2-2012

Summarizing rushes videos by motion, object, and event understanding

Feng WANG

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Summarizing Rushes Videos by Motion, Object, and Event Understanding

Article *in* IEEE Transactions on Multimedia · March 2012

DOI: 10.1109/TMM.2011.2165531 · Source: IEEE Xplore

2 authors, including:

Chong-Wah Ngo
City University of Hong Kong
276 PUBLICATIONS **8,837** CITATIONS

SEE PROFILE

Summarizing Rushes Videos by Motion, Object and Event Understanding

Feng Wang & Chong-Wah Ngo

***Abstract*— Rushes footages are considered as cheap gold mine with the potential for reuse in broadcasting and filmmaking industries. However, mining "gold" from unedited videos such as rushes is challenging as the reusable segments are buried in a large set of redundant information. In this paper, we propose a unified framework for stock footage classification and summarization to support video editors in navigating and organizing rushes videos. Our approach is composed of two steps. First, we employ motion features to filter the undesired camera motion and locate the *stock* footage. A Hierarchical Hidden Markov Model (HHMM) is proposed to model the motion feature distribution and classify video segments into different categories to decide their potential for reuse. Second, we generate a short video summary to facilitate quick browsing of the stock footages by including the objects and events that are important for storytelling. For objects, we detect the presence of persons and moving objects. For events, we extract a set of features to detect and describe visual (motion activities and scene changes) and audio events (speech clips). A representability measure is then proposed to select the most representative video clips for video summarization. Our experiments show that the proposed HHMM significantly outperforms other methods based on SVM, FSM and HMM. The automatically generated rushes summaries are also demonstrated to be easy-to-understand, containing little redundancy, and capable of including ground-truth objects and events with shorter durations and relatively pleasant rhythm based on the TRECVID 2007, 2008 and our subjective evaluations.**

Keywords: **Rushes video structuring, Video summarization, Motion analysis, Object and event understanding.**

I. Introduction

In the broadcasting and filmmaking industries, *rushes* is a term for raw footage (extra video, B-rolls footage), which is used to generate the final products such as TV programs and movies. Twenty to forty times as much materials may be shot as actually becomes part of the finished product. Producers see these large amount of raw footages as cheap gold mine. The "gold" refers to *stock* footages which are the "generic" clips with high potentials for reuse. However, cataloguing stock footage is a tedious task, since rushes are unstructured, and the stock footage is intertwined with lots of redundant materials.

In the past decades, research on video representation and analysis has been mainly founded on edited videos, *e.g.*, news, sports and movies, which are highly structured. In contrast to edited videos, rushes are characterized as unstructured and redundant. During video capture, the same scene may be taken for multiple times, *e.g.*, when the actor forgets his lines. This results in many repetitive shots in rushes. Different kinds of behind-the-scenes footages are included, such as the clap-board, the director's command, and the discussion between the actors and the director. Rushes also contain some unintentional camera motion, *e.g.*, when the cameraman adjusts the camera to focus on the actors before movie shooting.

Table I compares rushes video with another two video domains: movie product and home video. Previous works are mostly focused on the summarization of movie product, where the main challenge is the selection of representative and informative clips through content understanding. Home videos introduces additional challenge for clip selection due to poor visual quality because of amateur camera control. For example in [17], spatiotemporal factors such as jerkiness, infidelity and blurring are utilized to select high-quality shots. Rushes videos share some properties with these two video domains. As a preliminary version of movie product, rushes are captured by professional cameramen and in high viusal quality. However, similar to home videos, rushes are not edited and thus unintentional camera motion and redundant materials are included. In general, existing works in movie product and home video domains which have their respective assumptions on visual quality and content redundancy cannot be directly applied for rushes. Instead, there is a need to develop new techniques for identifying a reduced set of useful footages from high-quality but redundant and unusable materials. However, "gold mining" in rushes is difficult as the semantic understanding of video content remains a challenging problem. Furthermore, *stock* footages and unusable materials are intertwined with each other. Video structuring needs to be carried out together with the stock footage classification.

TABLE I
COMPARISON OF DIFFERENT VIDEO DOMAINS.

Vide domain	Home video	Movie product	Rushes video
Cameraman	Amateur	Professional	Professional
Editing	No or little	Professional	No
Redundancy	Much	No	Much

In TRECVID 2007 and 2008 BBC rushes summarization tasks [24], [25], the participants are required to produce short summaries for given rushes videos. The summaries should contain as much useful footages as possible with enjoyable rhythm, but least junk and redundant materials. Most systems follow a two-step procedure. First, the irrelevant scenes and retakes are detected and removed, where shot clustering are widely employed. Second, the most important video clips are then selected to compose a short summary by ranking the shot importance based on different features such as face occurrence, image saliency and motion intensity.

F. Wang is with the Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Rd, Shanghai, 200241 China. Tel: (86) 21-54345054. Email: fwang@cs.ecnu.edu.cn

C. W. Ngo is with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong. Tel: (852)2784-4390. Fax:(852)2788-8614. Email: cwngo@cs.cityu.edu.hk

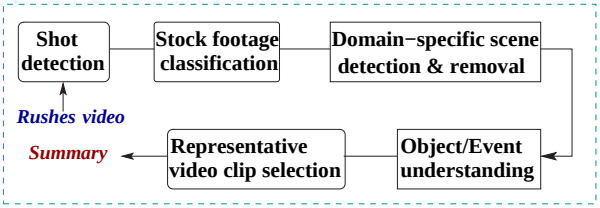

Fig. 1. Framework for content classification and summarization of rushes videos.

In this paper, we integrate two preliminary works in rushes exploration [21] and summarization [38] into a unified framework to present our approaches in details and provide comprehensive evaluations and comparisons with the existing systems. Figure 1 illustrates our framework to facilitate gold mining and quick browsing of rushes materials. We mainly address the following two problems. First, *how to classify useful footages for potential reuse?* We focus on exploring the use of motion features in locating stock footage. A Hierarchical Hidden Markov Model (HHMM) is proposed to structure the rushes video and classify each video segment into different categories according to the semantics of camera motion. The segments with intermediate camera motion are regarded as useless and filtered. The purpose of stock footage localization is to generate a clean version of the rushes videos with only useful materials so that we can grasp the desired content during video content analysis and summarization. Second, *how to organize the stock footage so that it can be efficiently browsed or searched?* Compared with edited videos, rushes videos contain duplicate clips due to the multiple takes of the same shot. We detect and remove these retakes before summary generation so as to save unnecessary time on video content analysis. An algorithm is then proposed to generate a short summary for each rushes video by selecting the most representative video clips based on the object and event understanding. By watching the produced summaries, the video editors can browse the video content quickly and decide their usefulness.

The remaining of this paper is organized as follows. Related works are discussed in Section II. Section III proposes our HHMM for rushes video structuring and stock footage classification. In Section IV, we present our approaches for irrelevant materials filtering and repetitive stock removal using domain-specific knowledge. A video summary is generated in Section V by the proposed representability measure based on object and event understanding. Section VI presents our experiment results, and Section VII concludes this paper.

II. Related Works

Rushes summarization is to produce a simplified version of the given video by reducing redundant information and the content that can be easily predicted by watching just a portion of the video. Video summarization is a challenging task due to the requirement of making decisions automatically according to the semantics of the given video. In the past several decades, different kinds of approaches have been proposed. A systematic review can be found in [35]. In [8], a set of non-redundant keyframes are obtained by fuzzy clustering and data pruning methods. Speech transcript is explored in [34] for video program summarization. In [16], a user attention model utilizing a set of audio-visual features is proposed. Object detection is employed in [12] for video abstraction in surveillance system. In [43], a perception curve that corresponds to human perception changes is constructed based on a number of visual features, including motion, contrast, special scenes, and statistical rhythm. The frames corresponding to the peak points of the perception curve are extracted for summarization. In [32], representative keyframes and metadata about video structure and motion are generated to summarize the video with the least information loss.

In contrast to the edited videos, rushes videos contain another kind of redundant information, i.e., inter-shot redundancy. Some junk shots may be inserted during video recording, and usually the same scene is taken for many times. For rushes summarization, these redundancy needs to be filtered and the useful materials should be located. In the annual TRECVID workshop [46] since 2005, different approaches have been proposed for rushes exploitation, including junk information filtering, retake detection, high-level feature detection, video browsing and summarization. In [33], a system is proposed to single out redundant and repetitive rushes data. High-level features, such as faces and buildings, are detected to help the editors select the useful content. In [10], [39], [40], after video structuring, camera motion classification and concept detection are performed for content analysis. In [2], [3], different features including motion activity, audio volume, face occurrence, color and object similarity are extracted for shot clustering. The representative items are then selected from each cluster to create tools for video content visualization, browsing and summarization. In [1], keywords are manually assigned to each shot. The shots are united into stories manually. Metadata is used for fast browsing. In [33], spatiotemporal slice is employed to quickly detect the repetitive shots to remove the inter-shot redundancy. In [36], the shots are clustered by SIFT features and one keyframe is selected from each shot cluster based on a number of rules, *e.g.*, selecting the most dominant face, or selecting the longest camera distance if no face exists. All these works aim at selecting the potentially useful footages by employing different features so as to faciliate more efficient browsing of rushes videos.

III. Stock Footage Classification and Localization by Motion Features

The stock footage localization is to extract the materials with high potential for reuse by the editors from the rushes collections. Three semantic categories are considered: *stock*, *outtake* and *shaky*. The concept *stock* represents the clips with intentional camera motion which have the potential for reuse, such as capturing an event with still camera and rotating the camera for a panoramic view. In contrast, those clips with intermediate camera motion, which are very likely to be discarded in the final production, are denoted as *outtake*. Examples include a quick zoom-in to get more details and a pan to change to another perspective. The third category,

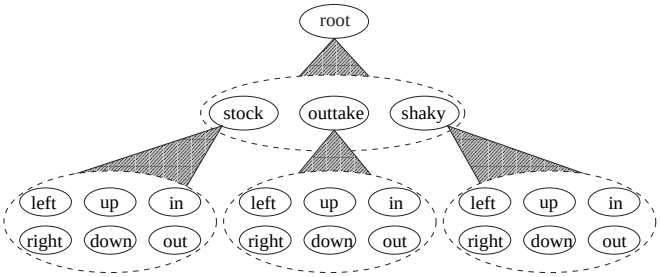

Fig. 2. An illustration of our two-level HHMM. Solid ellipses denote the substates, while dotted ellipses denote the sub-HMMs of the HHMM structure. (Notice that the substates in each sub-HMM are fully connected. For the simplicity of presenting the figure, we do not show the edges.)

shaky, represents the shaky artifacts which could be discarded from summarization.

Since rushes are raw footages without editing, the problem of structuring and categorization are intertwined. It is infeasible to structure the videos without knowing the underlying characteristics of frames. For example, structuring only by motion cannot obtain satisfactory performance due to the indiscriminative motion features of the three semantic concepts. In other words, there are two kinds of temporal structures that are intertwined: the camera motion transitions inside each category and the category transitions in the rushes videos. Simultaneous modeling of both temporal structures is required. Approaches such as [17] which measures the characteristics of video segments independently could not be directly adopted for not modeling the temporal relationship between segments. In this section, we propose a Hierarchical Hidden Markov Model (HHMM) for modeling the intertwined relationship between structuring and categorization.

HHMM is the generalization of HMM with hierarchical structure [42]. We use a two-level HHMM to encode the three semantic categories. Figure 2 illustrates the structure of our HHMM. On the top is an auxiliary root state. The first level is a sub-HMM which has three substates to represent *stock*, *outtake* and *shaky* respectively. Each substate is also a sub-HMM which is further decomposed into several substates in the lower level. Basically a substate in this level models certain aspect of low-level features to support the encoding of semantic concepts at the higher level. For each semantic concept, we use six substates, *left*, *right*, *up*, *down*, *in* and *out*, to model the six major movements respectively in horizontal, vertical and depth directions. This hierarchical model, on one hand, can alleviate the feature overlap problem by taking into account the temporal constraint. On the other hand, the higher-level substates make it possible to simultaneously structure and categorize the rushes on the whole sequence.

A. Motion Feature Extraction

In order to facilitate structuring and categorization, a shot should be partitioned into smaller segments which form an observation sequence for HHMM. In this paper, we investigate two kinds of settings: *fixed* and *adaptive* segments. The former one is obtained through equal partitioning of a shot into segments of fixed length, while adaptive segments are obtained by dividing a shot into segments each with consistent motion. Both types of segments have their strength and weakness. The fixed segment is easy to obtain in practice, but with inaccurate boundary and motion feature. Intuitively, adaptive segment may have better performance due to good boundary and motion feature. However, since shot segmentation by motion itself is a research issue, false and missed detections would introduce under- or over-segmentation that prohibit the finding of underlying semantic labels.

To obtain the observation sequence for HHMM, we extract three types of dominant motion: pan/track, tilt/boom and zoom/dolly from each segment. The inter-frame motion features are firstly estimated from each two adjacent frames. We apply Harris corner detector to extract the keypoints, $\mathbf{x}_t$, from the frame t. Their corresponding points, $\mathbf{x}_{t+1}$, in the next frame $t+1$, are estimated by the Singular Value Decomposition (SVD) of the 3D tensor structure [26]. Since the dominant features for rushes structuring and categorization are pan/track, tilt/boom and zoom/dolly, 2D camera motion model is sufficient for the representation of these three motion features. Therefore, we use the 2D 6-parameter affine model described as

$$\mathbf{x}_{t+1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

where $[a_{11}, a_{12}, a_{21}, a_{22}, v_1, v_2]^T$ are estimated from the matched points in the frame pair using the robust estimator LMedS [30]. The parameter v_1 and v_2 characterize the pan/track and tilt/boom respectively, while the parameter a_{11} and a_{22} describe the zoom/dolly motion. We extract a 3-dimensional motion feature vector $f = [v_1, v_2, z = (a_{11} + a_{22})/2]$ for each two adjacent frames. A sequence of motion vectors, $\{f\}$, is then obtained from the frame sequence in a segment. We use the median $o = \text{median}\{f\}$ as the observation for a segment. Then a T-segment string of a shot forms an observation sequence for HHMM, denoted as $O = (o_1, o_2 \cdots o_T)$.

B. HHMM Representation

A state in an HHMM consists of a string of substates from top to bottom levels. We use $k^d = q_{1:d} = \overline{q_1 q_2 \cdots q_d}$ to denote the substate string from top to level d, where the subscripts denote the hierarchical levels. We drop the superscript d for abbreviation when there is no confusion. Let D denote the maximum number of levels and Q denote the maximum size of any sub-HMM state spaces in HHMM. An HHMM can then be specified by $\Theta = \{\mathcal{A}, \mathcal{B}, \Pi, \mathcal{E}\}$. Explicitly, $\mathcal{A}$ denotes the transition probabilities $(\bigcup_{d=1}^{D} \bigcup_{k=1}^{Q^{d-1}} \{a_k^d\})$, where a_k^d is the transition matrix at level d with configuration k^{d-1}. $\mathcal{B}$ is the emission parameter which specifies the observation distributions. We assume that the motion features comply with Gaussian distribution $N(\mu, \Sigma)$, then $\mathcal{B} = (\bigcup_{i=1}^{Q^D} \{\mu_i, \Sigma_i\})$. Similarly, let π_k^d and e_k^d denote the prior and exiting probabilities at level d, then $\Pi = \bigcup_{d=1}^{D} \bigcup_{k_d=1}^{Q^{d-1}} \pi_k^d$ and $\mathcal{E} = \bigcup_{d=1}^{D} \bigcup_{k_d=1}^{Q^{d-1}} e_k^d$ are the prior and existing probabilities for HHMM model.

Fig. 3. An example of stock footage classification. The first row shows some snapshots of the video sequence. The second row lists the detected camera motion along the sequence. L: Left; R: Right; U: Up; D: Down; I: In; O: Out. The third row illustrates the state transitions at the higher level of HHMM.

C. HHMM Training and Classification

Given an observation sequence $O = (o_1, o_2 \cdots o_t \cdots o_T)$, HHMM training is to find Θ^* that maximizes the likelihood $L(\Theta)$. This is estimated by the Expectation-Maximization (EM) algorithm as in traditional HMM. Given an old parameter Θ and the missing data $K = (k_1, k_2, \cdots k_t \cdots k_T)$, the expectation of the complete-data likelihood of an updated parameter $\hat{\Theta}$ is written by

$$\begin{aligned} L(\hat{\Theta}, \Theta) &= E(\log p(O, K|\hat{\Theta})|O, \Theta) \\ &= \sum_K p(K|O, \Theta) \log p(O, K|\hat{\Theta}) \\ &\propto \sum_K p(O, K|\Theta) \log p(O, K|\hat{\Theta}) \end{aligned} \quad (1)$$

The E-step estimates the expectation $L(\hat{\Theta}, \Theta)$, and the M-step finds the value $\hat{\Theta}$ that maximizes the likelihood.

We define the probability of being in state k at time t and in state k' at time $t+1$ with transition at level d, given O and Θ, as $\xi_t(k, k', d) \stackrel{def}{=} p(k_t = k, k_{t+1} = k', e_t^{1:d} = 0, e_t^{d+1:D} = 1|O, \Theta)$. Similarly, we define the probability of being in state k at time t, given O and Θ, as $\gamma_t(k) \stackrel{def}{=} p(k_t = k|O, \Theta)$. In E-step, these two auxiliary variables are estimated by forward and backward algorithm [42]. In M-step, we can get the updated model parameter $\hat{L}$ as follows,

$$\hat{\pi}_q^d(i) = \frac{\sum_{t=1}^{T-1} \sum_{q'} \sum_{q''} \xi_t(q', \overline{qiq''}, d-1)}{\sum_{t=1}^{T-1} \sum_{q'} \sum_{q''} \sum_{i} \xi_t(q', \overline{qiq''}, d-1)} \quad (2)$$

$$\hat{e}_q^d(i) = \frac{\sum_{t=1}^{T-1} \sum_{q'} \sum_{k'} \sum_{d'<d} \xi_t(\overline{qiq'}, k', d')}{\sum_{t=1}^{T-1} \sum_{q'} \gamma_t(\overline{qiq'})} \quad (3)$$

$$\hat{a}_q^d(i,j) = \frac{\sum_{t=1}^{T-1} \sum_{q'} \sum_{q''} \xi_t(\overline{qiq'}, \overline{qjq''}, d)}{\sum_{t=1}^{T-1} \sum_{q'} \sum_{q''} \sum_{j} \xi_t(\overline{qiq'}, \overline{qjq''}, d)} \quad (4)$$

The above three equations estimate the prior probability, within-level transition probability and level-exiting probability respectively by normalizing ξ and γ. The temporal dimension of ξ and γ are marginalized out. Here we are only interested in the transitions made at level d, and q', q'' are the states at levels lower than d. The means and covariances of state k at the bottom level are then estimated by

$$\hat{\mu}_k = \frac{\sum_{t=1}^{T} o_t \gamma_t(k)}{\sum_{t=1}^{T} \gamma_t(k)} \quad (5)$$

$$\hat{\Sigma}_k = \frac{\sum_{t=1}^{T} (o_t - \mu_k)(o_t - \mu_k)^T \gamma_t(k)}{\sum_{t=1}^{T} \gamma_t(k)} \quad (6)$$

With the estimated parameters, HHMM is then used to classify each segment into the three semantic categories. Given an observation sequence of a shot $O = (o_1, o_2 \cdots o_t \cdots, o_T)$, we apply Viterbi algorithm [42] to obtain the underlying optimal state sequence, $K^* = (k_1^*, k_2^* \cdots, k_t^*, \cdots, k_T^*)$. Each k^* actually has two variables to indicate the substates of semantic label and motion feature in the two-level HHMM. The final solution is found in the higher-level variable string $K^{1*} = (k_1^{1*}, k_2^{1*} \cdots, k_t^{1*}, \cdots, k_T^{1*})$, which forms the labels of the semantic concepts for the segments. Meanwhile, the variations in the variable string K^{1*} indicate the locations of the semantic concept boundaries. Therefore, by using Viterbi algorithm on the segment string, the simultaneous structuring and categorization for a rushes shot can be efficiently achieved.

Figure 3 illustrates an example of stock footage localization. In the given video sequence, the detected camera motion composes the states in the bottom level of the HHMM (Figure 2) which models the motion pattern inside each semantic category. With the two-level HHMM, we can simultaneously structure and classify the sequence into different categories. As shown in Figure 3, an *outtake* and a *shaky* are inserted to adjust the camera setting before shooting the desired *stock* footage. This pattern frequently happens in rushes videos and is modelled by the high level of HHMM. Finally, the video segments of the concept *outtake* and *shaky* are pruned, while those of *stock* are retained for video summarization.

IV. Domain-Specific Scene Classification

In the remaining stocks after filtering the undesirable camera motion, there are still two kinds of footages that are less useful for editors, i.e., clapboard and retake scenes. The former does not contain any information relevant to video content and should be removed, while the latters are stocks repeatedly taken and contain redundant information.

Fig. 4. Detection of clapboard scenes. First row: keyframes from test video set; Second row: example clapboard scenes extracted from training set (Different colors are used for matching lines just to make the lines clearly viewable in different images).

A. Clapboard Detection

In rushes videos, each stock usually contains not only the movie play, but also some other materials that are irrelevant to the storytelling, such as camera adjustment and scene arrangement before movie shooting, and discussions between the director and the actors. In video summarization, we detect the clapboard scenes to partition a stock into subshots and separate the story-relevant materials from other elements by employing visual and audio features.

To detect the clapboard scenes, we employ the algorithm for Near-Duplicate Keyframe (NDK) detection in [22], [44]. A set of 50 example keyframes of the retake scenes are extracted from the training video set as shown in Figure 4. The regions of the boards are manually annotated. For the keyframes in the given rushes video, we detect the keypoints and match them with the example clapboard scenes. Figure 4 shows some matching lines between keyframes and the matched example boards. If enough matching lines are found in the annotated regions, the keyframe is detected as a clapboard scene.

Besides visual features, the clapboard scenes can be detected in speech transcripts. In most cases, the director controls the progress of movie capture by calling out keywords such as "standby", "action", "cut", "take xx", and "shot yy" (xx, yy are the sequence number of the current take). We employ an ASR (Automatic Speech Recognition) engine for speech recognition and then detect these keywords in the output transcripts. The movie play is located by a pair of keywords "action" and "cut".

The third feature is audio. In some cases, although the camera is faced to the actors, they might be discussing with the directors instead of acting. These kinds of footages cannot be detected by visual or speech cues. However, we observe that in audio track there is quite obvious boundary between movie play and unintentional materials, since the source of audio, manner of speaking and background noise are different in these two scenes. We classify the corresponding segments into three classes: silence, actor's lines, and noise. A number of features are extracted from audio track, including cepstral-flux, multi-channel cochlear decomposition, cepstral vector, low energy fraction, volume standard deviation, non-silence ratio, standard deviations of pitch and zero crossing rate, and smooth pitch ratio. An SVM is then employed for the classification of different audio scenes.

B. Retake Detection and Removal

During video capture, due to the mistakes of the actors and in order to achieve better effects, each stock is usually taken for many times. This results in many repetitive stocks. We detect all the retakes and select only one for summary generation. Retake detection is carried out by matching subshots in different stocks. The similarity of two subshots are calculated based on keyframe and ASR speech transcript comparison. Given two subshots s_i and s_j, two different cases are considered: i) s_i and s_j are repetitions of each other; ii) s_i is a part of s_j. For the second case, s_i is an incomplete version of s_j and is removed from the subshot list. For the first case, all subshots are complete. They are different takes of the same scene until the director gets an satisfactory one. In this case, we choose the last version and remove all the other repetitions by assuming that the capture of the same scene is stopped until the director gets a desired version.

V. Video Summarization by Object and Event Understanding

After removing the less useful information in the rushes video, in this section, a short video summary is generated to help the users quickly browse the video content. The summary is expected to include as much information as possible in limited duration with pleasant rhythm. Unlike most previous approaches by shot clustering based on low-level color, texture and audio features, we carry out summarization by exploring more semantic information, i.e., objects and events, to have better understanding of the video content.

A. Object and Event Understanding

For video summarization, the first step is to understand the video content. Since video is used to present objects and describe events, video content analysis, in nature, is about object and event detection. For objects, we mainly detect the presence of persons and moving objects, which are essential for content analysis of rushes videos. An event can be described from the following five aspects: ***Who, What, Where, When*** and ***How***. In our approach, we attempt to detect event occurrences by extracting related features to locate these five elements. To answer the question of "***What*** happen in the video" needs event recognition, which remains a challenging problem in general video domains. Fortunately, for video summarization, the task is to detect the presence of events so as to include them in the summary, while the recognition is usually unnecessary. Event detection can be carried out in both video and audio tracks, and most events are visually presented by the object activities, camera motion and scene changes.

Object detection actually answers the question of "***Who*** participate in the event". We consider two kinds of objects in video sequence. First, human plays an important role in different kinds of videos, especially in the rushes collections of movie products. We detect human faces by employing the face detector from CMU [45]. Second, an object in movement usually implies an event occurrence. To capture the objects in videos, we detect and track moving objects based on our previous work [27]. Some examples of face

Fig. 5. Object and face detection.

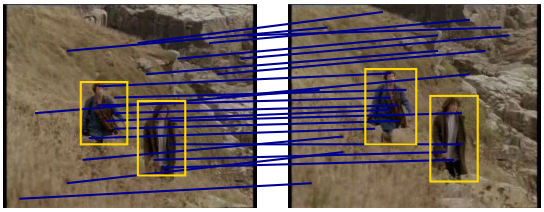

Fig. 6. Keypoint tracking in sampled frames.

and object detection are shown in Figure 4. Each object is represented as $o(o_{id}, t_s, t_e, p_o(t))$, where o_{id} is the identity of the object, t_s and t_e denote the time when the object appears and vanishes before the camera (***When***), and $p_o(t)$ includes a list of locations of the object at time t ($t_s \leq t \leq t_e$) (***Where*** and ***How***).

The scene actually answers the question of ***Where***. A new scene indicates the beginning or a new stage of an event. To detect the scene changes in events, we employ the algorithm in [44], [22] to track the keypoints in sampled frames. Basically we evenly select 3 frames every second. For each sampled frame, keypoints are detected and matched with the subsequent 5 frames by employing the algorithm in [44], [22]. An example of keypoint tracking is shown in Figure 6. Based on the results of object detection, a matching keypoint in two frames is assigned to an object if it lies on the object tracked in both frames. Otherwise, it is assigned to the background scene. The number of matching keypoints in the neighboring keyframes measures the consistency of the scenes. A scene change is detected if the number of matching lines between two keyframes is less than a threshold.

Camera motion is another indicator of event evolution. Different kinds of camera motion imply the intentions of the cameraman or director. For instance, a camera pan includes new objects and background scenes, while a zoom-in emphasizes some objects or people. In Section III-A, we have detected three types of dominant camera motion: pan/track, tilt/boom and zoom/dolly from each subshot. This helps us to capture the intentions of the movie makers and thus include the desired materials.

Besides visual events, we consider people's speech and dialogue based on speech recognition in audio track. Audio is useful for video summarization since there is usually little visual changes during a long dialogue or speech. Such events are important for the semantic completeness of the summary, but cannot be captured by visual detectors. In Section IV-A, we have employed an ASR engine to extract human's speech inside each stock. After stop words removal, we get a set of words. An audio event is detected if the density of meaningful words is high enough in a video clip.

By audio-visual event understanding, we get the following features:

a) A set of objects $O = \{o_i\}$. Each object is associated with its existence period and location information;
b) A set of object motion activities $\Phi = \{\phi_j\}$. Each element is associated with an object and its movements along the video sequence;
c) A list of camera motion $\Gamma = \{\gamma_k\}$. Each element is associated with the camera motion parameters and time information.
d) The scene changes $\Delta = \{\delta_l\}$ between neighboring frames with the period of each scene being recorded.
e) Dialogue or speech clips $\Omega = \{\omega_m\}$. Each clip is associated with the speech transcripts, and the beginning and ending time of each dialogue.

Although the above features cannot tell what exactly happen in the video, they are good indicators for the object and event occurrences. Furthermore, they also enable us to extract the elements of *Who*, *When*, *Where*, and *How* to describe the events. This is sufficient for video exploitation and summarization.

B. Representability Score

Video summary is to select the most representative video clips to explicitly describe video content. For this purpose, we propose a representability score for the candidate video clips based on object and event understanding. First, a given subshot is segmented into 1-second video clips. Each clip overlaps with the previous one by 300ms to enable the flexibility of clip segmentation and selection. These clips are used as candidates for composing video summaries.

Given a video clip v, five scores are defined to measure the representability of v for the five feature sets $\{O, \Phi, \Gamma, \Delta, \Omega\}$ respectively as follows:

$$R_v(O) = \sum_{o \in O} \int_{t_1}^{t_2} (1 - \frac{|t - (t_{so} + t_{eo})/2|}{t_{eo} - t_{so}}) dt \quad (7)$$

$$R_v(\Phi) = \frac{\sum_{\phi \in \Phi} \int_{t_1}^{t_2} f(t) dt}{\sum_{\phi \in \Phi} \int_{t_{s\phi}}^{t_{e\phi}} f(t) dt} \quad (8)$$

$$R_v(\Gamma) = \sum_{\gamma \in \Gamma} \int_{t_1}^{t_2} (1 - \frac{|t - (t_{s\gamma} + t_{e\gamma})/2|}{t_{e\gamma} - t_{s\gamma}}) dt \quad (9)$$

$$R_v(\Delta) = \frac{\sum_{\delta \in \Delta} \int_{t_1}^{t_2} \delta(t) dt}{\sum_{\delta \in \Delta} \int_{t_{s\delta}}^{t_{e\delta}} \delta(t) dt} \quad (10)$$

$$R_v(\Omega) = \frac{\|W(v) \bigcap W(\Omega)\|}{\|W(\Omega)\|} \quad (11)$$

where (t_1, t_2) denotes the temporal intersection of clip v and the corresponding object or event, (t_{so}, t_{eo}) denotes the existence period of object or event o, $f(t)$ is the motion intensity function at time t, and $W(v)$ is the set of words in speech transcript for a given video clip v. Equations 7 and 9 measure to what extent the clip v can include the presence of each object $o \in O$ and camera motion $\gamma \in \Gamma$ respectively. We assign larger weights to the video clips when the object or camera motion starts or ends (farther away from the midpoint of $[t_{so}, t_{eo}]$) as we think this is more important events, while

the progress of the object movement or camera motion can be easily predicted if the starting and ending clips have been included in the summary. In Equation 8, the representability of clip v for an activity event ϕ is measured by the amount of motion included by v. Equations 10 and 11 measure how many scene changes and words in speech transcripts are included by video clip v respectively.

Based on object and event understanding, we can identify the content inclusion of each video clip v_i as $Inc(v_i) =< O_{v_i}, \Phi_{v_i}, \Gamma_{v_i}, \Delta_{v_i}, \Omega_{v_i} >$, where $O_{v_i} \subseteq O$,$\Phi_{v_i} \subseteq \Phi$, $\Gamma_{v_i} \subseteq \Gamma$, $\Delta_{v_i} \subseteq \Delta$,$\Omega_{v_i} \subseteq \Omega$ are the sets of objects, motion activities, camera motion, scene changes, and speech clips respectively that lie in v_i. By equations 7-11, a representability score of a video clip v_i for v_j is defined as

$$Rep(v_i, v_j) = \frac{1}{\sqrt[4]{d(v_i, v_j)}} \cdot (w_O R_{v_i}(O_{v_j}) + w_\Phi R_{v_i}(\Phi_{v_j}) + w_\Gamma R_{v_i}(\Gamma_{v_j}) + w_\Delta R_{v_i}(\Delta_{v_j}) + w_\Omega R_{v_i}(\Omega_{v_j})) \quad (12)$$

where $d(v_i, v_j)$ is the temporal distance between the midpoints of v_i and v_j, $w_O = w_\Phi = w_\Gamma = w_\Delta = w_\Omega = 0.2$ are the weights for the five different features respectively. The score $Rep(v_i, v_j)$ measures to what extent v_i can represent the content in v_j. For instance, if $Rep(v_i, v_j)$ is high enough, i.e., most objects and events in v_j are also found in v_i, it is better to keep the more representative clip v_i in the summary and remove v_j to reduce the summary length and redundancy.

Based on Equation 12, for each video clip, we measure its representability for the neighboring clips. Figure 7 shows the representability curves of two video clips c_1 and c_2. The overall representability score of a video clip c is calculated as

$$s(c) = \sum_{c'} Rep(c, c') \quad (13)$$

where c' is the neighboring clip of c. In Figure 7, $s(c_1)$ and $s(c_2)$ correspond to the area below the two curves. The larger the representatability score is, the better the video clip can represent the neighboring ones. We just consider the representability of a clip for its 150 neighbors in order to smooth the storytelling of the summary. For instance, if one person appears again after a long time, the second clip should not be simply removed because this might correspond to a new event.

C. Summary Generation

Based on the calculated representability score, we select a set of video clips Sum that include all detected features and have the highest ratio of representability and total duration. Our algorithm works as follows.

1) *Initialize $Sum = \{\}$, and C is the set of all video clips in the descending order of their representability scores.*
2) *Select the clip $c_i \in C$ with the highest representability score. $Sum = Sum \bigcup \{c_i\}$.*
3) *Remove all clips $c_p \in C$ if $Rep(c_i, c_p)$ is larger than a threshold.*
4) *Goto 2 if C is not empty.*

With the above algorithm, we exclude the duplicate clips that can be represented by others. Next, we update Sum to improve the overall representability. For each video clip c_r that is not selected, we attempt two operations: inserting c_r to Sum or using c_r to replace one of its neighboring clip in Sum. The clips that can improve the representability the most are inserted or used to replace another one. This procedure continues until the summary length reaches the upper limit L (*e.g.*, 2% of the original video duration in TRECVID 2008) or the incremental representability score Δs gained by inserting one clip is less than a minimum value $MIN_s = \frac{1}{3L}$. Finally, all video clips in Sum are adjoined together to compose a video summary.

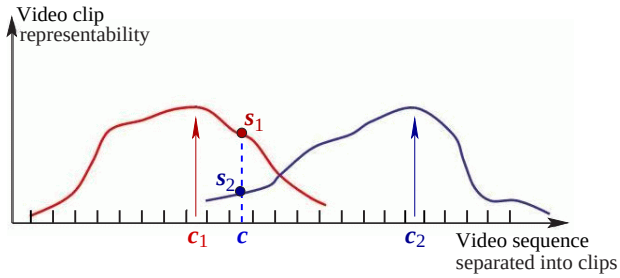

Fig. 7. Representability scores of two video clips c_1 and c_2. Given another video clip c, the values of $s_1 = Rep(c_1, c)$ and $s_2 = Rep(c_2, c)$ indicate how much information in c can be represented by c_1 and c_2 respectively.

The novelty of our algorithm comes from the employment of object and event understanding for selecting semantically meaningful clips. This is in contrast to existing works such as [7], [19] which sample clips based on activity or visual intensities through low- or mid-level feature analysis. In addition, the proposed representability measures the importance of a clip by judging from its contribution to the video content. This is also different from conventional strategies where the selection criteria are rule-based or depending on pairwise shot similarity.

VI. Experiments

We conduct experiments on the video data from TRECVID 2007 and 2008 BBC rushes summarization task [24], [25] to demonstrate the performance of both stock footage classification and rushes summarization. The video data consists of raw video footages mainly for five series of BBC drama programs. The TRECVID 2007 dataset includes 43 videos (about 18 hours) for development and 42 videos (17 hours) for testing. The video duration ranges from 3.3 to 36.4 minutes. In TRECVID 2008, another 39 videos (17.2 hours) are provided as testing data. For experiments, the videos are first partitioned into shots by employing our work in [20].

A. Stock Footage Classification

We compare the proposed HHMM with our previous work presented in TRECVID 2005 [23]. In [23], we experiment three approaches: Finite State Machine (FSM), Hidden Markov Model (HMM) and Support Vector Machine (SVM). In this paper, we use TRECVID 2007 BBC development dataset for training, and TRECVID 2008 testing set for evaluation. In the groundtruth data, 63.5%, 15.9% and 20.6% of the shots are belong to *stock*, *outtake* and *shaky* respectively.

Table II summaries and compares the properties of different approaches. FSM is actually a simplified HMM that the fuzzy transitions in HMM become deterministic. SVM, instead of modelling feature distribution, discriminates the three semantic concepts by hyper-plane in feature space. We use Radial Basis Function (RBF) as the kernel for SVM. Meanwhile, Gaussian distribution is used as kernel function in HMM and HHMM. Adaptive video segmentation is applied for FSM, while fixed segments of 1-second duration are used for SVM and HMM. HHMM is tested with both adaptive (A-HHMM) and fixed video segments (F-HHMM).

TABLE II
COMPARISON OF DIFFERENT METHOD'S PROPERTIES.

	Segment length	# Structure	# Feature	Kernel
FSM	adaptive	flattened	3	threshold
SVM	fixed	flattened	9	RBF
HMM	fixed	flattened	9	Gaussian
F-HHMM	fixed	hierarchical	3	Gaussian
A-HHMM	adaptive	hierarchical	3	Gaussian

TABLE III
SEMANTIC CATEGORIZATION ACCURACY ON THE TESTING VIDEO SET.

Approaches	Stock		Outtake		Shaky	
	Recall	Prec.	Recall	Prec.	Recall	Prec.
FSM	0.764	0.986	0.794	0.120	0.075	0.116
SVM	0.795	0.982	0.721	0.175	0.621	0.334
HMM	0.870	0.976	0.383	0.148	0.348	0.412
F-HHMM	**0.959**	**0.983**	**0.612**	**0.591**	**0.413**	**0.589**
A-HHMM	**0.977**	**0.988**	**0.537**	**0.633**	**0.731**	**0.695**

Table III shows the results of rushes footage classification for testing videos. The results are evaluated based on the number of frames being correctly or wrongly classified. From Table III, we can see that HHMM outperforms the other approaches. Overall, we have about 96% accuracy on *stock*, 60% on *outtake* and 70% on *shaky* in the testing set. Compared with the other two categories, *stock* footages are usually captured with perfect and consistent camera control. The motion pattern in *stock* is relatively simpler and easier to model with HHMM. Thus, *stock* is more separable from *shaky* and *outtake*. The classification errors mainly happen between *shaky* and *outtake* due to the complicated and varying camera motion in these two categories. This does not hurt much the performance of the whole system since both concepts are finally discarded. Since SVM assumes that the observations are independent and neglects the temporal relationship, the classification accuracy is pretty low. Instead, by exploiting the temporal relationship, HMM presents some improvement compared to SVM. Through experiments, hierarchical HMM shows better performance than flat HMM. A-HHMM shows slight improvement in *stock* classification compared to F-HHMM, and significant improvement in the classification of *shaky* segments.

B. TRECVID BBC Rushes Summarization

Given a video from the rushes test collection, the task is to automatically create an MPEG-1 summary clip less than or equal to a maximum duration that shows the main objects and events in the rushes video to be summarized. The summary should minimize the number of frames used and present the information in ways that maximize the usability of the summary and speed of recognizing objects and events in videos. In the experiments, based on domain specific knowledge presented in Section IV, the performance of clapboard detection is: $Recall = 95\%$ and $Precision = 78\%$. Most retakes can be removed with an accuracy of 94%.

1) TRECVID 2007 Evaluation: In TRECVID 2007, there are 1008 summaries (including 84 summaries of two baseline runs generated by CMU [9]) of 42 videos submitted from 22 teams for judgment [24]. The summaries' lengths are limited to be less than 4% of the original videos. During evaluation, five assessors are asked to watch and score the submitted summaries. Seven criterions are used for the subject evaluation: **EA**, **RE**, **IN**, **DU**, **XD**, **TT**, and **VT** (the detailed definitions of these criterions can be found in Table IV). The performance is judged based on the guidelines and evaluation provided by TRECVID 2007 BBC rushes summarization task [24].

TABLE IV
OUR RESULTS ON TRECVID 2007 RUSHES SUMMARIZATION TASK. (THE NUMBERS ARE THE MEDIANS OF THE SCORES FOR ALL 42 VIDEOS.)

EA - Easy to understand: 1 strongly disagree - 5 strongly agree;
RE - Little duplicate video: 1 strongly disagree - 5 strong agree;
IN - Fraction of inclusions found in the summary (0 - 1);
DU - Duration of the summary (sec);
XD - Difference between target and actual summary size (sec);
TT - Total time spent judging the inclusions (sec);
VT - Video play time (vs. pause) to judge the inclusions (sec).

Criterion	**IN**	**RE**	**EA**	**DU**	**XD**	**TT**	**VT**
Baseline 1	0.60	3.33	3.33	66.4	-2.28	110.67	66.67
Baseline 2	0.62	3.67	3.67	64.6	-0.89	109.17	63.83
Median. of 22 teams	0.47	3.67	3.33	59.33	5.23	93.17	59.09
Our result	**0.65**	**4.00**	**3.50**	**42.15**	**15.03**	**87.83**	**45.33**
Our Ranking	3	1	3	5	6	8	5

Table IV shows the evaluation results of TRECVID 2007 rushes summarization task. The two baselines are generated by CMU [9]. The first one evenly selects 1-second video clip for every 25 seconds. The second one performs shot clustering and selects one shot from each cluster. The detailed results of all teams can be found in [24]. As can be seen in Table IV, for the first three criterions: IN, EA, and RE that measure the usability of the summaries, we are ranked 3, 1, and 3 respectively. Considering the contradictions among these criterions (*e.g.*, higher IN usually introduce more redundancy and lower RE), our results are quite encouraging. Meanwhile, our summary duration (DU) and watching time (VT) are ranked 5 from 24 runs. As concluded in the evaluation report [24], only our system is significantly better than the baselines in terms of

EA (easy to understand), IN (inclusion of objects and events), and RE (little redundancy).

The major approaches adopted by other teams include shot clustering [9], [5], [37], video acceleration [7], [9], and highlight keyframe detection [4]. Shot clustering focuses on detecting and removing redundant shots based on different features such as SIFT [37] and color [9]. The problem of this kind of approaches lies in two aspects. First, without considering semantic information, only the visually similar shots are pruned. For instance, given a list of continuous shots describing a single event, the users can predict what happens by watching only one or few shots. However, based on shot clustering, these shots may not be similar to each other according to low-level features and thus more shots are unnecessarily included in the summary than actually needed. Second, two shots that are visually similar may appear in different events. Simply removing any one of them based on shot clustering will lead to incomplete event description. In our approach, the redundancy level is measured not only visually, but also semantically by object and event understanding. Thus, the generated summary can preserve the storyline of the video while keeping elegant. According to the experiments of [9], simply speeding up the video achieves relatively high IN, since human is able to capture most objects and events even when the video is played at rather high speed, *e.g.*, 25 times of the original one. But this approach inevitably includes much redundant and junk materials. In [4], highlight shots are extracted by combining keyframe extraction, face detection and motion estimation. This approach attempts to include the information that is potentially important for users. However, the redundancy problem is not addressed.

2) TRECVID 2008 Evaluation: In the most recent rushes summarization task organized by TRECVID 2008, 43 runs from 32 teams are submitted for evaluation. The summaries' lengths are further limited to 2% of the original videos. The evaluation process is the same as in TRECVID 2007. Among the eight criterions used this year, five (DU, XD, TT, VT, IN) are adopted from TRECVID 2007. In addition, JU and RE are used to measure the degree of junk materials and duplicate video clips in the summaries respectively. Another criterion TE is defined and replaces EA in 2007 to evaluate the enjoyability of the summaries. Table V shows and compares the detailed scores of our approach (VIREO.1) with some other systems.

As can be observed in Table V, it is not easy to select a single metric from the eight given criterions to evaluate the summaries. An ideal summary should include most necessary materials with the least junk and redundant information while making the storytelling smooth and enjoyable in as short time as possible. However, there are always contradictions between these criterions. For instance, a large IN usually introduces more junk and redundant materials, and results in a longer summary. On the other hand, the rhythm of a short summary with a lot of materials is usually unpleasant. To have a more comprehensive comparison among different systems, in Table V, four metrics are derived by combining different criterions to evaluate the summaries from different aspects.

The first metric $S_1 = IN * JU * RE * TE/DU$ combines the different criterions to calculate an overall score for each system. For this metric, our submission (VIREO.1) is ranked 3 among all 43 runs. Only QUT_GP and thu_intel get higher scores than us. By checking the raw scores, our summaries include 50% more useful footages (IN) than both of them.

To evaluate the performance of stock footage localization, we define $S_2 = IN * JU * RE$. A higher value of S_2 means that the system can pick up more useful materials while including least junk and redundant information. We are ranked 1 for this metric. Compared with other systems, we are ranked 4 for IN. At the same time, our summaries include the least junk materials (ranked 1 for JU). This demonstrates the effectiveness of our approach described in Section III for stock footage classification. For RE, our system scores around the average level among all runs. As presented in Section V, we just consider the representability of each clip for its near neighbors so as to smooth the storytelling. Some similar objects and scenes are inevitably included in a long video sequence. This is probably the price we pay for keeping a smooth storyline in the summary. Some other runs (such as JRS, COST292 and REGIM) are good at removing junk and redundant materials, but also exclude more useful footages at the same time.

Two metrics $S_3 = 10 * IN * TE/DU$ and $S_4 = 10 * IN * TE/TT$ are defined to measure the usability of the summary. For the users, a good summary should be able to tell the story completely with enjoyable rhythm in limited duration (S_3) or watching time (S_4). As shown in Table V, we are ranked 1 for both metrics. Our summary length is shorter than most other runs. Meanwhile, we include more useful materials and keep a relatively good rhythm of the summary. This helps the users to judge the usefulness of the rushes videos efficiently and enjoyably. For IN, only the runs from CMU and asahikasei get higher scores than ours. However, all these runs suffer from significantly lower TE which reduces the usabilities of the summaries. Our performances on these two metrics are mainly due to the proposed approach for video summarization in Section V. With the extracted visual and audio features, we detect the object/event occurrences and understand the video content in a more semantic way. This enables us to select the most representative clips for an elegant, complete, and pleasant summary of the video content.

3) Discussions: Since the scales of different criterions are not exactly the same, the above defined metrics might not be perfect. However, as listed in Table V, by these four metrics, we can find some systems with good performance for different aspects. The team QUT_GP [31] attempts to make summaries as enjoyable as possible. Based on shot clustering, the longest shot is selected. The number of faces, the amount of motion and the size of the cluster are used to rank and select shots for summary generation. In the submission from thu-intel [41], color, edge, face detection, motion intensity and audio information are used to select the most representative clips by hierarchical clustering. In PolyU's system [14], unsupervised clustering is also employed for keyframe and clip selection based on local color histogram feature. PicSOM [13] selects video clips by initially favoring the frames near the center of each shot using linear weighting. The scores of clips containing faces, speech, objects or camera motion

TABLE V

OUR RESULTS (VIREO.1) ON TRECVID 2008 RUSHES SUMMARIZATION TASK. THE SYSTEMS ARE RANKED AMONG 43 SUBMITTED RUNS. **JU:** DEGREE OF JUNK FRAMES IN THE SUMMARY (1-5), 5 MEANS LEAST JUNK MATERIALS IN SUMMARIES; **RE:** DEGREE OF DUPLICATE VIDEOS (1-5), 5 MEANS THE LEAST REDUNDANCY IN SUMMARIES; **TE:** DEGREE OF PLEASANT RHYTHM (1-5), 5 FOR THE BEST. $S_1 = IN * JU * RE * TE/DU$; $S_2 = IN * JU * RE$; $S_3 = 10 * IN * TE/DU$; $S_4 = 10 * IN * TE/TT$.

System	TRECVID 2008 Criterions								Derived Metrics							
	DU	XD	TT	VT	IN	JU	RE	TE	S_1	rank1	S_2	rank2	S_3	rank3	S_4	rank4
QUT_GP.1	21.5	7.17	32.67	24.33	0.44	3.67	3.67	3.33	0.918	1	5.926	5	0.681	2	0.448	2
thu-intel.2	19.6	12.32	31.67	21.67	0.42	3.67	3.67	3.00	0.866	2	5.657	8	0.643	3	0.398	5
VIREO.1	23.6	7.63	38.00	25.00	0.67	3.67	3.00	2.67	0.835	3	7.377	1	0.758	1	0.471	1
PolyU.1	26.0	3.07	36.00	27.00	0.47	3.67	3.67	3.33	0.811	4	6.330	2	0.602	4	0.435	3
COST292.1	22.8	8.44	31.00	24.67	0.31	3.67	4.00	3.33	0.665	5	4.551	23	0.453	15	0.333	10
PicSOM.1	22.1	4.05	32.33	25.00	0.44	3.33	3.33	3.00	0.662	6	4.879	15	0.597	5	0.408	4
thu-intel.1	28.1	4.09	39.00	28.67	0.42	3.67	3.67	3.00	0.604	7	5.657	7	0.448	16	0.323	12
JRS.1	18.5	13.38	25.33	20.00	0.22	3.67	4.00	3.33	0.581	8	3.230	38	0.396	23	0.289	19
asahikasei.1	19.5	9.64	34.67	20.00	0.69	3.00	3.00	1.67	0.532	11	6.210	3	0.591	6	0.332	11
REGIM.1	28.0	2.65	36.67	30.67	0.31	3.67	3.67	3.33	0.497	14	4.175	27	0.369	27	0.282	22
BU_FHG.1	22.9	7.94	38.67	24.67	0.58	3.00	3.00	2.00	0.456	19	5.220	10	0.507	7	0.300	17
CMU.2	33.9	0.40	56.67	35.67	0.81	3.00	2.00	1.67	0.239	39	4.860	16	0.399	22	0.239	36
CMU.1	33.9	0.40	53.33	33.00	0.80	3.00	2.00	1.67	0.236	40	4.800	18	0.394	24	0.251	33
cmubase3.1	33.9	0.40	58.67	34.67	0.83	2.33	2.00	1.33	0.152	42	3.868	31	0.326	39	0.188	40

are then increased using heuristic weights. In COST292' system [18], face detection, camera motion and MPEG-7 color layout descriptors for each frame are used as input to their clustering approach for summarization. The system puts emphasis on the enjoyability of the summary by following the storyline and some editing rules, for instance, it never displays segments shorter than 2 seconds. In asahikasei's submission [11], duplicate scenes are removed based on average color of scenes. Each scene is skimmed to keep motion of video constant. In Joanneum's submission [3], visual activity and face detection are employed to select the important clips based on defined rules. In [28], [29], BU_FHG models rushes videos as a hierarchical structure and employs k-NN clustering for redundancy removal. The most representative shot is selected from each cluster according to its length and sum of activity level for summarization. This system is implemented in the compressed domain and quite efficient. Since no semantic information is considered, low TE scores are achieved. CMU contributes three runs [6]. A baseline is first generated by fast forwarding the play of original videos by 50 times. The irrelevant clips are then detected and removed. Audio is added to the summary to improve the comprehensiveness. Fast-forwarding is good at covering most footages. However, as can be found in Table V for the CMU runs, the summaries are still filled with many repetitive shots and the enjoyability is inevitably unsatisfactory.

Overall, in TRECVID 2008 BBC rushes summarization task, shot clustering is widely employed to detect and remove redundant materials. Some low- or mid-level features such as shot length [31], motion intensity [29], [11], spatial image salience [15], and human face occurrence [3], [41], [18], [31] are used to heuristically rank the shot importance. These methods intend to highlight some content such as face and motion, and produce summaries that are good from aspects such as content inclusion, system efficiency and enjoyability. In comparison, our system which relies on object and event understanding offers better capability of selecting clips that maximize content inclusion. This undoubtedly leads to better understanding of summaries. The factors such as excluding undesirable content by stock localization and domain specific knowledge also greatly enhance the summary quality in terms of content conciseness and enjoyability.

For the efficiency issue, the overall complexity of the whole system is basically $O(\mathcal{L})$ where $\mathcal{L}$ is the original video length. The system time is about $5\mathcal{L}$ which varies according to the structure of the rushes videos (*e.g.*, the percentages of stock footages and retakes). The feature extraction stage, especially object and face detection, keypoint matching and tracking, consumes most of the computational time, up to 3 times of the video duration. Stock footage classification can be done in video playing time, while summary generation takes only few minutes. Overall, object and event based video summarization inevitably consumes more time while producing better summaries. According to the evaluation in [28], when system efficiency is considered, our system is still ranked 2, 4, and 13 among all the 43 runs under different evaluation settings.

C. Subjective Evaluation

After having compared our approach with other existing systems in TRECVID BBC rushes summarization task, in this section, we conduct a subjective evaluation for collecting user comments. Besides our approach, we also implement another two algorithms for comparison. The first one is done by simply speeding up the videos by 50 times as in [6]. For the second algorithm, after removing the junk materials, shots are clustered based on color and edge histograms. The longest shot from each cluster is selected. The number of faces and motion intensity are used to rank the shots for summary generation.

For evaluation, 39 videos from TRECVID 2008 BBC testing dataset are used. We invite 24 people and partition them into

6 groups to grade the summaries. Each group is assigned 6 to 7 videos. For each video, the original video is first played. The judges are then asked to watch and grade the summaries generated by different approaches. The order of playing the summaries is random and unknown to the judges. They are asked to score the summaries by answering the following questions:

1) *Can the summary completely describe the content in the original videos? (Completeness)*
2) *Is the summary with the least redundancy? (Elegancy)*
3) *How easy is it to grasp the story in the summary? (Easy to understand)*
4) *How pleasant do you feel to watch the summaries? (Enjoyability)*

The score for each question ranges from 1-10, with 10 meaning the best. The medians of the scores for different approaches are shown in Table VI.

TABLE VI
SUBJECTIVE EVALUATION AND COMPARISON OF SUMMARIES.

Approach	Completeness	Elegancy	Easy to understand	Enjoyability
Sampling	9.00	2.25	5.50	4.50
Shot clustering	6.75	8.25	7.50	5.25
Ours	8.50	8.00	8.75	7.50

The summaries by even sampling are assumed to contain most information in the original videos. From Table VI, we can see that this approach does not score much higher than ours. In term of elegancy of summaries, sampling approach inevitably include most of the repetitive shots and scores the lowest. Most judges comment that our summaries already include most materials that are important to describe the stories completely. Few missing objects/events do not affect their understanding of the storylines. The motion intensity is a useful hint to detect some events and decide the usability of the video clips. However, some events are actually not motion intensive and thus ignored by being assigned a low preference. For elegancy, motion intensity based approach achieves a slightly higher score than ours since we do not remove those similar shots far away from each other in order to smooth the storytelling.

The last two criterions are used to evaluate the usabilities of summaries. For both of them, our approach gets the highest scores. Most judges feel the rhythm of the sampling approach unpleasant. It is not easy to find what exactly happens in the video. The reason lies in two aspects. First, many junk and repetitive shots are included, which break the storytelling from time to time. Second, since the summary is played at 50 times faster than the original video, people have to receive too much visual information at high speed. Furthermore, it is difficult to locate the useful materials when playing the videos at a very high speed. Although it includes most materials, few people would not like to watch videos in such a rhythm for long time.

By mainly focusing on the motion intensive clips, the second approach distorts the rhythm of the storytelling, and thus reduces the usability and enjoyablity of the summaries. Instead, based on object and event understanding, our approach selects the most representative clips and removing those predictable ones to produce summaries. Since the clip selection follows the storyline of the original videos, our summaries proved to be elegant with good rhythm and users can easily capture the desired information.

VII. CONCLUSION

In this paper, we have presented our approaches for the exploitation of rushes videos by automatically stock footage localization and video summarization. In stock footage classification, by taking into account the sequential patterns of the motion features, the proposed two-level hierarchical hidden Markov model (HHMM) is capable of modelling statistical mapping from low-level motion features to high-level semantic concepts. Experimental results show that our approach significantly outperforms other methods such as SVM, FSM and HMM. In video summarization, the extracted visual and audio features are effective to detect the object and event occurrences. The proposed representability measure helps select the most representative video clips for summary generation. The evaluation results in TRECVID BBC summarization task and subjective evaluation show that our summaries are encouraging at preserving the semantic content and storyline of the given video with pleasant rhythm and concise information.

For the future work, the accuracy of stock footage localization can be improved by also considering the visual qualities of video segments and other multi-modality cues in addition to motion features. Our summarization work demonstrates that the analysis of objects and events show promising for video summarization towards semantic understanding. The proposed techniques of object and event understanding can be employed for both edited and home video summarization.

ACKNOWLEDGEMENT

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 119610).

REFERENCES

[1] B. P. Allen, V. A. Petrushin, G. Wei, and D. Roqueiro, “Semantic Web Techniques for Searching and Navigating Video Shots in BBC Rushes”, *NIST TRECVID Workshop*, 2006.

[2] W. Bailer and G. Thallinger, “A Framework for Multimedia Content Abstraction and its Application to Rushes Exploitation”, *ACM Conf. on Image and Video Retrieval*, 2007.

[3] W. Bailer and G. Thallinger, “Comparison of Content Selection Methods for Skimming Rushes Video”, *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.

[4] D. Byrne, P. Kehoe, H. Lee, C. Conaire and A. F. Smeaton, “A User-Centered Approach to Rushes Summarization Via Highlight-Detected keyframes”, *TRECVID Workshop on BBC Rushes Summarization at ACM Multimedia*, 2007.

[5] F. Chen, M. Cooper and J. Adcock, “Video summarization preserving dynamic content”, *TRECVID Workshop on BBC Rushes Summarization at ACM Multimedia*, 2007.

[6] M. G. Christel, A. G. Hauptmann, W. Lin, M. Chen, J. Yang, B. Maher, and R. Baron, “Exploring the Utility of Fast-Forward Surrogates for BBC Rushes”, *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.

[7] E. Dumont and B. Merialdo, “Split-Screen Dynamically Accelerated Video Summaries”, *TRECVID Workshop on BBC Rushes Summarization at ACM Multimedia*, 2007.

[8] A. M. Ferman and A. M. Tekalp, “Two-stage hierarchical video summary extraction to match low-level user browsing preerences”, *IEEE Trans. on Multimedia*, vol. 5, no. 2, pp. 244-256, 2003.

[9] A. G. Hauptmann, Michael G. Christel, W. Lin, Bryan Maher, J. Yang, Robert V. Baron, and G. Xiang, "Clever Clustering vs. Simple Speed-Up for Summarizing BBC Rushes", *TRECVID Workshop on BBC Rushes Summarization at ACM Multimedia*, 2007.
[10] X. S. Hua, T. Mei, W. Lai, M. Wang, J. Tang, G. J. Qi, L. Li, and Z. Gu, "Microsoft Research Asia TRECVID 2006 High-Level Feature Extraction and Rushes Exploitation", *NIST TRECVID Workshop*, 2006.
[11] K. Ishihara and Y. Sakai, "BBC rush summarization and High-Level Feature extraction In TRECVID2008", *NIST TRECVID Workshop*, 2008.
[12] C. Kim and J.-N. Hwang, "Object-Based Video Abstraction for Video Surveillance Systems", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 12, 2002.
[13] M. Koskela, M. Sjoberg, V. Viitaniemi, and J. Laaksonen, "PicSOM Experiments in TRECVID2008", *NIST TRECVID Workshop*, 2008.
[14] Yang Liu, Yan Liu, T. Ren, and K. C. Chan, "Rushes Video Summarization using Audio-Visual Information and Sequence Alignment", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[15] Z. Liu, E. Zavesky, B. Shahraray, D. Gibbon, and A. Basso, "Brief and High-Interest Video Summary Generation: Evaluating the AT&T Labs Rushes Summarization", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[16] Y. F. Ma, L. Lu, H. J. Zhang, and M. Li, *A User Attention Model for Video Summarization*, *ACM Multimedia Conf.*, 2002.
[17] T. Mei, X. S. Hua, C. Z. Zhu, H. Q. Zhou, and S. Li, "Home Video Visual Quality Assessment with Spatiotemporal Factors", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 6, 2007.
[18] S. Naci, U. Damnjanovic, B. Mansencal, J. B. Pineau, C. Kaes, M. Corvaglia, E. Rossi, and N. Aginako, "The COST292 Experimental Framework for Rushes Summarization Task in TRECVID 2008", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[19] J. Nam and A. H. Tewfik, "Dynamic Video Summarization and Visualization", *ACM Multimedia Conf.*, 1999.
[20] C. W. Ngo, T. C. Pong and Roland T. Chin, "Video partitioning by temporal slice coherency", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 8, 2001.
[21] C. W. Ngo, Z. Pan, and X. Y. Wei, "Hierarchical Hidden Markov Model for Rushes Structuring and Indexing", *ACM Int. Conf. on Image and Video Retrieval*, 2006.
[22] C. W. Ngo, W. L. Zhao, and Y. G. Jiang, "Fast Tracking of Near-Duplicate Keyframes in Broadcast Domain with Transitivity Propagation", *ACM Multimedia Conf.*, 2006.
[23] C. W. Ngo, Z. Pan, X. Wei, X. Wu, H. K. Tan, and W. Zhao, "Motion Driven Approaches to Shot Boundary Detection, Low-Level Feature Extraction and BBC Rushes Characterization at TRECVID 2005", *TRECVID Workshop*, 2005.
[24] Paul Over, Alan F. Smeaton, and Philip Kelly, "The TRECVID 2007 BBC Rushes Summarization Evaluation Pilot", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2007.
[25] Paul Over, Alan F. Smeaton, and G. Awad, "The TRECVID 2008 BBC Rushes Summarization Evaluation", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[26] Z. Pan and C. W Ngo, "Structuring home video by snippet detection and pattern parsing", *ACM SIGMM International workshop on Multimedia Information Retrieval*, 2004.
[27] Z. Pan and C. W. Ngo, "Moving Object Detection, Association and Selection in Home Videos", *IEEE Trans. on Multimedia*, vol. 9, no. 2, 2007.
[28] J. Ren and J. Jiang, "Hierarchical Modeling and Adaptive Clustering for Real-Time Summarization of Rushes Videos", *IEEE Trans. on Multimedia*, vol. 11, no. 5, 2009.
[29] J. Ren and J. Jiang, "Hierarchical Modeling and Adaptive Clustering for Real-Time Summarization of Rushes Videos in TRECVID'08", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[30] Peter J. Rousseeuw and Annick M. Leroy, "Robust regression and outlier detection", *Wiley New York*, 1987.
[31] J. Sasongko, C. Rohr, and D. Tjondronegoro, "Efficient Generation of Pleasant Video Summaries", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[32] L. Tang, T. Mei, and X. S. Hua, "Near-lossless video summarization", *ACM Multimedia*, 2009.
[33] S. Tang, Y. Zhang, J. Li, X. Pan, T. Xia, and M. Li, "Rushes Exploitation 2006 By CAS MCG", *NIST TRECVID Workshop*, 2006.
[34] C. M. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E. Delp, "Automated Video Program Summarization Using Speech Transcripts", *IEEE Trans. on Multimedia*, vol. 8, no. 4, pp. 775-791, 2006.
[35] B. T. Truong and S. Venkatesh, "Video Abstraction: A Systematic Review and Classification", *ACM Trans. on Multimedia Computing, Communications and Applications*, vol. 3, no. 1, 2006.
[36] B. T. Truong and S. Venkatesh, "Curtin at Trecvid 2006 - Rushes Summarization", *NIST TRECVID Workshop*, 2006.
[37] B. T. Truong and S. Venkatesh, "Generating Comprehensible Summaries of Rushes Sequences based on Robust Feature Matching", *TRECVID Workshop on Rushes Summarization at ACM Mltimedia*, 2007.
[38] F. Wang and C. W. Ngo, "Rushes Video Summarization by Object and Event Understanding", *TRECVID Workshop on Rushes Summarization at ACM Mltimedia*, 2007.
[39] M. Wang, X. S. Hua, R. Hong, J. Tang, G. J. Qi, and Y. Song, "Unified Video Annotation via Multigraph Learning", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 5, 2009.
[40] M. Wang, X. S. Hua, J. Tang, and R. Hong, "Beyond Distance Measurement: Constructing Neighborhood Similarity for Video Annotation", *IEEE Trans. on Multimedia*, vol. 11, no. 3, 2009.
[41] T. Wang, S. Feng, P. P. Wang, W. Hu, S. Zhang, W. Zhang, Y. Du, J. Li, J. Li, and Y. Zhang, "THU-Intel at Rushes Summarization of TRECVID 2008", *TRECVID Workshop on BBC Rushes Summarization in ACM Multimedia*, 2008.
[42] L. Xie, S. Chang, A. Divakaran, and H. Sun, "Learning Hierarchical Hidden Markov Models for Video Structure Discovery", *Columbia University Technical Report*, 2002.
[43] J. You, G. Liu, L. Sun, and H. Li, "A Multiple Visual Models Based Perceptive Analysis Framework for Multilevel Video Summarization", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 3, 2007.
[44] W. L. Zhao, C. W. Ngo, H. K. Tan, and X. Wu, "Near-Duplicate Keyframe Identification with Interest Point Matching and Pattern Learning", *IEEE Trans. on Multimedia*, vol. 9, pp. 1037-1048, 2007.
[45] Face Detection Project at CMU, "http://vasc.ri.cmu.edu/NNFaceDetector/".
[46] TRECVID Workshop, "http://www-nlpir.nist.gov/projects/trecvid".

Feng Wang received his PhD in Computer Science from the Hong Kong University of Science and Technology in 2007 and BSc from Fudan University, China, in 2001. Before joining East China Normal University as an associate professor in the Dept. of Computer Science and Technology, he was a research fellow in City University of Hong Kong and Institute Eurecom, France. His research interests include multimedia computing, pattern recognition and IT in education.

Chong-Wah Ngo (M02) received his Ph.D in Computer Science from the Hong Kong University of Science & Technology in 2000. He received his MSc and BSc, both in Computer Engineering, from Nanyang Technological University of Singapore. He is currently an Associate Professor in City University of Hong Kong. He was with Beckman Institute of University of Illinois in Urbana-Champaign as post-doctoral researcher, and with Microsoft Research Asia as visiting researcher. He is the program co-chair of ACM Multimedia Modeling (MMM) 2012 and International Conference on Multimedia Retrieval (ICMR) 2012. His research interests include video computing and multimedia information retrieval.