

# **An Automatic Video Classification System Based on a Combination of HMM and Video Summarization**

Cheng Lu, Mark S. Drew, and James Au  
School of Computing Science  
Simon Fraser University  
Vancouver, B.C., CANADA V5A1S6  
{clu,mark,ksau}@cs.sfu.ca  
(604) 291-4682 Fax (604)291-3045

**Keywords:** Feature Identification and Classification; Markov Chain Model; Identification Systems; Image Processing; Maximum Likelihood Classifier; Video Classification.

## **Abstract**

As digital libraries and video databases grow, we need methods to assist us in the synthesis and analysis of digital video. Since the information in video databases can be measured in thousands of gigabytes of uncompressed data, tools for efficient summarizing and indexing of video sequences are indispensable. In this paper, we present a method for effective classification of different types of videos that makes use of a video summarization that is in the form of a storyboard of keyframes. To produce the summarization, we first generate a universal basis on which to project a video frame that effectively reduces any video to the same lighting conditions. Each frame is represented by a compressed chromaticity signature. We then set out a multi-stage hierarchical clustering method to efficiently summarize a video. Finally we classify TV videos using a trained Hidden Markov Model that utilizes the compressed chromaticity signatures as well as temporal features of videos derived from their keyframe summaries.

## 1. Introduction

Video content classification is a necessary tool for efficient access, understanding and retrieval of videos. Different methods have been proposed in the literature for video program classification into predefined categories, e.g. a commercial detection system [1]. One successful study carrying out video classification was performed using a domain method relying on nearest neighbor clustering [2]. The positive aspect of this classification method is its simplicity. Each decision made in the process corresponds to a certain aspect of human visual perception and it is straightforward to understand the rules. However, like most other research work on video classification, [2] did not take advantage of temporal features in video, which form a very powerful cue in understanding video content. Therefore we explore video classification making use of Hidden Markov Models (HMM) that incorporate temporal information along with visual information in video, thus capturing all salient features, *viz.* spatial and color, as well as temporal.

Previously, we successfully set out a novel illumination-invariant color histogram approach that performs good video characterization [3]. In this method we form a vector of chromaticity coefficients for any video frame. The dimensionality experimentally determined for the feature vectors was 12 — we use 12-vectors for clustering and keyframe production. On the basis of these coefficients we produce keyframe-based succinct summarized expressions for video using a multistage hierarchical clustering algorithm [4]. Here we extend this work to provide the capacity to perform semantic content discrimination tasks for video. After video characterization and summarization, we obtain two types of features: (1) chromaticity signatures for keyframes,

each of which represents a scene; (2) temporal features including the *durations of scenes* in a video sequence and *transition characteristics between scenes*. We present a novel method that applies HMM to integrate the two features for video classification. This is motivated by the fact that a certain type of video usually contains a set of frequent scenes that have similar visual information, e.g. in news and basketball games, and also in most situations these types of videos also are characterized by individual approximately stable temporal patterns consisting of scene duration and transition characteristics.

The Hidden Markov Model is a popular technique widely used in pattern recognition [5]. It has a good capability to grasp temporal statistical properties of stochastic processes. The essence of the HMM process is to construct a model that explains the occurrence of observations (symbols) in a time sequence and use it to identify other observation sequences. Some researchers have applied HMM to video analysis and classification. In Nevenka's study [6], HMMs can be formed using face and text trajectories and then can classify a given video into one of four categories of TV programs: news, commercials, sitcoms and soaps. The key point of this approach is that the video content for these types of TV programs have to be satisfactorily characterized by capturing face and text trajectories appearing in the video. Huang et al. [7] built an HMM framework using audio and image features for video classification. Although the use of both audio and visual features can improve classification accuracy, it can make the system complicated and hard to maintain and extend. Also, because the visual features are extracted for every frame, the HMM process needs to carry a great deal of information about the detailed variance between frames, and yet lacks consideration of the entire visual trajectory.

In this paper, we set out a video classification method, based on the Hidden Markov Model, which utilizes the chromaticity signatures of keyframes from summarized video and effectively apprehends the entire temporal feature pattern for different types of videos.

Firstly, we use the illumination-invariant color histogram video characterization method proposed by Drew et al. [3] to produce a 12-vector feature for each frame; secondly we effectively carry out video summarization using a multistage hierarchical clustering, obtaining keyframes. Finally, we perform the video classification task using Hidden Markov Models. In our experiment, we apply our method to the task of classifying television programs into the four categories: news report, commercials, live basketball game, and live football game.

The rest of the papers is organized as follows. Section 2 presents the concept of Hidden Markov Models. Video classification method based on HMMs is proposed in section 3. Experimental results are given in section 4 and in section 5 we present the conclusion.

## **2. A Hidden Markov Model for Video Topic Classification**

This section illustrates the rationale for abstraction of a Markov Chain from a video sequence, and then a Hidden Markov Model is introduced to improve the simple Markov Chain by adding the ability to evaluate an observation sequence. Subsequently, a complete definition of Hidden Markov Model is given.

### **2.1 Markov Chains for Video Topic Categories**

Often we are interested in finding patterns that appear over a space of time. Consider a basketball video sequence, for example. A typical basketball video sequence is assumed to consist of a finite number of scenes such as middle court, left court, right court and close-ups. A

basketball video sequence must therefore undergo a routine such that one of these scenes is entered at a point in time, remains for some duration time, and then makes a transition into another scene. In most instances the duration within each scene is stable according to the basketball game rule. Another important cue in basketball videos is that each scene category has similar color information that is different from other categories.

We consider two problems in regard to the above example:

- Can we generate a general probabilistic pattern in time for basketball videos?
- Given a video sequence, is it a basketball video? Intuitively, if this video fits the basketball video pattern well it may be in the basketball class.

We first attempt to model the process in the time space consisting of a set of scene states. One way to do this is to assume that the state of the model depends on the previous states of the model. This is called the Markov assumption and it simplifies problems greatly.

A Markov Chain is a process that moves from one state to another state depending on the previous  $n$  states. The process is called an order  $n$  model where  $n$  is the number of previous states influencing the choice of the next state. The simplest Markov process is a *first-order* process, where the choice of state is made solely on the basis of the previous state. When considering the basketball videos, the first-order Markov assumption presumes that the current scene can always be predicted solely given the knowledge of the past scene. Here we connect a scene with a state of a Markov chain as shown in Figure 1. For a first order process with  $M$  states, there are  $M^2$  transitions between states since it is possible for any one state to follow another. Associated with each transition is a probability, called the state transition probability —this is the probability of moving from one state to another. These  $M^2$  probabilities may be collected together in a

straightforward way into a *state transition matrix*. Notice that these probabilities do not vary in time —this is an important assumption.

We can now define a first-order Markov process for a basketball video sequence as consisting of:

- States: a set of scenes.
- State transition matrix: the probability of the current state given the previous state.

Any temporal pattern of events occurring in videos that can be described in this manner is a Markov process.

## 2.2 Hidden Markov Models for Video Topic Classification

In some cases the patterns that we wish to find are not described sufficiently by a single Markov process. Returning to the second problem given in Section 2.1, classification by topics, when we have an unknown video sequence we cannot evaluate its video topic with modeled states in an existing Markov Chain. We see that the states of the unknown video sequence are probabilistically related to the states of the basketball video pattern — the pattern and unknown video states are closely linked. In this case we have two sets of states, the observable states (the state of the unknown video) and the hidden states (the state of the modeled basketball video pattern). We wish to identify the video topic based on the sequence of scenes of the unknown video and the Markov process without actually ever seeing the underlying video topic. Here, we model such processes using a Hidden Markov Model in which we call the underlying Markov process a *Hidden Markov process*, call the states in the hidden Markov process *hidden states* and call the unknown video sequence a *observation state (symbol) sequence*.

It is important to note that the number of states in the modeled Markov process and the number of scenes of an unknown video may be different. In a basketball video system with a finite set of states, it may be possible that those states are repeated a thousand times in an unknown video. The key in such cases is that in an HMM system there is an underlying hidden Markov process changing over time, and a set of observable states which are probabilistically related to the hidden states.

The diagram in Figure 1 shows the hidden and observable states in the basketball video example. It is assumed that the hidden states (the basketball video pattern) have been obtained by training a basketball video set and modeling states as a first-order Markov process so that these states are all connected to each other. In this diagram, three blue rectangles on the bottom part stand for the assumed three hidden states in the basketball video Markov process. The video sequence on the top part represents an observation state sequence. The connections between the hidden states and the observable states represent the probability of generating a particular observed state given that the Markov process is in a particular hidden state.

So, in addition to the probabilities defining the Markov process, we need another probability matrix, termed the *observation symbol probability matrix*, or *confusion matrix*, which contains the probabilities of occurrence of the observable states given a particular hidden state.

With the above modeling scheme, we arrive at the definition of a Hidden Markov Model set out in the following section.

### **2.3 Definition of Hidden Markov Model**

In an HMM, there are a finite number of states, each of which is associated with a (generally multidimensional) transition probability distribution. The HMM is always in one of these states.

At each clock time, the system enters a state based on a transition probability depending on the previous state. After this transition is made, an output observable symbol is generated based on an observation probability distribution, depending on the current state. It is only the output symbols, not the states that are visible to an external observer and therefore states are “hidden” to the outside; hence the name Hidden Markov Model. The elements of an HMM are:

- A set of  $N$  states,  $s = \{s_1, s_2, \dots, s_N\}$ , with the state at time  $t$  denoted by  $q_t \in s$ .
- The initial state probability distribution,  $\Pi = \{\pi_i\}$ , where

$$\pi_i = P[q_1 = s_i], \quad 1 \leq i \leq N$$

- The state transition probability matrix,  $A = \{a_{ij}\}$ , where

$$a_{ij} = P[q_t = s_j | q_{t-1} = s_i], \quad 1 \leq i, j \leq N,$$

with  $0 \leq a_{ij} \leq 1$  and the constraint that

$$\sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N$$

- The observation symbol probability for the observation,  $B = \{b_j(O_t)\}$ , where  $b_j(O_t)$  is the probability of observation  $O_t$  at time  $t$  given that the state is  $q_t = s_j$ ,

$$b_j(O_t) = P(O_t | q_t = s_j).$$

In a discrete HMM, the observation symbol probability is defined as:

$$b_j(O_t) = P[O_t = v_k | q_t = s_j], \quad 1 \leq j \leq N,$$

where  $V = \{v_1, v_2, \dots, v_M\}$  is the set of all possible observation symbols, and  $M$  is the number of different observation symbols.

Using a shorthand notation, an HMM is defined as the triplet

$$\lambda = (A, B, \Pi)$$

### 3. Incorporating Hidden Markov Models into Video Topic Classification

This section provides a complete scheme to establish a Hidden Markov Model based classification system by which a given video can be classified into one of several topic categories. While the idea of using HMMs for video classification has been used before [8], the use made here of keyframe summaries and videos that are notionally replaced by their summary versions [9] is a good deal more efficient and, indeed, demonstrably very effective. We first discuss how to deploy our Hidden Markov Model for video topic classification, and then we focus on the extraction of state vectors from video images. Two phases of the classification task: training and classification are described at the end.

#### 3.1 Deployment of Hidden Markov Models

Since we suggest applying Hidden Markov Models for video topic classification, the question of the model deployment naturally arises. Before such models can be built, we must specialize the HMM definition for this application.

For the application of HMM to video topic classification, we build an HMM for each topic category; each HMM element is briefly introduced in the following and as well more details on how to produce it will be given in later sections.

- Hidden states, denoted by  $S = \{s_1, s_2, \dots, s_N\}$ . Via training a collection of videos belonging to a topic category, a set of hidden states for this topic can be produced. Here each

hidden state in an HMM, which a 12-component vector, is associated with a typical keyframe extracted from the training set of videos.

- Observable states: denoted by  $V = \{v_1, v_2, \dots, v_M\}$ . In our HMM application, the sequence of observable states is a set of 12-component vectors for keyframes extracted from a given target video. Here, we define a *TSV* video sequence (Temporal and keyframe-based Summarized Video) using the 12-component keyframe vectors, which provides an efficient and effective representation for the given target video.
- State transition probability matrix:  $A = \{a_{ij}\}$ , which holds the probability of a hidden state given the previous hidden state. In our application, these probabilities characterize the temporal relationship between hidden states for a topic category of videos.
- Observation symbol probability matrix:  $B = \{b_j(O_i)\}$  is the set of all possible observable states. This matrix is also termed the confusion matrix, and contains the probability of observing a particular observable symbol at a time point, given that the model is in a particular hidden state at that time. In our application, the *usual observation probability is extended to express visual similarities between hidden states and observation states* (see section 3.4.2).
- Initial state distribution  $\pi_i = P[q_1 = s_i]$ , which contains the probability of the model being in a particular hidden state at the start point for an HMM. Clearly, this distribution means the probability of each hidden state occurring at the *beginning* in any video of a certain topic category. In our application, we use the probability of each state continuing itself as its initial distribution, that is  $\{a_{ii}\}$  in the state transition probability matrix  $A$ .

## 3.2 State Vector

We had developed a new low-dimensional video frame feature that is more insensitive to lighting change, motivated by color constancy work in physics-based vision, and applied the feature to keyframe production using hierarchical clustering [4]. The main point vis-à-vis video summarization is that any video is effectively moved into the same lighting environment, making it meaningful to project video features onto a *precomputed* universal basis set.

Lighting is first discounted by normalization of color-channel bands [3] . This step approximately but effectively removes dependence on both luminance and lighting color. Then image frames are moved into a chromaticity color space  $\{r,g\} = \{R,G\}/(R+G+B)$ . As well as reducing the dimensionality of color from 3 to 2 this also has the effect of removing shading. In order to make the method fairly robust to camera and object motion, and displacements, rotations, and scaling, we go over to a 2D histogram derived from DC components of frames. Chromaticity histograms are then compressed — i.e., we *treat the histograms as images*. Here, we use a wavelet-based compression [3] because this tends to strike a balance between simple low-pass filtering and retaining important details. The scaling function of biorthonormal wavelets, as a symmetrical low-pass filter, can be exploited to that end [4]. Starting with 128x128 histograms and using a 3-level wavelet compression we arrive at 16x16 histograms.

However, we found that compression of histograms could be improved if the histograms are first binarized, i.e., entries are replaced with 1 or 0. The rationale for this step is that chromaticity histograms are a kind of color signature for an image, similar to a palette. In work involving recovering the most plausible illuminant from pixel values in an image [10] it was found beneficial to utilize this kind of color signature. Here, the step of binarizing the histogram not only reduces the computational burden, since true chromaticity histograms need not be

computed, but also has the effect of producing far fewer negatives in the compressed histogram. Finally, we found that one further step could substantially improve the energy compaction of the representation: we carry out a  $16 \times 16$  Discrete Cosine Transform (DCT) on the compressed  $16 \times 16$  histogram. After zigzag ordering, we keep 21 DCT coefficients.

Since every image now lives in approximately the same lighting, we can in fact precompute a basis for the DCT 21-vectors, offline, which can then be reused for any new image or video. Here we determine a basis set by the Singular Value Decomposition (SVD) of the DCT 21-vectors. We found that 12 components in the new basis represent the entire DCT vector very well, based on the variance-accounted-for. As well, we found that energy compaction worked better using a spherical chromaticity  $\{r, g\} = \{R, G\} / \sqrt{R^2 + G^2 + B^2}$ , rather than the usual linear one  $\{r, g\} = \{R, G\} / (R + G + B)$ . Thus the method we set out here is to precompute a set of basis vectors, once and for all, and then form the 12-vector coefficients for any video frame with respect to this basis. So keyframe extraction can be carried out very efficiently, using only 12-component vectors.

A keyframe is extracted from each of the segmented scenes in a video. We use a hierarchical clustering scheme to segment a video into a sequence of scenes [4]. This method executes a bottom-up multi-level temporal merging process, where only adjacent frames or frame groups are merged, by calculating their vectors'  $L_2$  distance. Note that the temporal order is maintained throughout. A threshold based on variance within the cluster, compared to variance for the parent node, is assigned for determining the final clusters, and each of the final clusters is taken to correspond to a scene. Finally, a keyframe is extracted as the medoid of each cluster.

### 3.3 Hidden Markov Model Training

After keyframes are obtained as explained in the previous section, all the keyframe vectors from one topic class of videos are used to train an HMM for the topic. The process of training essentially involves creating a set of hidden states, finding the state transition probability matrix, and the initial probability distribution for the HMM. The upper part in Figure 2 shows the training phase for the HMM for basketball game videos.

#### 3.3.1 Finding Hidden States

We employ a clustering method called CLARANS [11] on the set of keyframe vectors to find the hidden states for an HMM.

For a general HMM, the Viterbi algorithm is normally used to find the most probable sequence of hidden states and the Baum-Welch algorithm is used for parameter estimation, given a sequence of observed states [5]. Those hidden states must be invariable and finite. However, in our video model, keyframes can be different from each other even though they represent a same scene. That is, a hidden state can correspond to different keyframes in a video sequence. For this reason, we use a clustering method to discover hidden states instead of Viterbi and Baum-Welch algorithms. Intuitively, the use of a clustering method to find states from a set of video keyframes is a good choice since the clustering process's ability to find structures and groups from the given data is satisfactory for the task of hidden states generation. However, clustering on a large set of multi-dimension data poses a challenge for an algorithm's efficiency. To address the efficiency concern, we choose the CLARANS (Clustering Large Applications based on RANdomized Search) clustering algorithm from spatial data mining, for the following reasons. First, CLARANS is based on k-medoid clustering. Unlike other partitioning methods, k-medoid based algorithms are very robust to the existence of outliers (i.e., data points that are very far away from the rest of the data points). Also, clusters found by a k-medoid based algorithm do

not depend on the order in which the objects are examined and are invariant with respect to translations and orthogonal transformations of data points. Second, CLARANS is effective for our 12-vector coefficients clustering, for which natural notions of similarity is Euclidean distance. Furthermore, CLARANS, which originally was developed for spatial data clustering, has been recognized as working efficiently on large data sets.

To find  $k$  clusters, the CLARANS approach is to determine a representative object for each cluster. This representative object, called a medoid, is meant to be the most centrally located object within the cluster. Once the medoids have been selected, each non-selected object is grouped with the medoid to which it is the most similar. More precisely, if  $O_j$  is a non-selected object, and  $O_i$  is a (selected) medoid, we say that  $O_j$  belongs to the cluster represented by  $O_i$ , if  $d(O_j, O_i) = \min_{O_e} d(O_j, O_e)$ , where the notation  $\min_{O_e}$  denotes the minimum over all medoids  $O_e$ , and the notation  $d(O_a, O_b)$  denotes the dissimilarity or distance between objects  $O_a$  and  $O_b$ . Finally, the quality of a clustering (i.e., the combined quality of the chosen medoids) is measured by the average dissimilarity between an object and the medoid of its cluster.

CLARANS begins with an arbitrary selection of  $k$  objects. In each step, a swap between a selected object  $O_i$  and a non-selected object  $O_h$  is made, as long as such a swap would result in an improvement of the quality of the clustering.

Let us consider a graph abstraction of this algorithm. In a graph including  $n$  objects, a node is represented by a set of  $k$  objects  $\{O_{m1}, O_{m2}, \dots, O_{mk}\}$  that are the selected medoids. The set of nodes in the graph is the set:  $\{\{O_{m1}, O_{m2}, \dots, O_{mk}\} \mid O_{m1}, O_{m2}, \dots, O_{mk} \text{ are objects in the data set}\}$ .

Two nodes are neighbors (i.e., connected by an arc) if their sets differ by only one object. More formally, two nodes  $S_1 = \{O_{m1}, O_{m2}, \dots, O_{mk}\}$  and  $S_2 = \{O_{w1}, O_{w2}, \dots, O_{wk}\}$  are neighbors if

and only if the cardinality of the intersection of  $S_1$  and  $S_2$  is  $k-1$ , i.e.  $|S_1 \cap S_2| = k - 1$ . It is easy to see that each node has  $k(n - k)$  neighbors. Since a node represents a collection of  $k$  medoids, each node corresponds to a clustering. Thus, each node can be assigned a cost that is defined to be the total dissimilarity between every object and the medoid of its cluster. Two neighboring nodes swap in each step, and this is equivalent to swapping between a selected object and a non-selected object which will lead to a decrease in the total cost.

### 3.3.2 Calculating State Transition Probabilities

The state transition probability matrix expresses the probability of moving from one hidden state to another. There are at most  $M^2$  transitions among the hidden states (with  $M$  is the number of states) since it is possible for any one state to follow another, or itself. From the above clustering method, we know that the states are the medoids of clusters of video keyframes, and thus each state corresponds to a typical scene. A video topic model would be described at any time as being in one of these states. Generally, the states are interconnected to each other in such a way that any state can be reached from any state (i.e., an ergodic model) at regularly spaced discrete times.

Consider the following example of a video topic training experiment.

**Example:** suppose we have a topic category training set consists of a 30-frame-long video sequence:  $V$ . After scene segmentation, a set of scenes and their keyframes are obtained as follows:

$$V : \{kf_1(5), kf_2(5), kf_3(3), kf_4(2), kf_5(5), kf_6(7), kf_7(3)\},$$

where  $kf\#s$  denote the keyframes of the scenes obtained, with the number in brackets standing for the number of frames that the front keyframe represents.

Via CLARANS clustering, these keyframes are grouped into 3 clusters illustrated in Figure 3, i.e., 3 states. Given this scenario, a Hidden Markov Model with three states has a state transition probability matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

where  $a_{ij}$  denotes the transition probability from state  $i$  to state  $j$ , and state transition coefficients have the properties:

$$a_{ij} \geq 0$$

$$\sum_{j=1}^N a_{ij} = 1$$

Considering Figure 3, we find there are three scenes falling into cluster  $S_1$ ; their keyframes are  $kf_1$ ,  $kf_3$  and  $kf_6$  respectively. The scene represented by keyframe  $kf_1$  in state  $S_1$  is in play for 5 frames and transits into the scene represented by  $kf_2$  in  $S_2$ . The  $kf_3$  scene is in play for 3 frames and enters state  $S_2$ . The  $kf_6$  scene is in play for 7 frames and enters state  $S_3$ . Also, it is easily calculated that the total number of frames entering into state  $S_1$  is  $5+3+7=15$ . Thus 2 out of 15 frames in state  $S_1$  transit into  $S_2$ , 1 of 15 frames transits into  $S_3$ , and the remaining 12 of 15 frames transit to frames of the same state  $S_1$  — and means the state stays in the state  $S_1$ . Hence, it is straightforward to conclude that state  $S_1$  has probability  $2/15$  of transiting to  $S_2$ ,  $1/15$  to transit to  $S_3$ , and  $12/15$  to continue staying in itself — thus  $a_{11} = 80\%$ ,  $a_{12} = 13\%$  and  $a_{13} = 7\%$ .

To state this scheme more formally, we define the *state transition probability* as follows:

$$a_{ij} = \frac{\text{the\_number\_of\_keyframes\_in\_state\_i\_followed\_by\_keyframes\_in\_state\_j}}{\text{total\_number\_of\_frames\_in\_state\_i}}, \quad i \neq j$$

and for any state  $i$ ,

$$a_{ii} = 1 - \sum_j a_{ij}, \quad i \neq j$$

In the light of Figure 3,  $a_{ij}$  actually refers to the ratio of the number of arrows contained in a state going to any other state to the total number of frames represented by the state.

### 3.3.3 Determining initial state probabilities

Initial state probabilities capture the likelihood for each state being at an initial point in the process of an HMM. In our video case, the temporal structure may be repeated in the video sequence. For example, in a news report, a live report usually comes after the anchorperson's introduction and then the video returns to the studio report. This kind of pattern can be repeated several times in a single news program. Another obvious example occurs in sports games such as a football game, in which there are a lot of cycles of attacks and pauses. Such temporal structures in videos require us to use an ergodic HMM, where each state can be reached from other states and can be revisited after leaving.

For an ergodic HMM, the initial state distribution of one state is also the probability of occurrence of that state in the model— i.e., the  $a_{ii}$  probability in the state transition probability matrix. Before these diagonal numbers can be used as initial state probabilities, however, they first need to be normalized. The normalized probabilities can be simply expressed as

$$\pi_i = P[q_1 = s_i] = \frac{a_{ii}}{\sum_i^N a_{ii}}, \quad 1 \leq i \leq N$$

## 3.4 Hidden Markov Model Classification

Via training, we can obtain a number of HMMs each of which corresponds to a video topic class of interest. These HMMs together form an HMM based classifier. The bottom part in Figure 2 shows a procedure for video topic classification. We begin with a given unknown video. An observation sequence is first generated from the given video, as in Section 3.4.1. In Section 3.4.2, we build an observable state probability matrix for each HMM, consisting of the visual similarity between hidden states and observable states. Then the observation sequence is fed into each HMM, and an evaluation algorithm is employed to compute the probability for each HMM given this sequence. In Section 3.4.3, we present and discuss such an evaluation algorithm.

### 3.4.1 Building Observable State Sequence

Given a test video sequence, a sequence of 12-component vectors of keyframes is first generated via the process of compressing the illumination invariant chromaticity histograms of video frames, segmenting the video into scenes, and extracting a keyframe from each scene, as set out in Section 3.2. Chromaticity vectors for keyframes are taken as observable states of HMMs. For HMM classification, a sequence of observable states which is fed into each HMM should contain not only chromaticity feature but also temporal feature of the video. Here we define a *temporal and keyframe-based summarized video* (TSV for short) to form the observable state sequence as following:

- The TSV is a sequence of video keyframe vectors ordered by time;
- The TSV notionally *repeats each keyframe vector a number of times equaling the number of frames of the scene* from which the keyframe was extracted.

**Example:** Suppose we have a given video frame sequence from time  $t=1$  to  $t=10$  as follows:

Frame =  $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}\}$

After scene segmentation, we obtain a sequence of keyframes as

Keyframe =  $\{kf_1(f_1, f_2, f_3), kf_2(f_4, f_5, f_6, f_7), kf_3(f_8, f_9, f_{10})\}$

Thus, the TSV for the given video is

$TSV = \{kf_1, kf_1, kf_1, kf_2, kf_2, kf_2, kf_2, kf_3, kf_3, kf_3\}$ .

Using the *TSV*, both the temporal feature and chromaticity feature can be maintained in the observable state sequence. To classify a given video, its *TSV*, as the observable state sequence, will be fed into each HMM.

### 3.4.2 Building Observable State Probability Matrix

To begin with, we need to build an observation symbol probability matrix for each HMM, containing the probabilities of observable symbols given a particular state. We have considered the original definition of a Hidden Markov Model in which *finite* observable symbols are modeled as the output of an HMM, and each state in the model is characterized by a probability distribution of all the possible observable symbols. The probabilities of the finite observable symbols are, in general, obtained by training a known data set.

But, in our special case, it is difficult to reveal the mechanism of matching the observation to the states by using this regular observation probability. In our video model, the hidden states refer to typical scene keyframes for a video topic category, and observable symbols refer to the sequence of keyframes in a given video. It should be noted that the relation between the two kinds of keyframes is the likelihood of matching, but not “output”. It also should be clear that in this video case the video keyframes as observable symbols are in an *infinite* set, and therefore there is no way to train the probabilities in advance.

We can, however, *extend* the original Hidden Markov Model to handle the video case where the observation probabilities simply express the likelihood of matching visual features between observation frames and state frames, i.e., the visual similarity or distance between the 12-coefficient vectors for frames.

We consider the keyframes from the given video as the observation symbols, and the probability of any observable symbol given a particular state is computed by the *Inverse Euclidean Distance* between the observable symbol vector and the hidden state vector. Hence the more similarity between vectors, the more probability between the symbol and state. *Inverse Euclidean Distance* is defined as the normalized Euclidean distance value, subtracted from 1:

$$\text{Inverse Euclidean Distance } (O, S) = 1 - \frac{\text{Euclidean dis.}(O, S)}{\sum_i \text{Euclidean dis.}(O, S_i)}$$

where  $O$  is an observed state vector at the current time, and  $S$  is any hidden state vector.

Before these inverse Euclidean distances can be used as observation probabilities, they first need to be normalized. Since the sum of the inverse distances of any observation symbol to all hidden states is  $N-1$ , where  $N$  is number of states, the normalized inverse Euclidean distance can be expressed as:

$$\text{Normalized Inverse Euclidean Distance } (O, S) = \frac{\text{Inverse Euclidean Dis. } (O, S)}{N - 1}$$

Note, however, that we emphasize that this usage of visual distance as observation probability does not conform to the definition in the original HMM. Even though the results of our experiment verify its suitability, this point still remains to be discussed and proved further.

### 3.4.3 Computing the Probability of an Observed State Sequence

We wish to calculate the probability of the observation sequence,  $O = O_1O_2\dots O_T$ , given the model  $\lambda = (\pi, A, B)$ , i.e.,  $\Pr(O | \lambda)$ . Consider the state sequence  $Q = q_1q_2\dots q_T$  where  $T$  is the length of the state sequence (the number of observations). We choose the so-called *Forward algorithm* [5], which had been proved efficient, to calculate this probability.

The *Forward algorithm* considers the partial probability

$$\partial_t(i) = \Pr(O_1O_2\dots O_t, q_t = S_i | \lambda),$$

i.e., the probability of the partial observation sequence  $O_1 O_2 \dots O_t$ , (until time  $t$ ) and state  $S_i$  at time  $t$ , given the model  $\lambda$ . We can solve for  $\partial_t(i)$  inductively, as follows:

- Initialization:

$$\alpha_1(i) = \pi(i)b_i(O_1), \quad 1 \leq i \leq N. \quad (1)$$

- Induction:

$$\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_{i=1}^n \alpha_t(i) a_{ij}, \quad 1 \leq t \leq T - 1 \text{ and } 1 \leq j \leq N \quad (2)$$

- Termination:

$$\Pr(O) = \sum_{j=1}^n \alpha_T(j) \quad (3)$$

(1) initializes the partial probability as the joint probability of state  $S_i$  and initial observation  $O_1$ . The induction step, which is the heart of the Forward algorithm, is illustrated in Figure 4. This figure shows how state  $S_j$  can be reached at time  $t+1$  from the  $N$  possible states,  $S_i$ ,  $1 \leq i \leq N$ , at time  $t$ . Since  $\partial_t(i)$  is the probability of the joint event that  $O_1 O_2 \dots O_t$  are observed, and the state at time  $t$  is  $S_i$ , the product  $\partial_t(i) \cdot a_{ij}$  is then the probability of the joint event that  $O_1$

$O_2 \dots O_t$  are observed, and state  $S_j$  is reached at time  $t+1$  via state  $S_i$  at time  $t$ . Summing this product over all the  $N$  possible states  $S_i$ ,  $1 \leq i \leq N$  at time  $t$  results in the probability of  $S_j$  at time  $t+1$  with all the accompanying previous partial observations. Once this is done and  $S_j$  is known, it is easy to see that  $\partial_{t+1}(j)$  is obtained by accounting for observation  $O_{t+1}$  in state  $j$ , i.e., by multiplying the summed quantity by the probability  $b_j(O_{t+1})$ . Then computation of (2) is performed for all states  $j$ ,  $1 \leq j \leq N$ , for a given  $t$ ; the computation is then iterated for  $t = 1, 2, \dots, T-1$ . Finally, (3) gives the desired calculation of  $\Pr(O | \lambda)$  as the sum of the terminal partial probability  $\partial_T(i)$ . This is the case since, by definition,

$$\partial_T(i) = \Pr(O_1 O_2 \dots O_T, q_T = S_i | \lambda),$$

and hence  $\Pr(O | \lambda)$  is just the sum of the  $\partial_T(i)$ 's.

The partial probability calculation is, in effect, based upon the lattice (or trellis) structure shown in Figure 5. The key is that since there are only  $N$  states (nodes at each time slot in the lattice), all the possible state sequences will re-merge into these  $N$  nodes, no matter how long the observation sequence. At time  $t=1$  (the first time slot in the lattice), we need to calculate values of  $\partial_1(i)$ ,  $1 \leq i \leq N$ . At times  $t = 2, 3, \dots, T$ , we only need to calculate values of  $\partial_t(j)$ ,  $1 \leq j \leq N$ , where each calculation involves only  $N$  previous values of  $\partial_{t-1}(i)$  because each of the  $N$  grid points is reached from the same  $N$  grid points at the previous time slot.

Given this *Forward algorithm*, it is straightforward to determine which of a number of video topic HMMs best describes an observation *TSV* sequence—the forward algorithm is evaluated for each, and that giving the highest probability is selected.

## 4. Experimental Results

We evaluate our HMM based classification method by classifying four types of TV program. They are news reports, commercials, live basketball games and live football games.

To set up our video data set, we collected 30 TV programs of 5 minutes duration each from broadcast TV as the training set for each video category. The collection spans different channels: NBC for basketball games, FOX for football games, CNN for news and a number of channels for commercials. Another set of 50 TV programs of the same length for each category recorded from the same channels is used as the testing set.

These TV programs were recorded on Super VHS format from cable television and digitized in MPEG1 format at 1.5 MBps (30 frames/sec). We assume that the input TV programs always belong to one of the four categories of TV programs.

In our experiment, we use two metrics to gauge the performance of our model. Their definitions are given as follows:

**Precision:** For a video topic class of interest, *Precision* is the ratio of correct classification for test videos made into this class over all the classifications made into this class. (i.e., “correct responses”).

$$precision(C) = \frac{|\{correct\} \cap \{classified\}|}{|\{classified\}|}, \text{ where } C \text{ is the target class.}$$

**Recall:** For a video topic class of interest, *Recall* is the ratio of correct classifications for test videos made to this class over all test videos correctly belonging to this class.

$$recall(C) = \frac{|\{correct\} \cap \{classified\}|}{|\{correct\}|}, \text{ where } C \text{ is the target class.}$$

In above definitions,  $\{correct\}$  denotes the set of test videos belonging to the class  $C$ ;  $\{classified\}$  denotes the set of test videos classified into the class  $C$  by our model.

The overall HMM classification results for four video topic categories are reported in Table 1. We show the Recall and Precision performance for HMMs in Table 2 and Table 3. Figures 6, 7, 8 and 9 illustrate the state transition probabilities, using 3-D charts for each of the HMMs: news, basketball game, commercial, and football game.

Overalls results yield an average 80% correct classification, thus showing that correct classification is possible in the majority of test videos.

Table 2 shows Recall values for the four HMMs that occur on the diagonal in Table 1. Comparing these four HMMs contained in our video topic classifier, we see that classification Recall performance varies. We obtained best classification results on videos of football and basketball topics. Both of their HMMs generate a classification Recall of about 90 percent over all relevant videos. Results are good because chromaticity features are frequent in these videos; for example, most frames in football videos contain green ground. Also the scenes' temporal transition pattern in football videos is prominent, due to the sport's game rules. The news HMM also achieves 80 percent for classification Recall. We can see that correct classifications should essentially be attributed to the occurring pattern of anchorpersons that are frequently present in the news videos. During the experiment, we find that most of the successfully classified football videos obtain much higher probabilities from the football HMM than those from other three models, but for basketball and news videos the differences of the probabilities from different HMMs are less considerable.

Contrasted to the good classification results achieved on sports and news, somewhat less successful results were achieved on the commercials HMM, but however results were still

impressive. In Table 2, the commercials HMM achieved a Recall of 60 percent. We also find that for most correctly classified commercial videos, the probabilities from other HMMs are very close to the probabilities from the commercials HMM. An important reason for this failure is due to the limitations of the state vectors used. These vectors only focus on the extraction of chromaticity features, and may be incapable of revealing the temporal pattern for commercial videos.

Table 3 shows the Precision value for each of the four HMMs. In our particular framework, the Precision measure indicates the interaction between HMMs, since it encapsulates the number of right and wrongly classified videos identified by one HMM. In this Table, we note that the football HMM and news HMM have relatively low precision values. Obviously, they become confused because of competition from the commercials HMM. Of the commercial videos, 14% are misclassified into news and 18% into basketball. Considering this observation, one can conclude that the commercials topic category suffers from the fact that often the commercial videos contain a large variety of chromaticity information with less evident temporal pattern present, so that the commercials HMM cannot generate considerably higher probabilities for commercial videos than other HMMs.

Figures 6-9 symbolize transition probabilities via 3-D graphs. The diagonal values on these figures indicate the probability for each state remaining itself. The off-diagonal values correspond to the probabilities of transitions among states. We see that diagonal values are much higher than the rest. That is reasonable because states often remain themselves for a period of time and then transition to some other state. In contrast, some values have very low probabilities, and we can think of these states as representing semantic scenes that have no temporally consecutive relationship with each other. These factors determine the variations in 3-D line

shapes observed. Considering the graph for commercials videos in Figure 8, for example, we note that the commercials class has a relatively low height for the diagonal points on the state plane but a high height for most other points, in comparison with the other three topics. This is caused by often-occurring short scene durations in commercials videos and frequent transitions to other scenes. The different line shapes for different topic HMMs highlights the advantage of using a temporal feature to identify video topics.

## 5 Conclusion

The primary goal of this work is to consider the use of an HMM for video topic classification. The usage of an HMM for video temporal modeling is justified both by our extended definition of the HMM and by the structure of video sequences. As a result of the theoretical aspects discussed in this work and based on our experimental results, the following conclusions are drawn:

- The Markov Chain approach has been extended to model a video sequence. It provides a way to effectively take advantage of the temporal feature in video data. One new aspect in our use of a Markov Chain model is to consider video scene segmentation where the states of the model are determined on the basis of video scenes.
- A new Hidden Markov Model based video classification system has been successfully developed. The HMM uses chromaticity 12-component vectors as the states, so that the Forward Algorithm that is used to compute the similarity between the observation sequence and a hidden state sequence can be executed quite efficiently.
- A uniform video framework to seamlessly integrate visual and temporal features has been presented. It can efficiently be used to model video sequences. It is flexible, and the

chromaticity vector visual feature used in our classification case could easily be replaced by other visual features, e.g. motion, texture or content-based objects etc. Hence, the method can potentially be extended to other video topics.

## 6 Future Work

Although the classification algorithm based on Hidden Markov Model works well in our experiment, there are certain limitations and future improvements are possible. Since video includes various visual features, we plan to explore the issue of HMM classification in terms of other attributes such as object information and investigate methods to extract a temporal feature for those attributes, for HMM processing.

## References:

- [1] Hauptmann, A. G., Witbrock, M. J. (1998) Story segmentation and detection of commercials in broadcast news video. Proceedings of Advances in Digital Libraries Conference, Santa Barbara, CA.
- [2] Zhou, W.-S., Vellaikal, A., Kuo, C. C. J. (2000) Rule-based video classification system for basketball video indexing. Proceedings on ACM multimedia 2000 workshops, pp. 213 – 216.
- [3] Drew, M. S., Wei, J., Li, Z. N. (1999) Illumination-invariant image retrieval and video segmentation. Pattern Recognition, 32:1369-1388.
- [4] Drew, M. S., Au, J. (2000) Video keyframe production by efficient clustering of compressed chromaticity signatures. ACM Multimedia, pp.365-368.

- [5] Rabiner, L. R., Juang, B. H., (1986) A tutorial on Hidden Markov Models. *IEEE ASSP Magazine*. pp. 4-15.
- [6] Wei, G., Agnihotri, L., Dimitrova, N. (2000) TV program classification based on face and text Processing. IEEE multimedia and Expo 2000, New York.
- [7] Huang, J., Liu, Z., Wang, Y., Chen, Y., Wong, E. K. (1999) Integration of multimodal features for video classification based on HMM. 1999 IEEE Third Workshop on Multimedia Signal Processing, Copenhagen, Denmark, pp. 53 – 58.
- [8] Huang, J., Liu, Z., and Wang, Y., Joint Video Scene Segmentation and Classification based on Hidden Markov Model, IEEE Int. Conf. on Multimedia and Expo (ICME2000), New York, NY, August 2000, Vol. 3 pp. 1551 –1554.
- [9] Lu, C., Drew, M.S., and Au, J., Classification of Summarized Video by using Hidden Markov Models on Compressed Chromaticity Signatures, ANNIE'01: Artificial Neural Networks In Engineering, pp. 645-650, St. Louis, Missouri, November 4th-7th, 2001.
- [10] Finlayson, G. D., Hubel, P. M., and Hordley, S. (1997) Colour by correlation, Fifth Color Image Conference, pp 6-11.
- [11] Ng, R., Han, J. (1994) Efficient and effective clustering method for spatial data mining. Proceeding of 1994 Int'l Conf. on Very Large Data Bases, Santiago, Chile, pp. 144-155.

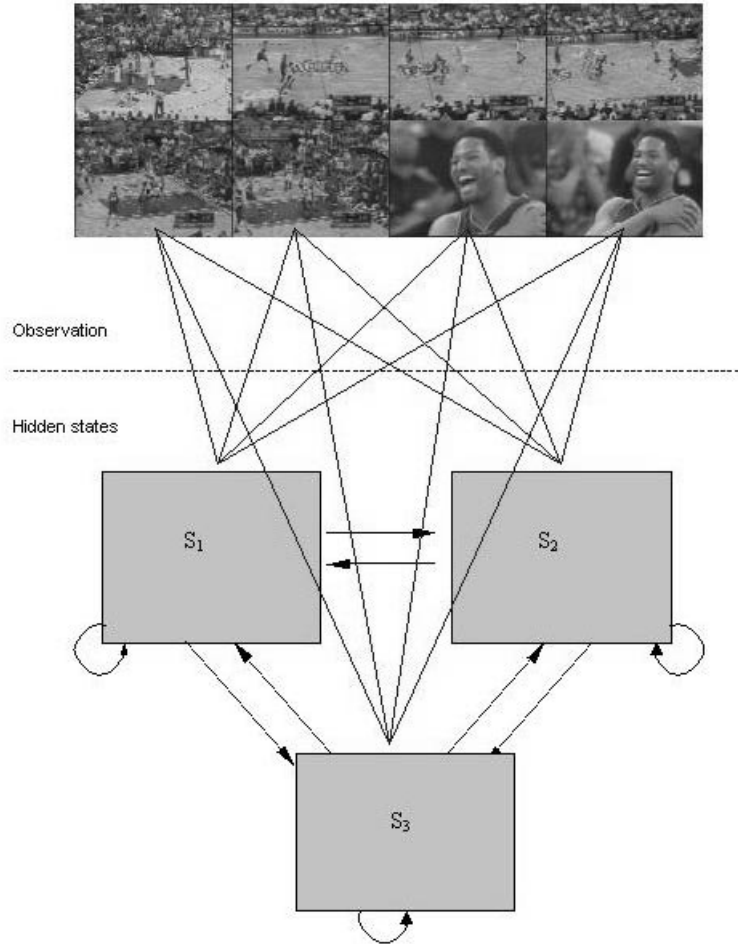


Figure 1: Dependency graph of hidden states and observation sequence where blue lines mean the relation between observable states above and below hidden states below, and arrows denote transitions among the hidden states.

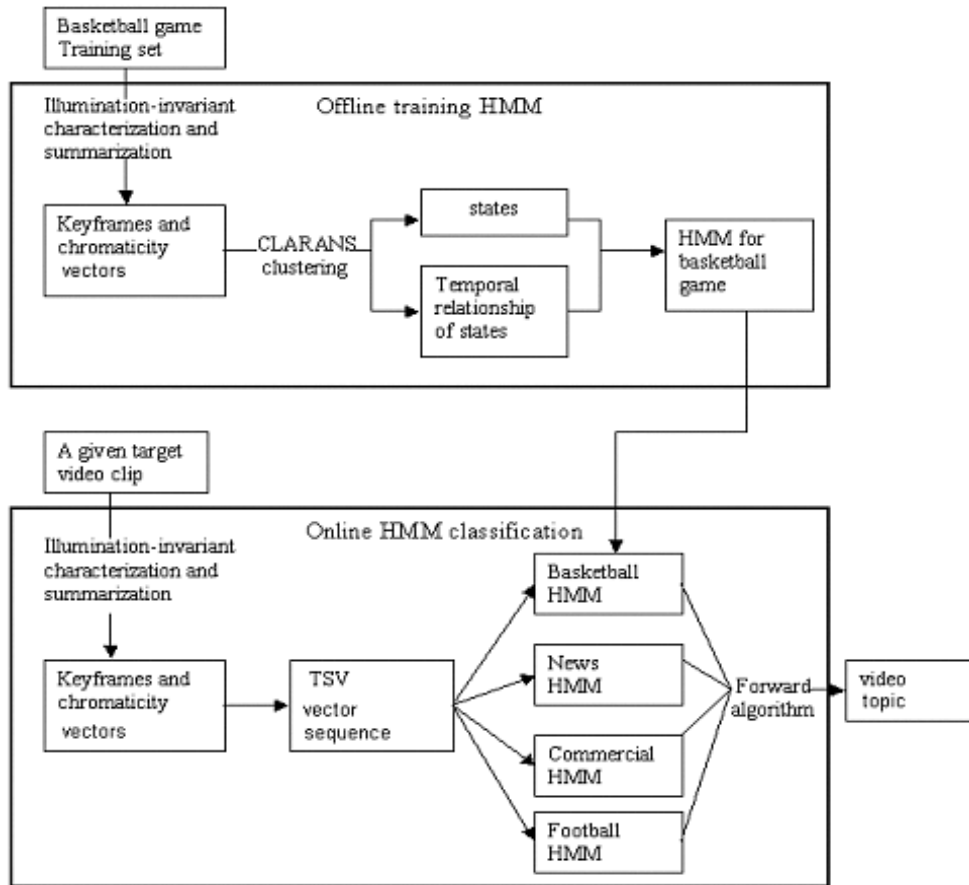


Figure 2: A Hidden Markov Model classification system

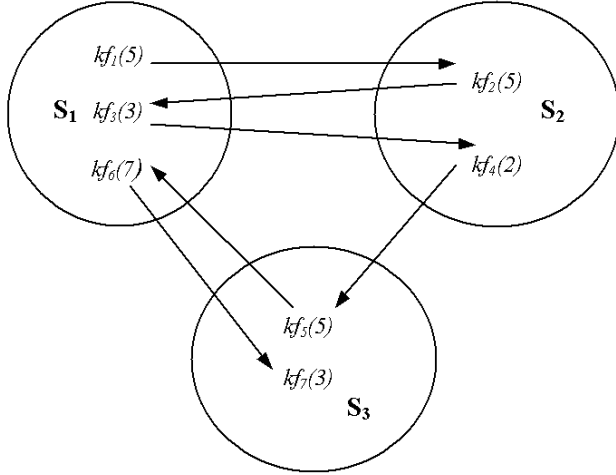


Figure 3: Illustration of three clusters of keyframes corresponding to three states:  $S_1$ ,  $S_2$  and  $S_3$ .

Values labeled  $kf$  denote the keyframes of the scenes obtained, with the number in brackets standing for the number of frames represented by the keyframe for that scene, and the subscript labeling which scene this is. For example,  $kf_6(7)$  means the keyframe for scene number 6, which stands for a total of 7 frames in that scene. The arrows connecting keyframes stand for transitions between corresponding scenes.

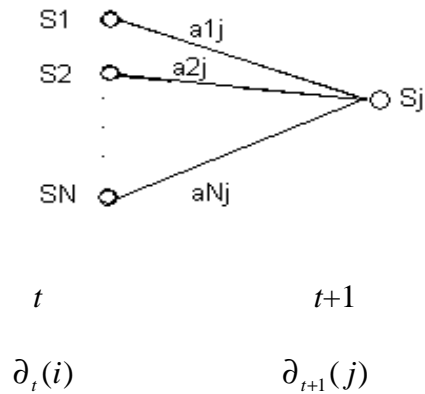


Figure 4: Illustration of the sequence of operations required for the computation of the partial probability  $\partial_{t+1}(j)$ .

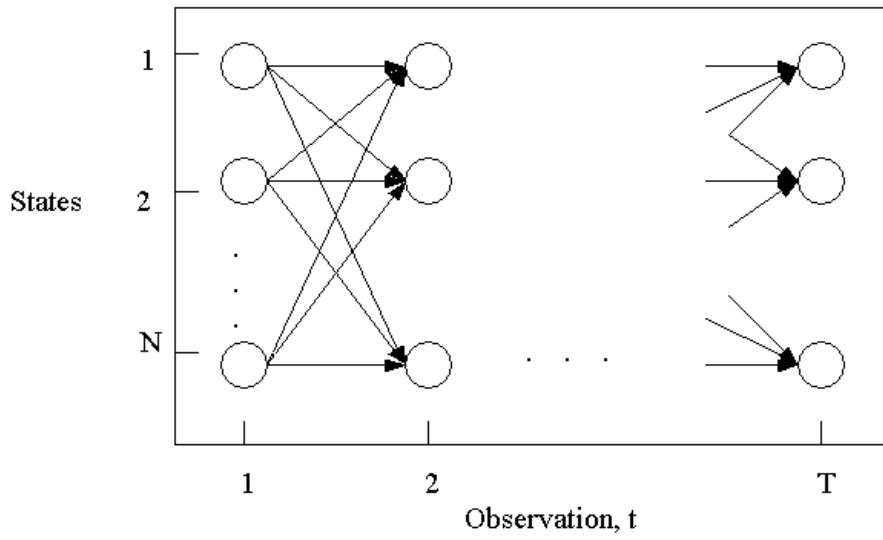


Figure 5: Implementation of the computation of  $\partial_t(i)$  in terms of a lattice of observations  $t$ , and states  $i$ .

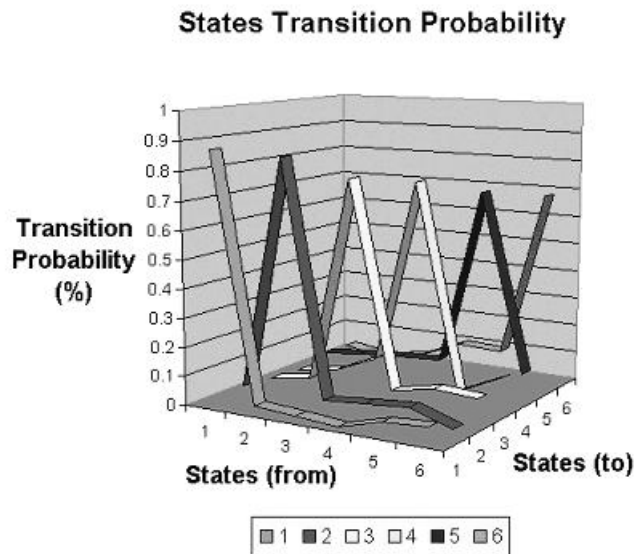


Figure 6: State transition probabilities in HMM for news videos

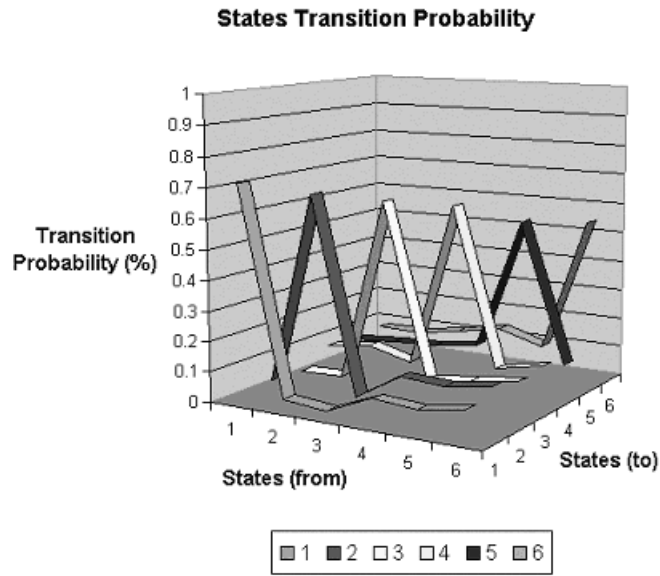


Figure 7: State transition probabilities of HMM for basketball game videos

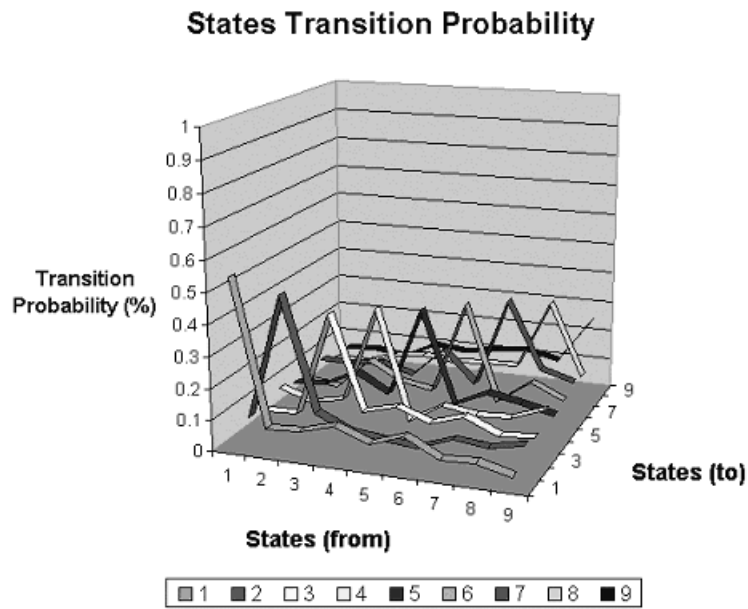


Figure 8: State transition probabilities in HMM for commercials videos

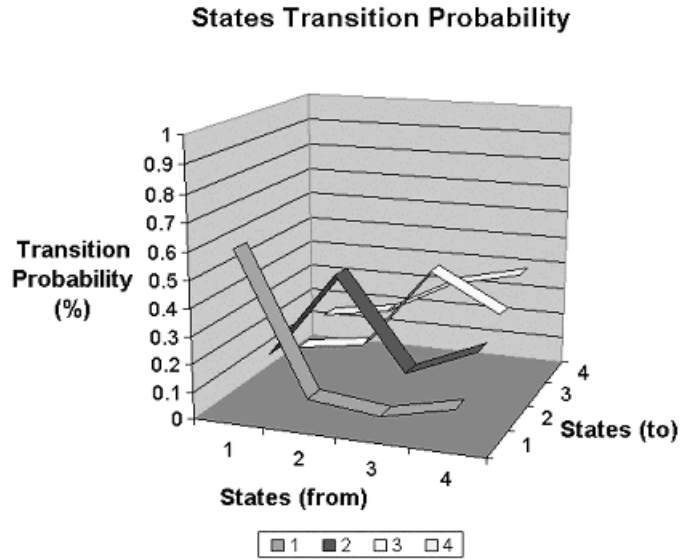


Figure 9: State transition probabilities in HMM for football game videos

Result		Output Class			
		News	Commercial	Basketball	Football
Relevant Class	News	82	6	12	0
	Commercial	14	60	18	8
	Basketball	10	6	84	0
	Football	4	6	0	90

Table 1: Overall classification results (unit: 100%)

News	Commercial	Basketball	Football
82	60	88	90

Table 2: Recall performance (unit: 100%)

News	Commercial	Basketball	Football
74	76	73	91

Table 3: Precision performance (unit: 100%)