Hierarchical Hidden Markov Model for Rushes Structuring and Indexing

Chong-Wah Ngo, Zailiang Pan and Xiaoyong Wei

Department of Computer Science,
City University of Hong Kong, Kowloon, Hong Kong
{cwngo,zerin,xiaoyong}@cs.cityu.edu.hk

Abstract. Rushes footage are considered as cheap gold mine with the potential for reuse in broadcasting and filmmaking industries. However, it is difficult to mine the "gold" from the rushes since usually only minimum metadata is available. This paper focuses on the structuring and indexing of the rushes to facilitate mining and retrieval of "gold". We present a new approach for rushes structuring and indexing based on motion feature. We model the problem by a two-level Hierarchical Hidden Markov Model (HHMM). The HHMM, on one hand, represents the semantic concepts in its higher level to provide simultaneous structuring and indexing, on the other hand, models the motion feature distributions in its lower level to support the encoding of the semantic concepts. The encouraging experimental results on TRECVID$'$05 BBC rushes demonstrate the effectiveness of our approach.

1 Introduction

In the broadcasting and filmmaking industries, *rushes* is a term for raw footage, which is used to generate the final productions such as TV programs and movies. Only a small portion of the rushes is actually used in the final productions. The "shoot-to-show" ratio, such as in BBC TV, ranges from 20 to 40. The producers see these large amount of raw footage as cheap gold mine. The "gold" refers to *stock* footage which is the "generic" clips with high potentials for reuse. However cataloguing the stock footage is a tedious task, since rushes is unstructured and relatively inaccessible with only a minimum metadata such as program/department name and date. Therefore, it becomes necessary to develop techniques for the structuring, indexing and retrieval of rushes.

In the past decades, researches on video representation and analysis are mainly founded on edited videos, e.g., news, sports and movies. The edited videos are highly structured. More importantly, multiple modalities such as textual, auditory and visual modalities are available for analysis in edited videos. The performance of most state-of-the-art video retrieval systems (e.g, Informedia project [1]) depends on the fusion of these modalities, especially the textual information which mainly comes from the captions and speech transcripts by OCR and ASR respectively. In contrast to edited videos, rushes are characterized by unstructured, natural sounds only and few or no on-screen texts. Thus little textual information can be acquired from rushes. These characteristics present a new aspect of research challenge for rushes retrieval.

In TRECVID′05 [2], several rushes retrieval techniques have been presented in the pilot task of BBC rushes exploration. All of them are mainly based on visual information as other modalities are absent or difficult to obtain. Allen and Petrushin [3] indexed the rushes shots by "visual words" which are the cluster centroid of color, texture and color+texture. The approach proposed by Foley et al [4] considered the color and texture features extracted from each keyframe, as well as the color, texture and shape of the semi-automatically segmented objects. The system allows the user to select features from either frame or object for retrieval. Snoek et al [5] re-attempted their MediaMill system, which is originally trained to index the 101 high-level concepts for news, to analyze the BBC rushes. These approaches basically port the retrieval system developed originally for edited videos directly to the rushes domain. Some issues peculiar to in rushes are not addressed, such as how to detect and manage the redundant footage due to low visual quality or unwanted motion. Ngo et al [6] attempted to solve this problem from the motion point of view. They tested three different approaches, Finite State Machine (FSM), Support Vector Machine (SVM) and Hidden Markov Model (HMM) for the characterization of BBC Rushes.

The main focus of this paper is to locate and index the stock footage. We consider three semantic categories: *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 zoom to get more details and a pan to change to another perspective. Beside those two extreme cases, we add another category, *shaky*, to represent the shaky artifacts which may be discarded or used for special effects such as to show a emergent situation. For rushes indexing, shot is usually regarded as the basic unit [3–5]. However, since the rushes are raw footage without editing, a shot may consist of different semantic concepts. Thus, in order to index rushes effectively, the temporal structure of rushes needs to be carefully analyzed so that the basic unit can be a subshot, and each subshot contains only one semantic label.

The problem of structuring and indexing is intertwined in rushes. The difficulty comes from the following two aspects. One aspect is that the motion features of the three semantic categories are highly overlapped. For example, a pan motion may come from a *stock* of side view on a moving vehicle, or a *outtake* of perspective change, or a *shaky* of one part of swing. The other aspect is that structuring and indexing rely on each other. Indexing requires the investigation of the temporal structure. As mentioned, an unstructured shot in rushes may be too long to be an appropriate unit for indexing. On the other hand, it is also infeasible to structure the videos without knowing the underlying characteristic of frames. For example, structuring only by motion cannot obtain satisfied performance due to the indiscriminate motion features of the three semantic concepts.

In this paper, we propose a new approach for structuring and indexing the rushes footage by Hierarchical Hidden Markov Model (HHMM) [7–9]. HHMM is the generalization of HMM with hierarchical structure. We use a two-level HHMM to encode the three semantic categories and model the sequential changes of their underlying motion features. Higher-level substates represent the semantic concepts, *stock*, *outtake* and *shaky*. Each of these substates is a sub-HMM that has its own substates in the lower level to describe the distribution of the motion features and their transitions. This hierarchical model, on one hand, can alleviate the feature overlap problem by taking into account the temporal constraint. For instance, an *outtake* pan can be distinguished from a *shaky* pan by considering that a *shaky* pan is very likely to have reverse motions before and after. On the other hand, the higher-level substates, which address the semantic concepts, make it possible to simultaneously structure and index the rushes on the whole sequence. The simultaneous decision on the whole sequence provides a way to decouple the interdependency between structuring and indexing.

The remaining of this paper is organized as follows. Section 2 describes our approach for rushes structuring and indexing. We first extract the motion features as the observation for each subshot. A two-level HHMM is then applied to model both the high-level concepts and the low-level motion features to structure and index the rushes. Section 3 presents the experiment results. Finally, Section 4 concludes the paper and discusses future work.

2 Rushes Structuring and Indexing by Hierarchical HMM

Figure 1 illustrates our two-level HHMM structure. On the top is the root state which is an auxiliary substate to make the structure representable by a single tree. 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 feature to support the encoding of semantic concepts at the higher level. For each semantic concepts, we use six substates, *left*, *right*, *up*, *down*, *in* and *out*, to model the six major movements respectively in horizontal, vertical and depth directions. Notice that the substates in each sub-HMM are fully connected. For the simplicity of presenting the figure, we do not show the edges in Figure 1.

In order to facilitate structuring and indexing, a shot in rushes should be partitioned into shorter unit, i.e. subshots. In practice, the subshot string of a shot forms an observation sequence for HHMM. The subshot should not be too long so as not to mix different semantic concepts. Meanwhile, it should not be too short in order to extract robust motion feature. In this paper, we investigate two kinds of subshot: *fixed* and *adaptive*. The former one is obtained through equal partitioning of a shot into segments of fixed length. Adaptive subshots, on the

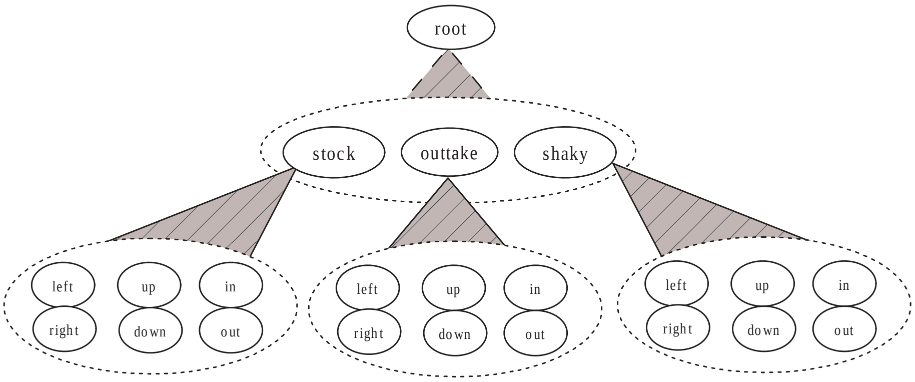

Fig. 1. An illustration of an HHMM of two levels. Solid ellipses denote the substates, while dotted ellipses denote the sub-HMMs of the HHMM structure.

other hand, are obtained by dividing a shot into segments each with consistent motion [10]. Both types of subshots have their strength and weakness. The fixed subshot is easy to obtain in practice, but with inaccurate subshot boundary and motion feature. Intuitively, adaptive subshot may have better performance due to good boundary and motion feature. However, since subshot segmentation by motion itself is a research issue, false and missed detections would introduce under or over segmented subshots that prohibit the finding of underlying semantic labels. For both fixed and adaptive schemes, over-segmentation at early stage can be remedied by HHMM if two adjacent segments have the same semantic labels. Under-segmentation, however, cannot be dealt with by HHMM.

2.1 HHMM Representation

A state in an HHMM actually consists of a string of substates from top to bottom levels. To denote the substate string from top to level d, we use a bar notation,

$$k^d = q_{1:d} = \overline{q_1 q_2 \cdots q_d}, \tag{1}$$

where the subscripts denote the hierarchical levels. We drop the superscript d for abbreviation when there is no confusion. Let D denotes the maximum number of levels and Q denotes the maximum size of any sub-HMM state spaces in HHMM. Then a HHMM can be specified by the following parameters,

$$\Theta = \{\mathcal{A}, \mathcal{B}, \Pi, \mathcal{E}\}. \tag{2}$$

Explicitly, $\mathcal{A}$ denotes the transition probabilities $(\bigcup\limits_{d=1}^{D} \bigcup\limits_{k=1}^{Q^{d-1}} \{a_k^d\})$, where a_k^d is the transition matrix at level d with configuration k^{d-1}. $\mathcal{B}$ is 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 denotes 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.

2.2 Motion Feature Extraction

To obtain the observation sequence for HHMM, we extract three dominant motions, pan/track, tilt/boom and zoom/dolly from each subshot. The inter-frame motion features are firstly estimated from each two adjacent frames. We apply Harris corner detector to extract the image feature points, $\mathbf{x}_t$, 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 [10]. Those matched point pairs in each frame pair are assumed to comply with a single camera motion model. Since the dominant features for rushes structuring and indexing is 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 [11]. RANSAC is not used due to the requirement of inlier threshold which is not easy to set. 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 vector, $\{f\}$, is then obtained from the frame sequence in a subshot. To suppress the outliers, we use the median, instead of average, motion vector as the observation for a subshot, that is,

$$o = \text{median}\{f\}. \tag{3}$$

Then a T-subshot string of a shot forms an observation sequence for HHMM, denoted as $O = (o_1, o_2 \cdots o_n \cdots o_T)$.

2.3 HHMM Parameter Learning by EM Algorithm

Given an observation sequence $O = (o_1, o_2 \cdots o_t \cdots o_T)$, the task of parameter learning is to find Θ^* that maximize the likelihood $L(\Theta)$. This is estimated by the Expectation-Maximization (EM) algorithm as in traditional HMM. Given an old parameter $\mathit{\Theta}$ 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 $\hat{\mathit{\Theta}}$ is written by

$$\begin{aligned} L(\hat{\mathit{\Theta}}, \mathit{\Theta}) &= E(\log p(O, K|\hat{\mathit{\Theta}})|O, \mathit{\Theta}) && (4)\\ &= \sum_K p(K|O, \mathit{\Theta}) \log p(O, K|\hat{\mathit{\Theta}}) && (5)\\ &\propto \sum_K p(O, K|\mathit{\Theta}) \log p(O, K|\hat{\mathit{\Theta}}) && (6)\end{aligned}$$

The E-step estimates the expectation $L(\hat{\mathit{\Theta}}, \mathit{\Theta})$, and the M-step finds the value $\hat{\mathit{\Theta}}$ 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 happens at level d, given O and $\mathit{\Theta}$, as

$$\xi_t(k, k', d) \overset{def}{=} p(k_t = k, k_{t+1} = k', e_t^{1:d} = 0, e_t^{d+1:D} = 1|O, \mathit{\Theta}). \tag{7}$$

Similarly, we define the probability of being in state k at time t, given O and $\mathit{\Theta}$, as follows

$$\gamma_t(k) \overset{def}{=} p(k_t = k|O, \mathit{\Theta}). \tag{8}$$

In E-step, these two auxiliary variables are estimated by forward and backward algorithm [8]. Then by marginalizing and normalizing the auxiliary variables ξ and γ in M-step, we can get the updated model parameter $\hat{L}$ as follows,

$$\hat{\pi}_q^d(i) = \frac{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \sum\limits_{q''} \xi_t(q', \overline{qiq''}, d-1)}{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \sum\limits_{q''} \sum\limits_{i} \xi_t(q', \overline{qiq''}, d-1)} \tag{9}$$

$$\hat{e}_q^d(i) = \frac{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \sum\limits_{k'} \sum\limits_{d'<d} \xi_t(\overline{qiq'}, k', d')}{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \gamma_t(\overline{qiq'})} \tag{10}$$

$$\hat{a}_q^d(i, j) = \frac{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \sum\limits_{q''} \xi_t(\overline{qiq'}, \overline{qjq''}, d)}{\sum\limits_{t=1}^{T-1} \sum\limits_{q'} \sum\limits_{q''} \sum\limits_{j} \xi_t(\overline{qiq'}, \overline{qjq''}, d)} \tag{11}$$

$$\hat{\mu}_k = \frac{\sum\limits_{t=1}^{T} o_t \gamma_t(k)}{\sum\limits_{t=1}^{T} \gamma_t(k)} \tag{12}$$

$$\hat{\Sigma}_k = \frac{\sum_{t=1}^{T}(o_t - \mu_k)(o_t - \mu_k)^T \gamma_t(k)}{\sum_{t=1}^{T}\gamma_t(k)} \tag{13}$$

2.4 Structuring and Indexing by Viterbi Algorithm

Our final goal is to structure and index rushes with the three semantic categories. Instead of deciding the subshots once at a time, such as using SVM, HHMM can perform simultaneous structuring and indexing upon the whole subshot string. Given an observation sequence of a shot, $O = (o_1, o_2 \cdots o_t \cdots, o_T)$, we apply Viterbi algorithm [8] 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 the higher-level variable string, $K^{1*} = (k_1^{1*}, k_2^{1*} \cdots, k_t^{1*}, \cdots, k_T^{1*})$, which forms the indices of the semantic concepts for a shot. Meanwhile, the variations in the variable string K^{1*} indicate the locations of the semantic concept boundary. Therefore, by using Viterbi algorithm on the subshot string, the simultaneous structuring and indexing for a rushes shot can be efficiently achieved.

3 Experiments and Results

We randomly select 60 videos (about 400k frames or 4.5 hours) from BBC rushes of TRECVID′05 corpus to evaluate our approach. The videos are manually structured and indexed with the semantic labels: *stock*, *outtake* and *shaky*. We divide the videos equally into two set: 30 videos for training and 30 videos for testing.

We partition each video into shots by [12] and each shot is further decomposed into subshots. For *fixed* subshot, we empirically set the fixed duration to one second. For adaptive subshot, we use the motion-based finite state machine [10] to partition shot into subshots. Each subshot, both fixed and adaptive, is labeled with the three categories based on the ground truth manually marked by human subjects. The two-level HHMM are then trained with EM algorithm. Since only the higher-level labels are available, this is a mixed learning procedure. In other words, the learning at higher level is supervised, while the learning at lower level is unsupervised. For abbreviation, we name the HHMM of fixed and adaptive subshots as F-HHMM and A-HHMM respectively.

We compared the proposed HHMM with our previous work presented in TRECVID′05 [6]. In [6], we experimented three approaches: Finite State Machine (FSM), Hidden Markov Model (HMM) and Support Vector Machine (SVM). Table 1 summaries and contrasts the properties of different approaches. FSM and HMM models are flattened HHMM with only the higher level. 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. Therefore, from the structure's point of view, HHMM has a two-level hierarchical structure, while the others are flattened. We use Radial Basis Function (RBF) as the kernel for SVM. Meanwhile, Gaussian distribution is used as kernel function in HMM and HHMM. Inside the FSM states, thresholding is used to determine which category an observation belongs to. We applied adaptive subshot detection for FSM and A-HHMM, while using fixed subshot for the others. Notice that we use nine-dimensional feature for HMM and SVM. Besides the three motion features (Section 2.2), we also use the motion variations as additional features in order to improve the discriminative power of a flattened structure. The details can be found in [6].

	Subshot	#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 1. Comparison of different method's properties.

3.1 Rushes Indexing

Table 2 and Table 3 show the indexing results of the training and testing videos respectively. The results are evaluated based on the number of frames being correctly or wrongly classified. The results show that HHMM outperforms the other approaches. Overall, we have about 96% accuracy on *stock*, 40% on *outtake* and 60% on *shaky* in the testing set. The results of SVM indicate that the feature distributions of the three semantic concepts severely overlapped among each other. Thus even the classification accuracy on training set is pretty low. SVM assumes that the observations are independent and neglect the temporal relationship between subshots. For example, a *shaky* tilt is more likely to be followed by a reverse tilt in the same *shaky* segment rather than an *outtake* tilt. By exploiting the temporal relationship, HHMM presents some improvement compared to SVM. Through experiments, hierarchical HMM shows better performance than flat HMM, particularly the accuracy of *shaky* is significantly improved. The reason perhaps lies in the fact that the movement of shaky artifacts usually has patterns such as swinging between left and right. The temporal relationship of the *shaky* can be captured by HHMM as a unique sequential pattern for recognition. The improvement of *outtake*, nevertheless, is less obvious than *shaky.* An *outtake* is usually a single movement, such as a zoom to get details or a pan to get another side of the scene. Thus the amount of sequential information to be captured by HHMM is limited. The sequential pattern of a *shaky* can be more distinctive if the turning points of motion are correctly located. This is why A-HHMM has better *shaky* accuracy than F-HHMM since the adaptively segmented subshots have more expressive power in describing the temporal structure of the *shaky* segments.

	Stock		Outtake		Shaky	
	Recall	Prec.	Recall	Prec.	Recall	Prec.
FSM	0.815	0.981	0.802	0.118	0.011	0.050
SVM	0.827	0.990	0.701	0.162	0.715	0.239
HMM	0.927	0.970	0.329	0.137	0.311	0.339
F-HHMM	**0.977**	**0.980**	**0.602**	**0.512**	**0.440**	**0.497**
A-HHMM	**0.976**	**0.983**	**0.648**	**0.551**	**0.546**	**0.515**

Table 2. The indexing accuracy on the training video set.

	Stock		Outtake		Shaky	
	Recall	Prec.	Recall	Prec.	Recall	Prec.
FSM	0.756	0.968	0.844	0.128	0.000	0.000
SVM	0.778	0.975	0.456	0.120	0.362	0.182
HMM	0.909	0.929	0.375	0.196	0.043	0.067
F-HHMM	**0.959**	**0.953**	**0.489**	**0.342**	**0.328**	**0.523**
A-HHMM	**0.962**	**0.963**	**0.408**	**0.427**	**0.624**	**0.597**

Table 3. The indexing accuracy on the testing video set.

3.2 Rushes Structuring

The results of structuring are basically assessed based on the accuracy of the subshot boundaries between the three categories. However, compared to shot boundaries, the subshot boundaries are fuzzy and the exact locations (in term of frame) are not easy to identify even with careful human inspection. In the experiments, a subshot boundary is counted as correct as long as we can find a matched boundary in the ground-truth within 1-second time frame. In our ground-truth, there are 83.4% of boundaries for transitions between *stock* and *outtake*, 16.3% between *stock* and *shaky* and 0.3% between *shaky* and *outtake*. Table 4 shows the structuring results in both training and testing set. From the table, we can find that HHMM has the best results with about 70% accuracy in training set and 60% in testing set. Compared with other approaches, the precision has more obvious improvement than the recall. This shows that by considering the hierarchal relationship among the features, we can remarkably remove the false alarms while retaining the correct boundaries.

	Training		Testing	
	Recall	Prec.	Recall	Prec.
FSM	0.614	0.282	0.593	0.279
SVM	0.769	0.281	0.763	0.289
HMM	0.461	0.419	0.395	0.379
F-HHMM	**0.615**	**0.712**	**0.610**	**0.605**
A-HHMM	**0.707**	**0.725**	**0.582**	**0.611**

Table 4. The structuring accuracy in both training and testing video set.

4 Conclusion and Future Work

In this paper, we have presented a novel approach for rushes structuring and indexing, which is one key component to mine the "gold" in rushes for film producers. By taking into account the sequential patterns of the motion features, the proposed two-level hierarchical hidden Markov model is capable of modelling statistical mapping from low-level motion features to high-level semantic concepts: *stock*, *outtake* and *shaky*. Experimental results show that our approach significantly outperforms other methods based on SVM, FSM and HMM. Currently, we only utilize motion features. Other indicators such as the visual qualities of a film and the cues derived from multi-modal features can be incorporated to further improve the accuracy of locating stock footage.

Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 118905), and a grant from City University of Hong Kong (Project No. 7001804).

References

1. Hauptmann, A.: Lessons for the future from a decade of informedia video analysis research. In: CIVR'05, International Conference on Image and Video Retrieval. (2005)
2. TRECVID: (http://www-nlpir.nist.gov/projects/trecvid)
3. Allen, B.P., Petrushin, V.A.: Searching for relevent video shots in bbc rushes using semantic web techniques. In: Proceedings of the TRECVID Workshops. (2005)
4. Foley, C., et al.: Trecvid 2005 experiments at Dublin City University. In: Proceedings of the TRECVID Workshops. (2005)
5. Snoek, C.G.M., et al.: The mediamill trecvid 2005 semantic video search engine. In: Proceedings of the TRECVID Workshops. (2005)
6. Ngo, C.W., et al.: Motion driven approaches to shot boundary detection, low-level feature extraction and bbc rush characterization. In: Proceedings of the TRECVID Workshops. (2005)
7. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden Markov model: Analysis and applications. Machine Learning **32**(1) (1998) 41–62
8. Xie, L., et al.: Learning hierarchical hidden Markov models for video structure discovery. Technical report, Columbia University (2002)
9. Murphy, K., Paskin, M.: Linear time inference in hierarchical HMMs. In: Proceedings of Neural Information Processing Systems. (2001)
10. Pan, Z., Ngo, C.W.: Structuring home video by snippet detection and pattern parsing. In: MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, ACM Press (2004) 69–76
11. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection. Wiley New York (1987)
12. Ngo, C.W., Pong, T.C., Chin, R.T.: Video partitioning by temporal slice coherency. IEEE Trans. Circuits Syst. Video Technol. **11**(8) (2001) 941– 953