test for the acceptance decisions. Similar argument is applied here for taking the geometric mean as in (10).

*3) System Performance:* We have experimented our approach on three feature-length films: *Gone in 60 Seconds*, *Dr. No - 007* and *Mummy Returns*. Each movie contains thousands of shots. The matching follows similar procedure as used in Section III-A3. However, the matching technique is slightly different. In movies, there usually is not a concrete or clear boundary between two adjacent scenes due to editing effects. Movie chapters are sometimes changed with a smooth transition. Therefore, matching based on the boundaries is not meaningful and often returns incorrect measures. Instead, we use a "recovery" method. Suppose there is a set of the

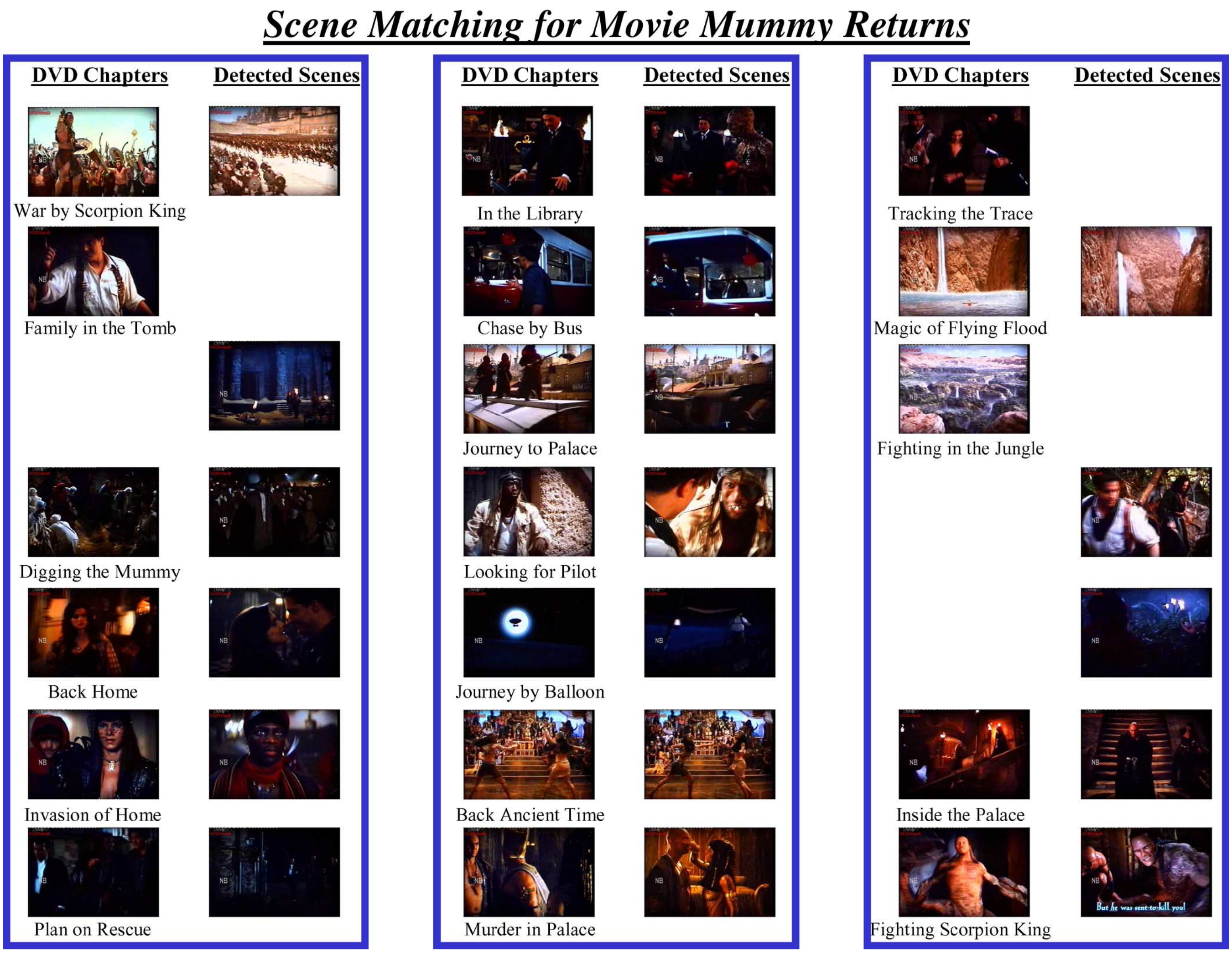


Fig. 10. Matching of the scenes for the movie *Mummy Returns*. It shows the keyframes of the ground truth scenes that are obtained from the DVD chapters and the keyframes of the detected scenes. The keyframes of the ground truth scenes are accompanied with their titles. The matches scenes are shown with their keyframes aligned. Pairs with blank spaces are the mis-matches, i.e., insertions and deletions. (Color version available online at http://ieeexplore.ieee.org.)

reference scenes $\{T_1, T_2, \ldots, T_n\}$ and a set of the detected scenes $\{S_1, S_2, \ldots, S_k\}$. A reference scene $T_m$ is said to be "recovered", if a majority of this scene ($>50\%$) overlaps one of the detected scenes. The "recovery" is a one-to-one correspondence, i.e., one reference scene can only be matched with at most one detected scene, and one detected scene can cover at most one reference scene. The scene matching for the movie *Mummy Returns* is shown in Fig. 10. In this example, we consider the chapters provided by the DVD as the ground truth scenes. The keyframes of both the ground truth scenes and the detected scenes are presented. Again, we use the precision and recall measures defined in Section III-A3 for the performance evaluation. Detailed results for movie scene segmentation are shown in Table III.

TABLE III
ACCURACY MEASURES FOR THREE FEATURE MOVIES

| ***Measures*** | ***Gone in 60 Seconds*** | ***Dr. No - 007*** | ***Mummy Returns*** |
|---|---|---|---|
| Length | 01:46:09 | 01:30:55 | 01:45:33 |
| Num. of Frames | 152665 | 130811 | 151802 |
| Num. of Shot | 2237 | 677 | 1600 |
| Num. of Scenes | 29 | 17 | 18 |
| Detected Scenes | 25 | 20 | 18 |
| Match | 24 | 14 | 15 |
| Insertion | 1 | 3 | 3 |
| Deletion | 5 | 6 | 3 |
| Precision | 0.960 | 0.700 | 0.833 |
| Recall | 0.828 | 0.824 | 0.833 |

## IV. DISCUSSIONS

The idea of the *central concept* is also applicable to other video domains. For example, in television talk shows, one major distinction between the commercials and the real TV talk shows is that, in the talk shows there often exists a repeating pattern between the host and the guest, which the commercials do not possess. The feature to distinguish this *central concept* involves the number of the repeating shots in the segment. Another example is the news video segmentation. In this task, each news segment is composed of the shots that are coherent to a certain news focus. Non-news segments include commercials, lead-in/out, reporter chit-chatting, etc. The text information, closed captions (CC) and automatic speech recognition (ASR) output, can be used as the features for constructing the posterior distribution. In this case, the semantic relations between the keywords appearing in the shots can be analyzed. Shots that have the same news focus should possess similar distributions of the keywords. The MCMC framework can find the places where the distributions of the keywords change to detect the scene boundaries.

There is another temporal segmentation process on the lower-level video structure, which is commonly known as the shot boundary detection. Shot level segmentation and the scene segmentation have their similarities and differences. A shot is defined as a series of continuous frames with consistent background settings. This assumption naturally leads to the color consistency constraints, and it does not refer to any high level semantic meanings. On the other hand, scene segmentation involves more semantic coherence. For example, in home videos, shots within the same scene are coherent to each other in terms of the same events or the same physical sites. In feature films, shots in the same scene are related to the same subtheme of the movie story line. In both the cases, the color similarity constraint is insufficient for the segmentation. The high-level semantics are often bridged by analyzing the patterns of other types of low-level features, like video pace and the visual content in the films or the narration in the news programs.

## V. Conclusion

In this paper, we have presented a general statistical framework for the temporal scene segmentation of videos. We have solved the scene segmentation task by automatically determining the places where the *central concept* changes. A target distribution of the model parameters, including the number of scenes and their corresponding boundary locations, is constructed to model the probabilities of the video shots being declared as the scene boundaries, and the solution is achieved by performing the sampling from this target distribution using the MCMC technique. In the iterative process of MCMC, the posterior probability is computed based on the model prior, conditional prior and the data likelihood given the parameters, and the updates are determined based on the posterior probabilities and the transition probabilities. The method has been applied to several home videos and three feature films, and high accuracy measures have been obtained (Tables I–III).

## References


[1] B. Adams, C. Dorai, and S. Venkatesh, "Novel approach to determining tempo and dramatic story sections in motion pictures," in *Proc. Int. Conf. Image Processing (ICIP)*, 2000, vol. 2, pp. 283–286.

[2] A. Amir, M. Berg, S. F. Chang, G. Iyengar, C. Y. Lin, A. Natsev, C. Neti, H. Nock, M. Naphade, W. Hsu, J. Smith, B. Tseng, Y. Wu, and D. Q Zhang, "IBM research TRECVID 2003 video retrieval system," in *TRECVID 2003*, 2003.

[3] A. Aner and J. R. Kender, "Video summaries through mosaic-based shot and scene clustering," in *Eur. Conf. Computer Vision*, 2002.

[4] L. Chaisorn, T.-S. Chua, and C.-H. Lee, "The segmentation of news video into story units," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, 2002, vol. 1, pp. 73–76.

[5] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun, "EM, MCMC, and chain flipping for structure from motion with unknown correspondence," *Mach. Learn., Special Issue on Markov Chain Monte Carlo Methods*, vol. 50, pp. 45–71, 2003.

[6] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and bayesian restoration of images," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-6, pp. 721–741, 1984.

[7] P. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[8] U. Grenander and M. I. Miller, "Representation of knowledge in complex systems," *J. Roy. Statist. Soc.*, ser. B, vol. 56, no. 4, 1994.

[9] B. Gunsel, A. Ferman, and A. Tekalp, "Temporal video segmentation using unsupervised clustering and semantic object tracking," *J. Electron. Imag.*, vol. 7, no. 3, 1998.

[10] F. Han, Z. W. Tu, and S. C. Zhu, "Range image segmentation by an effective jump-diffusion method," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1138–1153, Sep. 2004.

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

[12] W. K. Hasting, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[13] K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, F. Sugaya, and Y. Nakajima, "Shot boundary determination on MPEG compressed domain and story segmentation experiments for TRECVID 2004," in *TREC Video Retrieval Evaluation Forum*, 2004.

[14] W. Hsu and S. F. Chang, "Generative, discriminative, and ensemble learning on multi-model perceptual fusion toward news video story segmentation," in *Int. Conf. Multimedia and Expo*, 2004.

[15] [Online]. Available: http://www.m-w.com

[16] O. Javed, S. Khan, Z. Rasheed, and M. Shah, "Visual content based segmentation of talk and game shows," *Int. J. Comput. Applicat.*, 2002.

[17] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Eur. Conf. Computer Vision*, 2004.

[18] J. R. Kender and B. L. Yeo, "Video scene segmentation via continuous video coherence," in *Int. Conf. Computer Vision and Pattern Recognition*, 1998.

[19] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Scene determination based on video and audio features," in *Int. Conf. Multimedia Computing and Systems*, 1999.

[20] Y. Li, S. Narayanan, and C.-C. J. Kuo, "Movie content analysis indexing and skimming," in *Video Mining*. Norwell, MA: Kluwer, 2003.

[21] C. W. Ngo, H. J. Zhang, R. T. Chin, and T. C. Pong, "Motion-based video representation for scene change detection," *Int. J. Comput. Vis.*, 2001.

[22] D. B. Phillips and A. F. M. Smith, "Bayesian model comparison via jump diffusion," in *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman & Hall, 1995, ch. 13.

[23] Z. Rasheed and M. Shah, "Scene detection in Hollywood movies and TV shows," in *Int. Conf. Computer Vision and Pattern Recognition*, 2003.

[24] J. Senegas, "A Markov chain Monte Carlo approach to stereovision," in *Eur. Conf. Computer Vision*, 2002.

[25] H. Sundaram and S. F. Chang, "Video scene segmentation using video and audio features," in *Int. Conf. Multimedia and Expo*, 2000.

[26] W. Tavanapong and J. Y. Zhou, "Shot clustering techniques for story browsing," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 517–527, Aug. 2004.

[27] Z. W. Tu and S. C. Zhu, "Image segmentation by data driven Markov chain Monte Carlo," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657–673, May 2002.

[28] M. Yeung, B. Yeo, and B. Liu, "Segmentation of videos by clustering and graph analysis," *Comput. Vis. Image Understand.*, vol. 71, no. 1, pp. 94–109, Jul. 1998.

[29] M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *SPIE Conf. Multimedia Computing and Networking*, 1995.

[30] Y. Zhai and M. Shah, "A multi-level framework for video shot structuring," in *Int. Conf. Image Analysis and Recognition*, 2005.

[31] Y. Zhai, Z. Rasheed, and M. Shah, "University of Central Florida at TRECVID 2003," in *TREC Video Retreival Evaluation Forum*, Gaithersburg, MD, 2003.

[32] H. J. Zhang, J. Wu, D. Zhong, and S. W. Somaliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, no. 4, 1997.
[33] S. C. Zhu and A. L. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, 1996.

**Yun Zhai** (S'05) received the B.S. degree in computer science from Bethune-Cookman College, Daytona Beach, FL, in 2001. He is currently pursuing the Ph.D. degree at the School of Computer Science, University of Central Florida, Orlando. His research interests include computer vision, multimedia processing and video indexing and retrieval. The current emphasis of his work is on the content-based image and video retrieval.

**Mubarak Shah** (F'03) is a Professor of computer science, and the founding Director of the Computer Vision Laboratory at University of Central Florida (UCF), Orlando. He is a co-author of two books: *Video Registration* (Kluwer, 2003) and *Motion-Based Recognition* (Kluwer, 1997). He has supervised several Ph.D., M.S., and B.S. students to completion, and is currently directing 20 Ph.D. and several B.S. students. He is a researcher in computer vision and has published close to 150 papers in leading journals and conferences on topics including activity and gesture recognition, violence detection, event ontology, object tracking (fixed camera, moving camera, multiple overlapping, and nonoverlapping cameras), video segmentation, story and scene segmentation, view morphing, ATR, wide-baseline matching, and video registration.

Dr. Shah was an IEEE Distinguished Visitor speaker for 1997–2000, and is often invited to present seminars, tutorials and invited talks around the world. He received the Harris Corporation Engineering Achievement Award in 1999, the TOKTEN awards from UNDP in 1995, 1997, and 2000; Teaching Incentive Program award in 1995 and 2003, Research Incentive Award in 2003, and IEEE Outstanding Engineering Educator Award in 1997. He is an editor of and international book series on video Computing, editor in chief of *Machine Vision and Applications* and an associate editor of *Pattern Recognition*. He was an associate editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and a guest editor of the special issue of *IJCV* on video computing.