Statistical Framework for Optimal Segmentation of Video

Viachaslau Parshyn, Liming Chen

A Research Report, Lab. LIRIS, Ecole Centrale de Lyon

LYON 2006

Abstract. Automatic segmentation of videos is widely used for structuring and is a necessary preliminary step for many applications. In this report we propose an optimal strategy for the temporal segmentation issue on the basis of statistical modeling. The goal is to maximize the performance metrics of recall and precision directly, which is in contrast to the conventional approaches where the output segments are chosen based rather on intuition, by recovering, for instance, the most probable path through the model space. Application of optimal strategy requires the proper choice of model on the video of interest. Based on the theory of hidden Markov models and their extensions, we consider a video as a stochastic automaton – statistical generalization of the finite state machine. This enables us to take into account the correlation between semantic segments at different levels of abstraction (for hierarchical models) and the non-uniform distribution of segment duration. The resulting segmentation technique is considered as opposed to the conventional Viterbi algorithm.

1 Introduction

In practical applications it is difficult to find features or keys which would enable unambiguous segmentation of video. The ambiguity can be caused by the unreliability of the key detection or by the absence of the direct dependency. In the conventional deterministic approach this uncertainty is often ignored or is taken into account very roughly at the expense of the significant growth of system complexity. In this report we propose a statistical approach, enabling the keys to be treated in a probabilistic manner. This allows one to take into account “soft” grammar constraints imposed on the semantic structure and expressed in the form of probability distributions. Moreover, the multiple keys, being considered as statistical variables, can be more easily fused into one, more reliable decision in the case of their collisions. Based on the theory of hidden Markov models and their extensions, we consider a video as a stochastic automaton – statistical generalization of the finite state machine. This enables us to take into account the correlation between semantic segments at different levels of abstraction (for hierarchical models) and the non-uniform distribution of segment duration.

Further in this report we first consider the general principles how to chose the optimal segments based on the corresponding probability estimates. In contrast to the conventional approach which chooses the single best path for the state variables, we focus on the state transitions so as to find the optimal segmentation in terms of recall and precision. Then we consider the video segmentation task based, more specifically, on a hidden Markov model and its extensions.

2 Segmentation Principles

2.1 Optimality Criterion

We consider video segmentation as detection of segment boundaries at discrete time moments given an input set of features extracted from raw video. These time moments or candidate points of segment boundaries can be chosen in various ways. In the tasks considered in this report they are camera shot boundaries since the semantic segments of interest are defined as groups of shots. Alternatively the candidate points might be determined by the boundaries of mid-level events or simply chosen at discrete times regularly spaced with an interval providing acceptable temporal resolution.

To indicate the absence or presence of a segment boundary at time index t we use a binary variable $s_t \in \{0,1\}$. So, the aim of segmentation of a video is to find an optimal sequence $s \equiv \{s_1, s_2, ..., s_T\}$, where T is the number of candidate points within the video. If segments differ by their semantic meaning, we should also provide semantic labels $\{p_t, f_t\}$ of contiguous segments adjacent to each segment boundary at time *t*, where *p* is the type of the preceding segment and *f* – the type of the following one. Let's denote the sequence of *N* time indexes corresponding to scene boundaries as $b \equiv \{b_1, b_2, ..., b_N\}$. As each segment must have the same semantic label at the ends, the following constraints are imposed:

$$f_{b_i} = p_{b_{i+1}}, \forall i = 1,2,...,N-1. \quad (1)$$

In the general case of hierarchical content structure semantic segments are identified by their type defined at the current semantic level and by the type of the corresponding higher-level segments. We suppose in this case that all these nested identifiers for each segment are enumerated into one label.

To deal properly with the uncertainty of real observable data, we consider the segmentation task in a probabilistic manner by modeling the video as a stochastic process. The task is, then, to find optimal values of random variables s_t at each time index t as well as the corresponding segment labels given a set of observable data generated according to a probabilistic law. But what criterion of optimality should be used? The common approach is to find the most probable sequence of appropriate state variables related to an input video. In boundary-based segmentation methods, when semantic labels of segments are not of interest, these variables are our binary indicators of segment boundaries *s*, as it is the case for story segmentation in [HSU 03, HSU 04]. Alternatively, in segment-based segmentation methods, the temporal dynamics within segments are modeled by a sequence of states, often using hidden Markov models (HMM). For example, TV news broadcasts are segmented into story units in

[LEK 02] using a four-states ergodic HMM; in [EIC 99] logical units of news programs are segmented and classified into six main types where each unit type is represented with a HMM. The most probable sequence of states is computed using computationally effective procedures based on dynamic programming, such as a Viterbi algorithm.

Let's consider this approach from the perspective of the measures used to numerically evaluate the segmentation performance. Recall and precision frequently serve as such measures. They are widespread in information retrieval [RIJ 79, LEW 91] and are standard in story segmentation [GUI 04]. The performance measures are obtained by comparing the actual and claimed segment boundaries of the same video. This is illustrated in Figure 1 where the chain of actual segments is represented by the upper stripe and that of claimed ones – by the lower; different segment types are encoded by different color. An actual boundary is defined to be detected if there is at least one claimed boundary which lies in the vicinity measured by a temporal ambiguity τ and both the boundaries separate the segments of the same type. Otherwise the actual boundary is defined as a miss. Similarly a claimed boundary is defined as a correct one if there is at least one actual boundary within the limits of the ambiguity τ (which is assumed to be the same as for actual boundaries) and both the boundaries separate the segments of the same type. Otherwise the claimed boundary is defined as a false alarm. In fact, an ambiguity window 2τ (see Figure 1) is considered around each actual boundary – if one or several claimed boundaries, which separate the same segments, fall into this window, the corresponding boundaries are defined to be detected and correct (similarly we could place the ambiguity window around each claimed boundary, as the ambiguity time is the same for the actual and claimed segments). If the time interval between two consecutive claimed boundaries is less than 2τ, then it is possible that they are both correct and correspond to the same actual boundary and vice versa. Therefore the number of correct and detected boundaries is not generally the same.

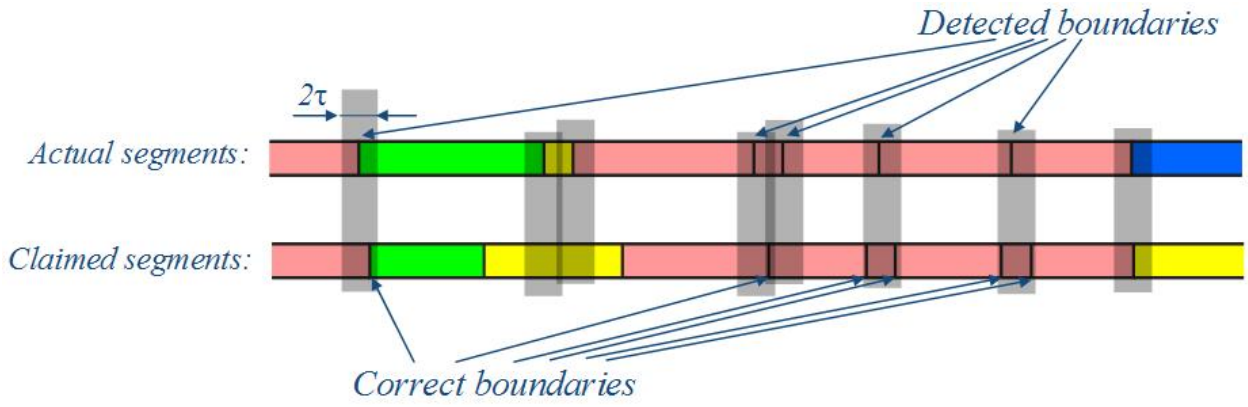

Figure 1. Comparison of segment boundaries.

Recall and precision measure the proportion of actual segment boundaries detected and the proportion of correct claimed segment boundaries respectively. Denoting the number of actual boundaries detected as $N_{a.d.}$, the number of correct clamed boundaries – as $N_{c.c.}$, the number of false alarms – as $N_{f.a.}$, the number of misses – as N_m, recall r and precision p are written as:

$$r = \frac{N_{a.d.}}{N_{a.d.} + N_m}, \tag{2}$$

$$p = \frac{N_{c.c.}}{N_{c.c.} + N_{f.a.}}. \tag{3}$$

System performance measured by recall and precision focuses on time indexes corresponding to segment boundaries. Thus there is no need to take into account all the candidate points at the same time, like in the methods where the most probable sequence of states is found for the whole video. Moreover, in the most cases the moments of absence of segment boundaries are predominant, and the minor points of segment boundaries become negligible when optimizing the whole state sequence. This can deteriorate considerably the segmentation performance. Consider, for example, the situation where a segment boundary can be surely detected in a time range covering several candidate points, but the probability to find this boundary at each single point is quite low. Segmentation through finding the most probable state path for the whole video is likely to ignore the boundary, resulting in increase of number of misses and, hence, low recall.

In this report we derive the optimal decision rule for the segment boundary detection based on recall and precision which are chosen to measure the system performance. Let's suppose that a fixed number N of distinct candidate points are claimed as segment boundaries and the total number of actual boundaries is N_a. It is not difficult to see that the denominator in expression (2) and (3) is equal to N_a and N respectively. Hence, in order to maximize recall and precision, N claimed boundary should be selected so that to provide the maximum values for $N_{a.d.}$ and $N_{c.c.}$. This minimizes the number of false alarms and the number of misses written as

$$N_{f.a.} = N - N_{c.c.}, \tag{4}$$

$$N_m = N_a - N_{a.d.}. \tag{5}$$

Let's further assume that segments cannot be of zero duration and that the coincidence between a claimed boundary and an actual one (allowing us to consider the claimed boundary to be correct and the actual one to be detected) is established only in the case where these boundaries occur exactly at one time (i.e. the time ambiguity τ is zero). Under these assumptions each correct claimed boundary correspond to one and only one actual boundary detected and, hence,

$$N_{a.d.} = N_{c.c.} \tag{6}$$

which is the only value to be maximized.

Let's now derive an expression for $N_{c.c}$. To distinguish the claimed (computed) segment boundaries the actual ones, we use a tilde. Thus, the result of computed segmentation for an input video is denoted as a sequence of tuples $\{\tilde{s}_t, \tilde{p}_t, \tilde{f}_t\}$ while the actual subdivision into segments is represented as $\{s_t, p_t, f_t\}$ where, as earlier, $s \in \{0,1\}$ is an indicator of the presence (s=1) or absence (s=0) of segment boundary, p and f – the labels of segments preceding and following the point under consideration, t – a time index. Then, since each claimed segment boundary b_i is considered to be correct if it coincides with an actual one, $N_{c.c}$ is written as

$$N_{c.c.} = \sum_{i=1}^{N} \delta(s_{b_i} - 1, p_{b_i} - \tilde{p}_{b_i}, f_{b_i} - \tilde{f}_{b_i}), \tag{7}$$

where the discrete delta function δ is defined for three arbitrary variables x, y, z as

$$\delta(x, y, z) = \begin{cases} 1, \textit{ if } x = 0, y = 0, z = 0 \\ 0 \textit{ otherwise} \end{cases} \tag{8}$$

As an input video is modeled as a stochastic process, $N_{c.c}$ is a random variable, and we consider its expected value instead:

$$E\{N_{c.c.}\} = \sum_{i=1}^{N} E\{\delta(s_{b_i} - 1, p_{b_i} - \tilde{p}_{b_i}, f_{b_i} - \tilde{f}_{b_i})\} = \sum_{i=1}^{N} P(s_{b_i} = 1, p_{b_i} = \tilde{p}_{b_i}, f_{b_i} = \tilde{f}_{b_i}), \tag{9}$$

where $P(s_i = 1, p_i, f_i)$ denotes the posterior probability of the presence of a boundary between segments p_i and f_i at candidate point i.

Hence, assuming that the probability $P(s_t = 1, p_t, f_t)$ of segment boundary is pre-calculated for each candidate point t and each segment labels pair $\{p_t, f_t\}$, *the optimal segmentation selects N segment boundaries so as to maximize the rightmost sum of expression (9)*. The more is N, the more points of low probability are generally selected and, hence, the less is the relative expected number of correct boundaries among them. On the other hand, the value N should be high enough to provide an acceptable level of misses. So, this value controls the trade-off between the number of false alarms and the number of misses and, hence, between precision and recall.

N can be chosen so as to provide the maximum of an integral performance measure. In this report it is a $F1$ measure which is a harmonic mean of recall and precision:

$$F1 = \frac{2 * r * p}{r + p}. \tag{10}$$

As it follows from experimental evaluations, F1 has a maximum when recall and precision are approximately equal. From expression (4) - (6) follows that equal recall and precision are

provided when $N=N_a$, or, as N_a is considered as a statistical variable, N is selected as expected number of N_a:

$$N = E\{N_a\}. \tag{11}$$

By analogy with expression (9) the expected number of N_a is calculated as:

$$E\{N_a\} = \sum_{i=1}^{T} E\{\delta(s_i - 1)\} = \sum_{i=1}^{T} P(s_i = 1). \tag{12}$$

2.2 Computing Optimal Segment Boundaries

According to our optimal decision rule for segmentation we wish to select N segment boundaries so as to maximize expression (9). A straightforward exhaustive search over all possible boundary arrangements has an exponential computational complexity on N and thus is unfeasible in most cases. A simple and computationally effective algorithm can be proposed in the particular case where the segments are not labeled. In this case the only input data are a sequence of segment boundary probabilities $\{P_1, P_2, ..., P_T\}$, where $P_i \equiv P(s_i = 1)$. N maximal values can be selected by scanning this sequence and extracting the maximal value N times, which yields the computational complexity on the order of $N \cdot T$. Alternatively, the sequence can be sorted in ascending order of probability and N first values be related to segment boundaries, which yields the complexity on the order of $T\log(T)$ required for sequence sorting.

In the general case, where the segments are distinguished by their label, segment boundaries cannot be selected independently from each other because of constraints of expression **(1)** imposed on segment labels. To attain feasible computational complexity in this case, we propose the following procedure. Omitting variable s we denote the probability of transition from segment p_t to a segment f_t at time moment t as $P(p_t, f_t)$. Given this probability for each time point $t = 1, ..., T$ and for each pair of segment labels, the task is to select N distinct segment boundaries $\{b_1, b_2, ..., b_N\}$ and the corresponding segment labels p_{b_i} and f_{b_i} so as to maximize the sum

$$\sum_{i=1}^{N} P(p_{b_i}, f_{b_i}) \tag{13}$$

taking into account the constraints of expression **(1)**. We define the following variable:

$$M(n, f, t) \equiv \max_{\substack{b_1,...,b_n \\ p_{b_1},...,p_{b_n} \\ f_{b_1},...,f_{b_{n-1}}}} \{\sum_{i=1}^{n-1} P(p_{b_i}, f_{b_i}) + P(p_{b_n}, f)\}, \tag{14}$$

where it is assumed that $1 \le b_1 < ... < b_n \le t$ and expression **(1)** holds true. $M(n, f, t)$ is the best score of expression (13) corresponding to *n* segment boundaries selected for first *t* candidate points given that the last segment is labeled as *f*. By induction we have

$$M(n, f, t) = \max_{1 \le f' \le m, 1 \le t' < t} \{M(n-1, f', t') + \max_{t' < b_n \le t} [P(p_{b_n} = f', f_{b_n} = f)]\}, \tag{15}$$

where m denotes the number of segment labels. To actually retrieve the sequence of optimal segment boundaries, we need to keep track of arguments which maximized expression (15). We do this via the arrays $L(n, f, t)$ and $B(n, f, t)$. The complete procedure for finding the best segment boundaries can be now stated as follows:

1) Initializaton:

$$M(1, f, t) = \max_{\substack{1 \le b_1 \le t \\ 1 \le p_{b_1} \le m}} \{P(p_{b_1}, f_{b_1} = f)\},\ 1 \le f \le m, 1 \le t \le T \tag{16}$$

$$L(1, f, t) = \arg\max_{1 \le p_{b_1} \le m} \max_{1 \le b_1 \le t} \{P(p_{b_1}, f_{b_1} = f)\},\ 1 \le f \le m, 1 \le t \le T \tag{17}$$

$$B(1, f, t) = \arg\max_{1 \le b_1 \le t} \max_{1 \le p_{b_1} \le m} \{P(p_{b_1}, f_{b_1} = f)\},\ 1 \le f \le m, 1 \le t \le T \tag{18}$$

2) Recursion:

$$M(n, f, t) = \max_{1 \le f' \le m, n-1 \le t' < t} \{M(n-1, f', t') + \max_{t' < b_n \le t} [P(p_{b_n} = f', f_{b_n} = f)]\}, \tag{19}$$

$$L(n, f, t) = \arg\max_{1 \le f' \le m} \max_{n-1 \le t' < t} \{M(n-1, f', t') + \max_{t' < b_n \le t} [P(p_{b_n} = f', f_{b_n} = f)]\}, \tag{20}$$

$$B(n, f, t) = \arg\max_{t' < b_n \le t} \max_{1 \le f' \le m, n-1 \le t' < t} \{M(n-1, f', t') + P(p_{b_n} = f', f_{b_n} = f)\}, \tag{21}$$

$$2 \le n < N, 1 \le f \le m, n \le t \le T.$$

3) Termination:

$$\{b_N, p_{b_N}, f_{b_N}\} = \arg\max_{t < b_N \le T, 1 \le p_{b_N} \le m, 1 \le f_{b_N} \le m} \max_{N-1 \le t \le T} \{M(N-1, p_{b_N}, t) + P(p_{b_N}, f_{b_N})\}. \tag{22}$$

4) Segment boundaries backtracking:

$$b_n = B(n, f_{b_{n+1}}, b_{n+1}), \tag{23}$$

$$p_{b_n} = L(n, f_{b_{n+1}}, b_{n+1}), \tag{24}$$

$$f_{b_n} = p_{b_{n+1}}, \tag{25}$$

$$n = N-1, N-2, ..., 1.$$

As calculation $M(n, f, t)$ requires on the order of $m \cdot T^2$ operations for each possible triple $\{n, f, t\}$, the resulting computational complexity of the procedure is on the order of $m^2 N \cdot T^3$.

2.3 Ambiguity of Segment Boundary Position

In practical applications segmentation performance measures tolerate some temporal ambiguity τ between detected and actual boundaries when deciding whether there is correspondence between them [GUI 04]. Taking into account this ambiguity allows us to detect boundaries more reliably. In this subsection we propose a required extension to our optimal segmentation rule. For the purpose of simplicity we suppose hereafter in this subsection that labels of segments are not of interest and consider only their positions.

A typical value of τ is about 5 sec [GUI 04] which is normally less than segment length. We assume that segments cannot be shorter than 2τ. In this case it is not possible that two or more actual boundaries correspond to one claimed boundary. As so, if we wish to minimize the number of misses for a fixed number of claimed boundaries, these boundaries should be placed no closer than 2τ from each other as this provides the maximum number of potential correspondences. Several claimed boundaries, however, can still correspond to one actual boundary. This can be used to "artificially" augment precision by claiming several boundaries in the vicinity of highly probable actual ones where the probability of false alarms is low. That is why we propose a stricter criterion of one-to-one correspondences between claimed and actual boundaries. The maximal number of these correspondences is the number of correct claimed boundaries $N_{c.c.}$ and the number of actual boundaries detected $N_{a.d.}$. As it was earlier, expression (6) holds true and our task is to select N boundaries so at to maximize $N_{c.c.}$. According to the stricter correspondence criterion these boundaries must be spaced no closer to each other than 2τ to minimize the number of misses and false alarms at the same time.

Given an input sequence of segment boundary probabilities $\{P_1, P_2, ..., P_T\}$ let's derive an optimal segmentation rule. Denote as G_i the set of candidate points lying in the vicinity $[t_i - \tau, t_i + \tau]$ of an arbitrary candidate point i occurring at time t_i. Under our assumption only one actual boundary can be found in this region. Hence, the probability of a single claimed boundary placed at point i to be correct is written as

$$P(c(i) = 1) = \sum_{j \in G_i} P_j \ , \tag{26}$$

where $c(i) \in \{0,1\}$ is indicator function which is equal to 1 when a boundary claimed at point i is correct and 0 otherwise. Since claimed boundaries are not closer to each other than 2τ and, hence, their corresponding regions G are not overlapped, the expected number of correct boundaries $N_{c.c.}$ is calculated as

$$E\{N_{c.c.}\} = E\{\sum_{i=1}^{N} \delta(c(b_i) - 1)\} = \sum_{i=1}^{N} P(c(b_i) = 1) = \sum_{i=1}^{N} \sum_{j \in G_i} P_j \ . \tag{27}$$

Optimal segment boundaries are chosen so as to maximize expression (27). We propose to do this iteratively. At each iteration step the sum of expression (26) is computed at each candidate point. The point *i* with the maximal sum is claimed then as a segment boundary and the points in G_i are excluded from the further analysis.

3 Hidden Markov Models

To obtain estimates of segment boundary probability which are required by our optimal segmentation rules considered above, we need to properly choose a model describing an input video. Hidden Markov models (HMM) are powerful tools for modeling the dynamics of different processes evolving in time, such as video [DIM 00, HUA 99, BOR 98] and speech signals [RAB 89, BEN 99]. In this section we provide basic definition and assumptions that underlying these models, consider their different variations suitable for the purpose of video modeling and derive expressions required for segmentation.

3.1 Basic Model

A basic HMM is a stochastic process which at any discrete time $t = 1,2,...,T$ is at one of a set of *N* distinct states $Q^* = \{1,2,..., N\}$. We denote the actual state at time *t* as $q_t \in Q^*$. The dynamics of the process is then described as a sequence $Q = \{q_1, q_2,..., q_T\}$. At each time moment the model changes the state (or remains at the same state) according to the probability values associated with the state. A complete probabilistic description of a stochastic process requires specification of the current state depending on all the predecessor states. The HMM is defined as the special case of a first order Markov chain, where the probability to be in the current state q_t is determined completely by the predecessor state, i.e.

$$P(q_t = j \mid q_{t-1} = i, q_{t-2},...) = P(q_t = j \mid q_{t-1} = i) \equiv a_{ij}, \tag{1}$$

where a_{ij} denotes the probability of transition from state *i* to state *j*. It is supposed that the HMM is stationary and, hence, a_{ij} is independent on the time index. The initial state is chosen according to the probability denoted as

$$\pi_i \equiv P(q_1 = i). \tag{2}$$

We collect all state transition probabilities into one matrix $A = \{a_{ij}\}$ which satisfies the following stochastic constraints:

$$a_{ij} \geq 0,\ 1 \leq i \leq N,\ 1 \leq j \leq N, \tag{3}$$

$$\sum_{j=1}^{N} a_{ij} = 1,\ 1 \le i \le N\,. \tag{4}$$

Depending on applications, additional constraints can be imposed to matrix *A*. Forcing some coefficients to be zero we can forbid the corresponding transitions. Thus, different topologies can be defined that are usually depicted graphically so that the allowed state transitions are shown by arrows. One such model is presented in Figure 1. This is a left-right or Bakis model [BAK 76], for which low numbered states can only make transitions to higher number states or to themselves, i.e. $a_{ij} = 0$ for each $j < i$. This model is suitable for processes whose properties change over time, such as speech signals. If every state of the HMM could be reached from every other state in a single step, the corresponding topology includes all possible connections and is called an ergodic or circular model.

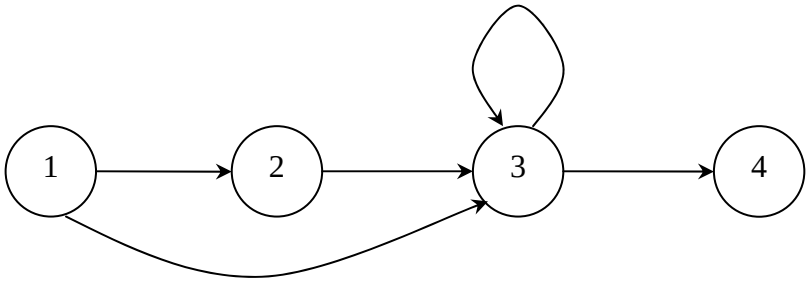

Figure 1. A 4-state left-right HMM.

The states of the HMM are not observable directly (i.e. “hidden”) but generate a vector of measurable features according to a probabilistic function. We denote the feature vector observed at time *t* as D_t. It is assumed that this vector is conditioned only on the current state. We denote the corresponding probability distributions as $B = \{b_j(D_t)\}$, where

$$b_j(D_t) = P(D_t \mid q_t = j),\ 1 \le j \le N\,. \tag{5}$$

The presented above HMM describes double stochastic process. The primary process is not observable, or is hidden, and is determined as a first order Markov chain. The secondary process $D = \{D_1, D_2, ..., D_T\}$ is an observable representation of the primary process generated according to a probabilistic rule. The joint description of these two processes is given by defining the matrix of initial state probabilities $\Pi = \{\pi_i\}$, matrix of transition probabilities *A* and probability distributions for generating observations *B*. This description is a complete specification of a basic HMM.

A widespread approach to the task of video segmentation is to model an input video with a single HMM. The states of the HMM are stationary parts of the video, such as frames or camera shots. Semantic segments are then related to subsequences of the states. The HMM can

be thought as an opaque box, where the sequence of features *D* is observable, while the sequence of the states is hidden. In the simplest case each segment is assigned to a unique state. For example, two different HMM topologies – a two-states ergodic and a left-right one (see Figure 2) – are explored in [ALA 01]. The aim is to separate dialog scenes from non-dialog scenes in movies. The elementary time units in this example are camera shots and state transitions are explored at shot change moments. The limitation of such an approach is that it is not general enough to separate several contiguous semantic segments of the same type. In the more general case each segment is represented by a sequence consisting of different HMM states. For example, news video is divided into story units of the same type in [LEK 02] using a four-states ergodic HMM. This model allows the authors to track dynamic patterns of shots corresponding to news stories.

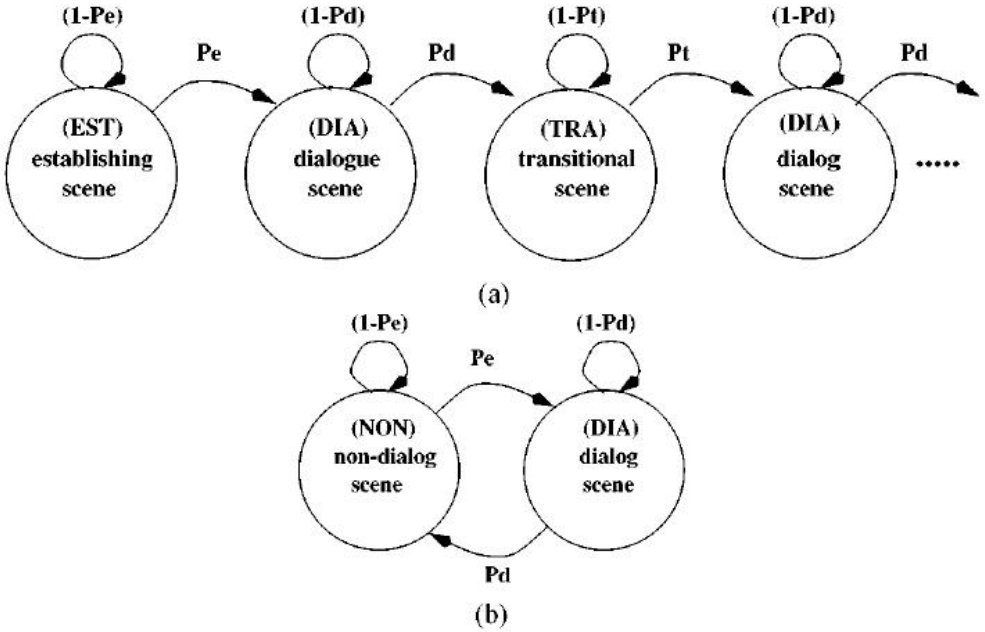

Figure 2. Left-right (a) and circular (b) HMM for modeling dialog scenes in movies [ALA 01].

Semantic segments are commonly detected through reconstructing the full sequence of the HMM states. If each segment is represented by a unique state, then the resulting segments are the corresponding groups of repetitive state labels. If segments are modeled as subsequences of states of several types, then segment boundaries are found as transition to or from unique states which begin or terminate the corresponding subsequences. The common criterion used to find the best sequence of HMM states is maximizing the posterior probability of the sequence $P(Q \mid D)$ which is equivalent to maximizing the joint probability $P(Q, D)$. This probability is written as

$$P(Q,D) = P(Q)P(D \mid Q) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \prod_{t=1}^{T} b_{q_t}(D_t). \qquad (6)$$

The straightforward maximization of this expression using full search over all possible state sequences requires on the order of $2TN^T$ operations which is infeasible for the most applications. Fortunately, there exists a computationally effective technique for finding this best state sequence, based on dynamic programming, and it is called the Viterbi algorithm [VIT 67].

To write down the Viterbi algorithm, let's first define the following variable:

$$\delta_t(i) = \max_{q_1,q_2,...,q_{t-1}} P(q_1, q_2,..., q_t = i, D_1, D_2,..., D_t) . \tag{7}$$

This variable is the highest probability for the first $t-1$ states. It allows one to find the probability of the whole optimal path recursively using the following rule:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(D_{t+1}) . \tag{8}$$

In addition, we define for each t and j the variable $\psi_t(j)$ which is the argument maximizing expression (8). This variable is needed to retrieve the best state sequence after the maximum probability of the whole state sequence has been found. Denoting as $\tilde{P}$ the optimal value for the probability and as $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2,..., \tilde{q}_T\}$ the optimal state sequence, the Viterbi algorithm is resumed as follows:

1) Initialization:

$$\delta_1(i) = \pi_i b_i(D_1), \ \psi_1(i) = 0, \ 1 \le i \le N \tag{9}$$

2) Recursion:

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \{\delta_t(i) a_{ij}\} b_j(D_{t+1}), \tag{10}$$

$$\psi_{t+1}(j) = \arg\max_{1 \le i \le N} \{\delta_t(i) a_{ij}\}, \tag{11}$$

$$1 \le t < T, \ 1 \le j \le N .$$

3) Termination:

$$\tilde{P} = \max_{1 \le i \le N} \{\delta_T(i)\}, \tag{12}$$

$$\tilde{q}_T = \arg\max_{1 \le i \le N} \{\delta_T(i)\} . \tag{13}$$

4) State sequence backtracking:

$$\tilde{q}_t = \psi_{t+1}(\tilde{q}_{t+1}), \ t = T-1, T-2,..., 1 . \tag{14}$$

It is easy to see that the computational complexity of the Viterbi algorithm is on the order of $N^2 \cdot T$.

As it was discussed above in this report, segmentation via reconstruction of complete state sequence does not necessarily lead to the optimal system performance. To find the optimal segment boundaries according to our optimality criterion, we need to estimate the posterior probability of segment boundaries at each candidate point. For this purpose we first

define $\xi_t(i, j)$, the probability of transition from state *i* at time *t* to state *j* at time $t+1$, given the observation *D*:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j \mid D). \tag{15}$$

For computationally effective calculation of this value we use the forward-backward procedure [RAB 89] as follows. Consider the forward variable $\alpha_t(i)$ defined as the probability of the partial observation sequence until time *t* and state *i* at time *t*:

$$\alpha_t(i) \equiv P(D_1, D_2, ..., D_t, q_t = i). \tag{16}$$

This variable is calculated by induction as

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(D_{t+1}),\ 1 \le t < T,\ 1 \le j \le N, \tag{17}$$

where initial value is

$$\alpha_1(i) = \pi_i b_i(D_1),\ 1 \le i \le N. \tag{18}$$

In a similar manner a backward variable $\beta_t(i)$ is defined as

$$\beta_t(i) \equiv P(D_{t+1}, D_{t+2}, ..., D_T, q_t = i). \tag{19}$$

Initialized with

$$\beta_T(i) = 1,\ 1 \le i \le N, \tag{20}$$

it is calculated by the following induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(D_{t+1}) \beta_{t+1}(j),\ t = T-1, T-2, ..., 1,\ 1 \le i \le N. \tag{21}$$

After applying the forward-backward procedure, variable $\xi_t(i, j)$ is calculated as

$$\xi_t(i, j) = \frac{P(q_t = i, q_{t+1} = j)}{P(D)} = \frac{\alpha_t(i) a_{ij} b_j(D_{t+1}) \beta_{t+1}(j)}{P(D)}, \tag{22}$$

where $P(D)$ can be calculated, for instance, as

$$P(D) = \sum_{i=1}^{N} \alpha_T(i). \tag{23}$$

Segment boundaries are related to transitions between the HMM states. Hence, the candidate points of these boundaries are $T-1$ potential transitions within the sequence of *T* states. If a segment boundary corresponds to a single pair of states *i* and *j*, as for instance in the case where each segment is represented by one state, then its posterior probability is $\xi_t(i, j)$. In the general case segments are modeled by subsequences consisting of different states. To separate these subsequences, one could mark their beginning or the end with a special state or model segments with non-overlapping sets of states. Let's denote the set of states which can end an arbitrary segment $s1$ as G_1 and the set of states which can begin an arbitrary set $s2$ – as G_2. Then

the probability that a boundary between segments $s1$ to $s2$ corresponds to the transition between states q_t and q_{t+1} is computed as

$$\sum_{i \in G_1} \sum_{j \in G_2} \xi_t(i, j) . \tag{24}$$

3.2 Hierarchical Model

The content of video is often organized in a hierarchical manner, e.g. a tennis match can be divided first into sets, then the sets are decomposed into games etc. In this subsection we present a generalization of the basic HMM, called a hierarchical HMM (HHMM) [SHA 98], which models this organization directly. These models have found a wide use in many domains of application with hierarchical structure, such as image and video segmentation [PHU 05, ZHE 04], visual action recognition [NGU 05, MOO 01, HOE 01], spatial navigation [BUI 01, THE 01] and handwriting recognition [SHA 98]. The advantage of HHMMs is that they take into account statistical dependences existing between structural elements at multiple levels of coarseness, thus enabling to model long-term correlations between observable feature vectors.

A HHMM is a structured process defined as a Markov chain whose states are hidden and modeled with their proper lower-level Markov chains. At the lowest level of the hierarchy this process is an ordinary HMM, whose states generate observable feature vectors according to a probabilistic rule. The states of higher levels aggregate the lower-level state chains. Therefore they generally correspond to sequences of feature vectors. These sequences are generated in a recursive manner by activation the corresponding sub-models which may be composed of sub-models as well. This process terminates when states of the lowest-level are reached. The lowest-level states are called *production states* as they are the only states which emit observable data. The states of the higher-levels do not generate observable features directly and are called *internal* or *abstract states*.

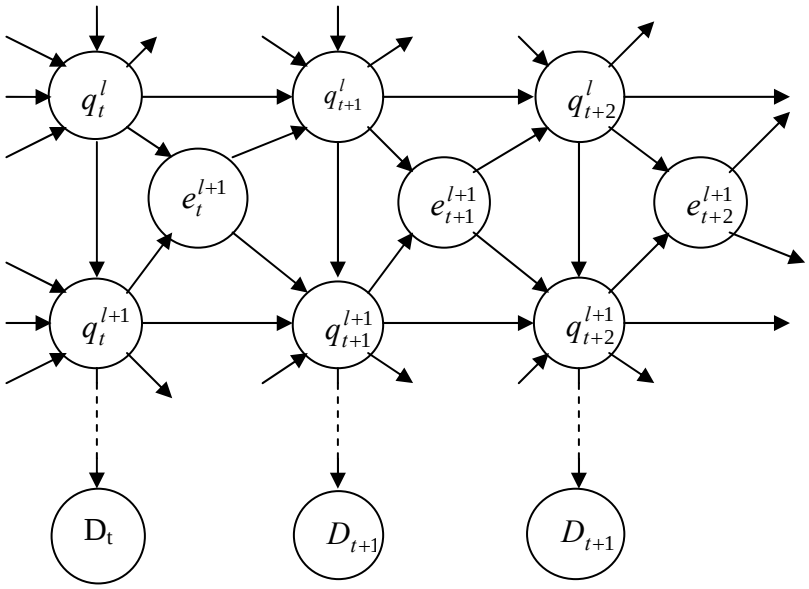

Figure 3. DBN representation of a HHMM at level l and $l+1$ at time t, t+1, t+2. q_t^l denotes the state at time t, level l; e_t^l is an indicator variable that the HMM at level l has finished at time t; D_t is the observable feature vector.

A HHMM can be graphically represented as a dynamic Bayesian network (DBN) [MUR 01], as shown in Figure 3. The state of the model at level l and time t is denoted as q_t^l. When the model enters the abstract state, the corresponding sub-model is activated in a recursive manner. This activation is called a *vertical transition*. When the sub-model is finished (which may engender activations of lower level states recursively), the control returns to the upper-level state it was called from. Then a state transition within the same level, called a *horizontal transition*, occurs. A sub-model finishes when a special *end* state is reached. This state never emits observable data and immediately engenders the transition to the calling state. To indicate that the sub-model is about to enter the end state, the corresponding indicator variable of the DBN representation $e_t^l \in \{0,1\}$ is set to 1, otherwise it is equal to 0.

The calling context of vertical transitions is stored in a depth-limited stack. Any HHMM can be converted to an ordinary HMM by enumerating all possible states in the stack, from the highest model level up to the lowest one. Assuming that the HHMM has L levels and that all production states are at the lowest level L, the states of the equivalent HMM are encoded by mapping the calling context $q_t^{1:L} = \{q_t^1,...,q_t^L\}$ of each production state into integers. The same sub-model of the HHMM can be shared by several sub-models of the upper level. In the HMM representation this shared sub-model must be duplicated for each calling context, which generally results in a larger model. So, the power of the HHMMs is in the ability to reuse its substructures. As a result, they have a more compact representation, and the less number of

parameters simplifies their learning. The hierarchical representation of HHMMs also allows us to specify their topology or constraints on possible state transition in a more natural way. Using a chain of sub-models allows the authors to impose a constraint on the minimum number of the corresponding semantic segments, e.g. a game segment consists of no less than 4 points. At the same time, these sub-models are not duplicated superfluously.

In order to give a strict formal definition of the HHMM, let's specify conditional probability distributions of each node type in the corresponding DBN representation (see Figure 3). Consider first the lowest level L of the model. The states of this level follow the rules of a regular HMM, whose parameters are determined by its position in the HHMM encoded by the vector of higher state variables $q_t^{1:L-1} = \{q_t^1, ..., q_t^{L-1}\}$. For simplicity of notations we represent this vector by the integer k. When the HMM is activated, its initial state j is selected according to the prior distribution $\pi_k^L(j)$ defined for the parent state vector encoded by k. Then at subsequent time moments it undergoes a change of state according to the state transition matrix A_k^L until the *end* state is reached. In the DBN representation the system never enters the *end* state, but the corresponding variable e_t^L is set to 1 instead, indicating that the higher-level sub-model can now change its state. Thus the conditional probability of a state at level L is written as

$$P(q_t^L = j \mid q_{t-1}^L = i, e_{t-1}^L = f, q_t^{1:L-1} = k) = \begin{cases} \widetilde{A}_k^L(i,j), \text{ if } f = 0 \\ \pi_k^L(j), \text{ if } f = 1 \end{cases} \tag{25}$$

where it is assumed that $i, j \neq end$. Matrix $\widetilde{A}_k^l$ is the state transition matrix at level l given that the parent variables are in state k and the *end* state is never reached, i.e. it is defined from the following equality:

$$\widetilde{A}_k^l(i,j)(1 - A_k^l(i,end)) = A_k^l(i,j). \tag{26}$$

The conditional probability for e_t^L is determined as

$$P(e_t^L = 1 \mid q_t^{1:L-1} = k, q_t^L = i) = A_k^L(i,end). \tag{27}$$

The observable feature vector D_t is generated according to a probability function conditioned on the whole stack configuration $q_t^{1:L}$.

To write down the conditional probabilities for intermediate level l, we need also to take into consideration the variable e_t^{l+1} indicating whether the sub-model has finished or not. If this variable is 0, which means that the sub-model has not finished, the state transition at level l is forbidden. Hence, the conditional probability of state q_t^l is written as

$$P(q_t^l = j \mid q_{t-1}^l = i, e_{t-1}^{l+1} = b, e_{t-1}^l = f, q_t^{1:l-1} = k) = \begin{cases} \delta_{ij}, \text{ if } b = 0 \\ \tilde{A}_k^l(i,j), \text{ if } b = 1 \text{ and } f = 0 \\ \pi_k^l(j), \text{ if } b = 1 \text{ and } f = 1 \end{cases} \quad (28)$$

where δ_{ij} is the Kronecker delta. The variable e_t^l can be set to 1 only when the state q_t^l is allowed to enter a final state. Therefore, its conditional probability is written as

$$P(e_t^l = 1 \mid q_t^l = i, q_t^{q:l-1} = k, e_t^{l+1} = b) = \begin{cases} 0, \text{ if } b = 0 \\ A_k^l(i, end), \text{ if } b = 1 \end{cases} \quad (29)$$

The conditional probabilities for the top level of the HHMM are written similarly to expression (28) and (29). The only difference that the no parent states are to be specified, that is why the conditioning on $q_t^{1:l-1} = k$ must be omitted.

3.3 State Duration Modeling

The proper modeling of semantic segments of video should account for their duration constraints which can be formulated as the corresponding probability distribution. If a segment is modeled with a single state of a HMM, the inherent duration probability density is always meet a geometric distribution. Indeed, the probability of the Markov chain to remain at a state *i* during first *d* time moments is written as

$$P(q_1 = i, ..., q_{d-1} = i, q_d \neq i) = (a_{ii})^{d-1}(1 - a_{ii}). \quad (30)$$

This geometric distribution is often not appropriate. For example, segments of short duration are unlikely as they have not enough time to convey the semantics to a viewer, while according to this distribution they should be of the highest probability (see the left part of Figure 4).

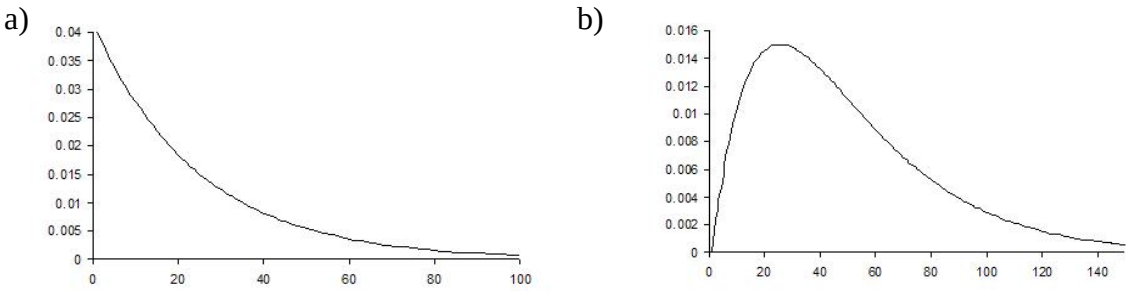

Figure 4. A sample plot of the inherent duration probability for the 1-state (a) and 2-state (b) Markov chain (a=0.96).

The duration distribution can be fit more freely if the segment is modeled by a chain consisting of several different states. To make this distribution to be decreasing when the duration approaches zero, two state are enough. Consider the two-state chain presented in Figure

5. Denoting as $P_1(x)$ and $P_2(x)$ the probability of remaining x times in state 1 and 2 respectively, the probability of remaining in the whole chain is written as

$$P(d) = \sum_{x=1}^{d-1} P_1(x) P_2(d-x) = \sum_{x=1}^{d-1} (a_{11})^{x-1} (1-a_{11}) (a_{22})^{d-x-1} (1-a_{22}), \tag{31}$$

where the second equality follows from expression (30). Assuming for simplicity that $a_{11} = a_{22} \equiv a$, expression (31) is continued as

$$P(d) = \frac{1-a^2}{a^2} a^d \sum_{x=1}^{d-1} 1 = (d-1)(1-a)^2 a^{d-2}. \tag{32}$$

This is a second-order Erlang distribution, a discrete counterpart of the gamma-distribution, which, for instance, has been shown to be good fit for the probability density function of shot duration in [VAS 97]. A sample plot of this distribution is shown in the right part of Figure 4.

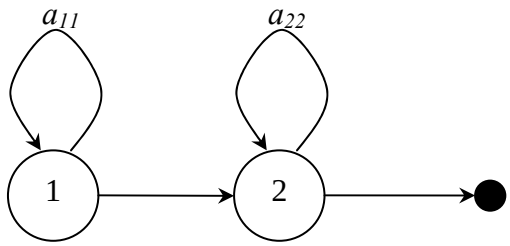

Figure 5. A two-state HMM.

Markov chains of sufficient size can model general probability distributions [CRY 88]. Hence, in order to properly realize the state duration, the HMM can be expanded so that its states are expanded to sub-models which have their own topology and transition probability. The resulting structure is called the expanded state HMM (ESHMM) [RUS 87]. The lower-level sub-models are regular HMMs whose states have the same emission probability functions. They usually have a compact left-right topology. The transition coefficients can be learned with the Baum-Welch procedure [RAB 89], an EM-algorithm commonly used for HMM parameters estimation. Alternatively, these coefficients can be calculated directly from the estimated statistical moments [BON 96].

In many applications the state duration distributions in the ESHMM are fitted with quite compact sub-models, thus not increasing crucially the computational complexity with respect to the original HMM. For instance, in [BON 96] three states are assumed to be enough for modeling phone durations in the task of speech recognition. Since the complexity of the probability computations for the regular HMM is quadratic with respect to the total number of the states, the resulting three times growing in the total size of the model engender at most a nine times increase in the computational burden.

The ESHMM is suitable for the tasks where the segment duration distributions are fixed and can be fitted only once during the preliminary learning. Sometimes, however, there is a need to recalculate these distributions at each time step. These recalculations with the ESHMM lead to unacceptable computational complexity. Such a need in the re-estimation of the duration probability arises, for example, when the time units corresponding to the states are not of regular duration, while the distributions of segment duration are defined in the domain of natural time measured in regular units. This is the case in our task of narrative video segmentation into logical story units, or scenes. The elementary time units are camera shots whose duration is not regular and can change from 1-2 seconds to half a minute or even more. The shot length can change considerably from one scene to another, depending on the conveyed semantic, while the time distribution of scene duration remains more or less stable. We estimate the probability of a scene change as a function of shots length and the time duration of the scene. The resulting state transition probabilities of the corresponding model are dependent from these terms as well and change from one candidate point to another. Such a non-stationary system seems to be modeled more effectively with an extension to the regular HMM where the state duration probability is modeled explicitly. This kind of a model is called a variable duration HMM [RAB 89] or a hidden semi-Markov model (HSMM) [RUS 85].

The functional difference of the HSMM in respect to the regular HMM, is that in the HSMM the transitions from the states back to themselves are prohibited, i.e. the diagonal elements of the state transition matrix $a_{ii} = 0$. Instead of the value of a_{ii}, which implicitly define the state duration in the regular HMM, the occupancy of the state is now determined by an explicit probability distribution. For the practical aspects discussed above, in this report we extend the HSMM to be non-stationary in the sense that state duration distributions are defined at each time step. The evolution of the process described by the HSMM is defined as follows. An initial state q_1 is chosen according to the initial state distribution π_i. Once activated, each state i remains unchanged during x consecutive time moments, where x is chosen according to the state duration density $p_i^t(x)$, which is supposed to be non-stationary and dependent on the state activation time t. It is assumed that the duration density $p_i^t(x)$ is defined to be non-zero up to a maximum possible duration value τ_i^t. When state i is finished, the sequence of observable feature vectors is generated according to the joint observation density $b_i(D_{t:t+x-1})$. The next state j ($j \neq i$) is chosen then according to the state transition probabilities a_{ij}.

To be applied to the HSMM, the forward-backward procedure, used for computationally effective calculation of the posterior state transition probabilities, is modified as follows. We

assume that the first state begins at time $t = 1$, and the last state ends at $t = T$, i.e. the model comprises only entire state duration intervals. The forward variable $\alpha_t(i)$ is now defined as

$$\alpha_t(i) = \begin{cases} P(D_{1:t}, q_t = i, q_{t+1} \neq i), \text{ if } t < T \\ P(D_{1:t}, q_t = i), \text{ if } t = T \end{cases}, \ 1 \leq t \leq T, \tag{33}$$

where $D_{i:j}$, $j > i$, denotes the sub-sequence of observable data $D_i, D_{i+1}, ..., D_j$. In the other words, the forward variable defines the probability of observing *t* first data vectors and the state *i* finishing at time *t*. The variable is initialized as

$$\alpha_1(i) = \pi_i p_i^1(1) \cdot b_i(D_1), \ 1 \leq i \leq N. \tag{34}$$

For the subsequent time moments $t = 2, ..., T$ we have the following induction:

$$\alpha_t(j) = \pi_j p_j^1(t) \cdot b_j(D_{1:t}) + \sum_{i=1}^{N} \sum_{\substack{1 \leq k \leq t-1 \\ k \geq t - \tau_j^k}} \alpha_k(i) a_{ij} p_j^k(t-k) b_j(D_{k+1:t}), \ 1 \leq j \leq N. \tag{35}$$

The first term of this expression disappears when time *t* exceeds the maximum possible state duration τ_j^1. The value τ_j^k limits the range for the second sum of the second term for time *t* so that the state duration does not exceed its maximum allowed value (in algorithmic realization this limit can be effectively tracked with a queue of values τ_j^k, whose elements are discarded when $t > \tau_j^k + k$). The probability of observing the whole sequence of feature vectors is written in terms of the α's as

$$P(D_{1:T}) = \sum_{i=1}^{N} \alpha_T(i). \tag{36}$$

We also define two backward variables as

$$\beta_t(i) = P(D_{t+1:T} \mid q_t = i, q_{t+1} \neq i), \ 1 \leq i \leq N, \tag{37}$$

$$\beta_t^*(i) = P(D_{t+1:T} \mid q_t \neq i, q_{t+1} = i), \ 1 \leq i \leq N, \tag{3-38}$$

i.e. $\beta_t(i)$ and $\beta_t^*(i)$ are the probabilities of partial feature vector sequence $D_{t+1:T}$ given that state *i* ends at time *t* and given that state *i* begins at time $t+1$ respectively. We initialize the recursion as

$$\beta_T(i) = 1, \ 1 \leq i \leq N. \tag{39}$$

Then for $t = T-1, T-2, ..., 1$ by induction we have

$$\beta_t^*(i) = \sum_{x=1}^{\min\{\tau_i^t, T-t\}} \beta_{t+x}(i) p_i^t(x) b_i(D_{t+1:t+x}), \ 1 \leq i \leq N, \tag{40}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_t^*(j), \ 1 \leq i \leq N. \tag{41}$$

The posterior probability of state transitions are computed based on the forward-backward variables as

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid D_{1:T}) = \frac{\alpha_t(i) a_{ij} \beta_t^*(j)}{P(D_{1:T})}. \tag{42}$$

To find the most probable sequence of HSMM states, the Viterbi algorithm must be modified so that to account all possible durations of states. Defining the variable $\delta_t(i)$ to be the probability of the best state sequence such that the last state *i* ends at time *t*, by induction we have

$$\delta_t(j) = \max_{\substack{1 \le i \le N \\ k \ge t - \tau_j^k}} \max_{1 \le k \le t-1} \{\delta_k(i) a_{ij} p_j^k(t-k) b_j(D_{k+1:t})\} + \pi_j p_j^1(t) b_j(D_{1:t}),\ 1 \le j \le N. \tag{43}$$

The observable feature vectors are usually assumed to be conditionally independent on each other. Therefore the joint probability of these vectors measured at an arbitrary time run from *j* to *k* at a model state *i* is calculated as

$$b_i(D_{j:k}) = \prod_{l=j}^{k} b_i(D_l). \tag{44}$$

Taking into account this equality, the comparisons of the expressions for the forward-backward variables for the basic HMM (17) - (21) and the HSMM (34) - (41) allows us to conclude, that the HSMM requires about $\tau^2/2$ times the computation, where τ denotes the average value of τ_j^k. The same is true for the Viterbi procedure as well. This increase in computational burden is, however, not crucial in our task of narrative video segmentation, since the model is applied only once for an input video and the main computational efforts are still required for the feature vector extraction. A pruning theorem is proposed in [BON 93], which reduces significantly the search space in the Viterbi induction (43). The resulting increase of computational effort is reported to be about 3.2 times with respect to a conventional HMM, which is usually considerable lower than the use of the original technique. The pruning theorem requires, however, that the state duration distributions be log-convex, which is difficult to provide for our non-stationary model.

3.4 Autoregressive Model

The conventional HMM assumes that the observable feature vectors are statistically dependent only on the current states. However it is often the case that there is a strong inherent correlation between consecutive feature vectors, which breaks this assumption. To deal properly with unwanted dependencies, we could consider the joint probabilities of several consecutive feature vectors. But this would require expanding the dimension of the probability functions, which

would make more difficult their learning. Alternatively, we could fit the time series of feature vectors with some model, which would allow us to get rid of the information redundancy and pass to a sequence of independent data. An extension to the conventional HMM, where the initial sequence of feature vectors is considered as an autoregressive process, is called an autoregressive HMM (ARHMM). This model was initially proposed for speech signals [JUA 85].

A time series $d_1, d_2, ..., d_T$ is said to represent an autoregressive process, if it can be written as

$$d_t = \mu - \sum_{k=1}^{p} a_k d_{t-k} + e_t , \tag{45}$$

where a_k are p constant coefficients, μ is the process mean, e_t is assumed to be a white noise process with mean zero and variance δ^2. The functional difference of the ARHMM is that it does not assume any longer the conditional independence of the current observable feature vector from the past observations, i.e. in the general case

$$P(D_t \mid q_t, q_{t-1}, ..., q_1; D_{t-1}, D_{t-2}, ..., D_1) \neq P(D_t \mid q_t) . \tag{46}$$

We assume that observable vector D_t consists of K statistically independent components, i.e. $D_t = \{d_t^1, d_t^2, ..., d_t^K\}$. Thus, an autoregressive model can be applied independently for each component and, hence, its upper index is hereafter omitted. As it follows from expression (45), an observable feature can be written as

$$d_t = \hat{d}_t + e_t , \tag{47}$$

where $\hat{d}_t$ denotes the predicted value calculated as

$$\hat{d}_t = \mu - \sum_{k=1}^{p} a_k d_{t-k} . \tag{48}$$

In the other words, values a_k can be considered linear prediction coefficients. Then the independent statistical variable e_t is written as the difference between the real and predicted values of the feature:

$$e_t = d_t - \hat{d}_t = d_t - \mu + \sum_{k=1}^{p} a_k d_{t-k} . \tag{49}$$

In the ARHMM e_t is assumed to be conditionally dependent only on the current state and, in fact, replaces the feature value of the conventional HMM. We additionally assume that this value has a Gaussian distribution. The probability of observing e_t at model state i is substituted in the ARHMM by the following value

$$b_i(e_t) \equiv P(e_t \mid q_t = i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2\sigma_i^2} \left(d_t - \mu_i + \sum_{l=1}^{p} d_{t-l} a_l^i \right) \right), \tag{50}$$

where σ_i and μ_i are the deviation and the mean corresponding to state *i*, a_l^i - *l*-th autoregressive coefficient corresponding to state *i*. It is assumed that in the general case the observation are generated by different mechanisms at different states. That is why the autoregressive parameters in expression (50) are defined depending on the current state.

To estimate the autoregressive parameters of the ARHMM, we use the maximum likelihood learning criterion. As our final task is the video segmentation, it is assumed that the model is trained on a pre-segmented set of videos. We further assume that each semantic segment corresponds to a single state, i.e. the learning videos are, in fact, marked up into states. Given a training video of length *T*, the maximum likelihood estimates are selected so as to maximize the log-likelihood of the observable sequence written as

$$P_{ML} \equiv \log P(d_1, d_2, ..., d_T \mid q_1, q_2, ..., q_T) = \sum_{t=1}^{T} \log b_i(e_t) . \tag{51}$$

Substituting expression (50) for $b_i(e_t)$, we write the log-likelihood as

$$P_{ML} = \alpha - \beta \sum_{t=1}^{T} \left(\log \sigma_{q_t}^2 + \frac{1}{\sigma_{q_t}^2} \left(d_t - \mu_{q_t} + \sum_{k=1}^{p} a_k^{q_t} d_{t-k} \right)^2 \right), \tag{52}$$

where α and β are inessential constants. The optimal autoregressive parameters can be found independently for each state by equaling the partial derivatives to zero:

$$\partial P_{ML} / \partial a_l^i \propto \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right) d_{t-l} = 0 , \tag{53}$$

$$\partial P_{ML} / \partial \mu_i \propto \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right) = 0 . \tag{54}$$

The resulting system of linear equations can be rewritten as

$$\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \begin{bmatrix} d_t d_{t-1} \\ d_t d_{t-p} \\ .... \\ d_t d_{t-p} \\ d_t \end{bmatrix} = \sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \begin{bmatrix} d_{t-1} d_{t-1} & d_{t-2} d_{t-1} & d_{t-p} d_{t-1} & d_{t-1} \\ d_{t-1} d_{t-2} & d_{t-2} d_{t-2} & d_{t-p} d_{t-2} & d_{t-2} \\ & & & \\ d_{t-1} d_{t-p} & d_{t-2} d_{t-p} & d_{t-p} d_{t-p} & d_{t-p} \\ d_{t-1} & d_{t-2} & d_{t-p} & 1 \end{bmatrix} \begin{bmatrix} a_1^i \\ a_2^i \\ \\ a_p^i \\ \mu_j \end{bmatrix} . \tag{55}$$

After solving this system with respect to a_k^i and μ_i, these parameters can be used to estimate variation σ_i^2 by equaling the corresponding partial derivative to zero, which yields

$$\sigma_i^2 = \frac{\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} \left(d_t - \mu_i + \sum_{k=1}^{p} a_k^i d_{t-k} \right)^2}{\sum_{\substack{t=1 \\ s.t. q_t = i}}^{T} 1}. \qquad (56)$$

Expression (55) and (56) can be easily generalized for the case where several learning videos are provided by extending the sums on *t* to all the available data.

4 Conclusions

A statistical framework has been proposed for the task of video segmentation which focuses on the detection of segment boundaries. The common approach to the task is to select the single best model of the whole video. This does not necessarily lead to the optimal segmentation performance which is commonly measured in terms of recall and precision. In our approach we select segment boundaries so as to maximize the performance metrics directly. The approach is based on the posterior probabilities of the boundaries estimated at each candidate point. It is finally formulated as a task of constrained optimization, for which a computationally feasible algorithm, applicable to the general case of multiple semantic segments, is proposed.

The posterior probabilities of segment boundaries can be estimated in different ways, depending on the particular model of the video. In this report we describe a hidden Markov model and its modifications which have been shown to be effective tools for modeling the dynamics of time sequences, such as video. A basic model is first defined, and its application to the video segmentation task is considered. Several modifications of this model are presented then, which allow us to overcome some inherent limitations: a hierarchical extension used to model multi-level semantic structure; a hidden semi-Markov model which enable the use of arbitrary distributions of state duration; an autoregressive version which deals properly with statistical interdependencies existing between consecutive feature vectors.

References

[ALA 01] Alatan A.A., Akansu A.N., Wolf W., “Multi-Modal Dialogue Scene Detection Using Hidden Markov Models for Content-Based Multimedia Indexing”, Multimedia Tools and Applications, Vol. 14, No. 2, pp. 137-151, 2001.

[BAK 76] Bakis R., “Continuous speech recognition by statistical methods”, *Proc. ASA Meeting*, Washington DC, 1976.

[BEN 99] Benjio Y., “Markovian Models for Sequential Data”, *Neural Computing Survey*, Vol.2, pp. 129-162, 1999.

[BON 93] Bonafonte A., Ros X., Marino J.B., “An Efficient Algorithm to Find the Best State Sequence in HSMM”, *Proc. of Eurospeech*, pp. 1547-1550, 1993.

[BON 96] Bonafonte A., Vidal J., Nogueiras A., “Duration Modeling with Expanded HMM Applied to Speech Recognition”, *Proc. Int. Conf. on Spoken Language Processing*, USA, pp. 1097-1100, 1996.

[BOR 98] Boreczky J., Wilcox L., “A Hidden Markov Model Framework for Video Segmentation Using Audio and Image Features”, Proc. IEEE Conf. on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3741-3744, 1998.

[BUI 01] Bui H., Venkatesh S., West G., “Tracking and Surveillance in Wide-Area Spatial Environments Using the Abstract Hidden Markov Model”, *Int. J. of Pattern Recognition and AI*, 2001.

[CRY 88] Crystal T.H., House A.S., “Segmental Durations in Connected Speech Signals: Current Results”, *Journal of Acoustic Society of America*, Vol. 83, No.4, pp. 1553-1573, 1988.

[DIM 00] Dimitrova N., Agnihotri L., Wei G., “Video Classification Based on HMM Using Text and Faces”, *European Signal Processing Conference*, Tampere, Finland, 2000.

[EIC 99] Eickeler S., Muller S., “Content-Based Video Indexing of TV Broadcast News Using Hidden Markov Models”, *IEEE ICASSP*, USA, pp. 2997-3000, 1999.

[GUI 04] “Guidelines for the TRECVID 2004 Evaluations”, in http://www-nlpir.nist.gov/projects/tv2004/tv2004.html, 2004.

[HOE 01] Hoey J., “Hierarchical Unsupervised Learning of Facial Expression Categories”, *ICCV Workshop on Detection and Recognition of Events in Video*, 2001.

[HSU 03] Hsu W., Chang S.-F., “A Statistical Framework for Fusing Mid-Level Perceptual Features in News Story Segmentation”, *IEEE Int. Conference ICME*, 2003.

[HSU 04] Hsu W., Kennedy L., Huang C.-W., Chang S.-F., Lin C.-Y., Iyengar G., “News Video Story Segmentation Using Fusion of Multi-Level Multi-Modal Features in TRECVID 2003”, *IEEE Int. Conference ICASSP*, 2004.

[HUA 99] Huang J., Liu Z., Wang Y., Chen Y., Wong E.K., “Integration of Multi-modal Features for Video Scene Classification Based on HMM”, *IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, 1999.

[JUA 85] Juang B.H., Rabiner L.R., “Mixture Autoregressive Hidden Markov Models for Speech Signals”, *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-33, No. 6, pp. 1404-1413, 1985.

[LEK 02] Lekha Chaisorn, Tat-Seng Chua, Chin-Hui Lee, “The Segmentation of News Video into Story Units”, *Proc. IEEE ICME*, 2002.

[LEW 91] Lewis D.D., “Evaluating Text Categorization”, *Proc. of the Speech and Natural Language Workshop*, pp.312-318, 1991.

[MOO 01] Moore D., Essa I., “Recognizing Multitasked Activities Using Stochastic Context-Free Grammars”, *CVPR Workshop on Models vs Exemplars in Computer Vision*, 2001.

[MUR 01] Murphy K.P., Paskin M.A., “Linear Time Inference in Hierarchical HMMs”, Proc. of Neural Information Processing Systems, Vancouver, Canada, 2001.

[NGU 05] Nguen N., Venkatesh S., “Discovery of Activity Structures Using the Hierarchical Hidden Markov Model”, *16th British Machine Vision Conference*, Oxford, UK, 2005.

[PHU 05] Phung D.Q., Duong T.V., Venkatesh S., Bui H.H., “Topic Transition Detection Using Hierarchical Hidden Markov and Semi-Markov Models”, Proc. of ACM Multimedia, Singapore, pp. 11-20, 2005.

[RAB 89] Rabiner L.R., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”. *Proc. of the IEEE*, 77, no. 2, pp. 257-286, Feb. 1989.

[RIJ 79] Rijsbergen C.J., “Information Retrieval”, Butterworths, 1979.

[RUS 85] Russel M.J., Moore R.K., “Explicit modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition”, *Proc. IEEE Int. Conference on Acoustic, Speech and Signal Processing*, pp. 5-8, 1985.

[RUS 87] Russel M.J., Cook A.E., “Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition”, *Proc. IEEE Int. Conference on Acoustic, Speech and Signal Processing*, Dallas, pp. 2376-2379, 1987.

[SHA 98] Shai Fine, Yoram Singer, Naftali Tishbi, “The Hierarchical Hidden Markov Model: Analysis and Applications”, *Machine Learning*, Vol. 32, pp. 41-62, 1998.

[THE 01] Theocharous G., Rohanimanesh K., Mahadevan S., “Learning Hierarchical Partially Observed Markov Decision Process Models for Robot Navigation”, IEEE *ICRA*, Seoul, Korea, 2001.

[VAS 97] Vasconcelos N., Lippman A., “A Bayesian Video Modeling Framework for Shot Segmentation and Content Characterization”, *Proc. of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[VIT 67] Viterbi A.J., “Error bounds for convolutional codes and an asymptotically optimal decoding algorithm”, *IEEE Trans. Informat. Theory*, Vol. IT-13, pp. 260-269, Apr. 1967.

[ZHE 04] Zhen Ye, Cheng-Chang Lu, “A Wavelet Domain Hierarchical Hidden Markov Model”, *Proc. IEEE ICIP’04*, pp.3491-3494, 2004.