Audiovisual Pattern Recognition Using HMM for Content-based Multimedia Indexing

F. Oppini, R. Leonardi
Dept. of Electronics for Automation, Univ. of Brescia, 25123 Brescia (BS)
Email: {oppini@video., leon@}ing.unibs.it

Abstract

The aim of this work consists in the development of automatic techniques for the extraction of content-based information from audiovisual data. The focus has been placed on providing tools for analyzing both audio and visual streams, for translating the signal samples into sequences of indices. The signal classification are performed by means of Hidden Markov Models (HMM), used in an innovative approach: the input signal is considered as a non-stationary stochastic process, modeled by a HMM in which each state stands for a different class of the signal. This defines an adaptive classification scheme for which a set of new training algorithms has been developed. Several samples from the MPEG-7 content set have been analyzed using the proposed classification scheme, demonstrating the performance of the overall approach to provide insights of the content of the audio-visual material.

1. Introduction

Recent developments in computer technology have allowed for large collections of archived digital material and the demand for new solutions based on automatic techniques for the analysis of audiovisual data is increasing. In this work the analysis of multimedia data is based on the extraction of audiovisual indices that might effectively synthesize the semantic features of the input signals. Such analysis concerns different levels of semantic abstraction, starting from audio samples and video frames, up to complex combinations of shots forming scenes carrying semantic significance.

In the next sextion, the principle of HMM-based classification is described, outlining differences in comparison with the usage of HMMs in speech recognition. In Section 3, the basic problems concerning the utilization of HMMs in classification tasks are described, while Section 4 deals with HMM-based classifiers for the characterization of the content of audiovisual material.

2. HMM-based Classification

2.1. Hidden Markov Models

Let us formally define the elements of a HMM. It is characterized by the followings:

1) N, the number of states in the model. Although the states are hidden, for our practical applications there is a physical significance attached to the states of the model.

2) The state transition probability distribution $\boldsymbol{A}=\{a_{ij}\}$, where

$$a_{ij}=\mathrm{P}(q_{t+1}=S_j/q_t=S_i) \qquad (1)$$

For the cases of interest the states are interconnected in such a way that any state can be reached from any other state (i.e., an ergodic model).

3) The observation symbol probability distribution in state j, $\boldsymbol{B}=\{b_j(\boldsymbol{o}_t)\}$, where

$$b_j(\boldsymbol{o}_t)=\mathrm{P}(\boldsymbol{o}_t \text{ at } t/q_t=S_j), \qquad 1\le j\le N,\ 1\le t\le T \tag{2}$$

4) The initial state distribution $\boldsymbol{\pi}=\{\pi_i\}$, where

$$\pi_i=\mathrm{P}(q_1=S_i)\ , \qquad 1\le i\le N \tag{3}$$

It can be seen that a complete definition of a HMM requires the specification of the model parameter, N, and the specification of the three probability measures $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\pi}$. For convenience, the compact notation

$$\lambda=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi}) \tag{4}$$

is used to indicate the complete parameter set of the model.

2.2. The Usage of HMMs in Speech Recognition [1]

The development of Hidden Markov Models has found applications mainly in the field of speech recognition. In this approach, we assume to have a vocabulary of V words (or sequences of sounds) to be recognized, each of these being modeled by a distinct HMM; we also assume that for each word in the vocabulary there are K occurrences of the spoken word, that form the training set. From these spoken words the respective observation sequences are extracted, i.e. some appropriate representation of the spectral and temporal characteristics of the word. The basic strategy in order to implement the speech recognition using HMMs are the following:

1) For each word v in the vocabulary, an HMM λ^v is constructed so as to estimate the model parameters $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$ that optimize the likelihood of the training set observation vectors $\boldsymbol{O}$ for the vth word:

$$\max_{\lambda^v} P\left(\boldsymbol{O}/\lambda^v\right) \qquad 1\le v\le V \tag{5}$$

2) From each unknown word which is to be recognized, the respective observation sequence $\boldsymbol{O}$ is extracted via a feature analysis of the speech associated with the word; the V model likelihoods are then computed for all possible models, $P(\boldsymbol{O}/\lambda^v)$, $1\le v\le V$; finally the word corresponding to the highest model likelihood is selected, i.e.

$$v^* = \underset{1\le v\le V}{\operatorname{argmax}} P\left(\boldsymbol{O}/\lambda^v\right) \tag{6}$$

The training procedure can be achieved by means of the Baum–Welch algorithm, which identifies a local maximum of the likelihood function, while the probability computation step is generally performed using the Viterbi algorithm.

2.3. HMM based classification

Now let us examine a possible utilization of Hidden Markov Models for the separation of audiovisual signals into separate classes. Given an input signal, the objective is to partition it in segments belonging to V distinct classes which have been a priori defined. A possible way to proceed consists in modeling each class with a distinct HMM. It would be meaningful to be able to attach a physical significance to the model states (as it is the case speech recognition where a one-to-one correspondence can be established between the model states and the sounds composing each phoneme).

An effective alternative possibility consists in the usage of a classifier formed by a single HMM, in which each state stands for a different class of the signal (with $V=N$). The probability distributions of the observation $\boldsymbol{B}$ model the statistical properties of the observation vectors that belong to the respective classes, while the transition matrix $\boldsymbol{A}$ shows the probability that a given class at time t is followed by another one at time $t+1$. The system passing through the states of the HMM follows the evolution of the input signal among the classes; based on the correspondence between states of the model and classes of the signal, the sequence of states recognized by the system represents the sequence of classes identified from the input signal. Figure 1 shows the structure of such a HMM, designed to distinguish among three classes.

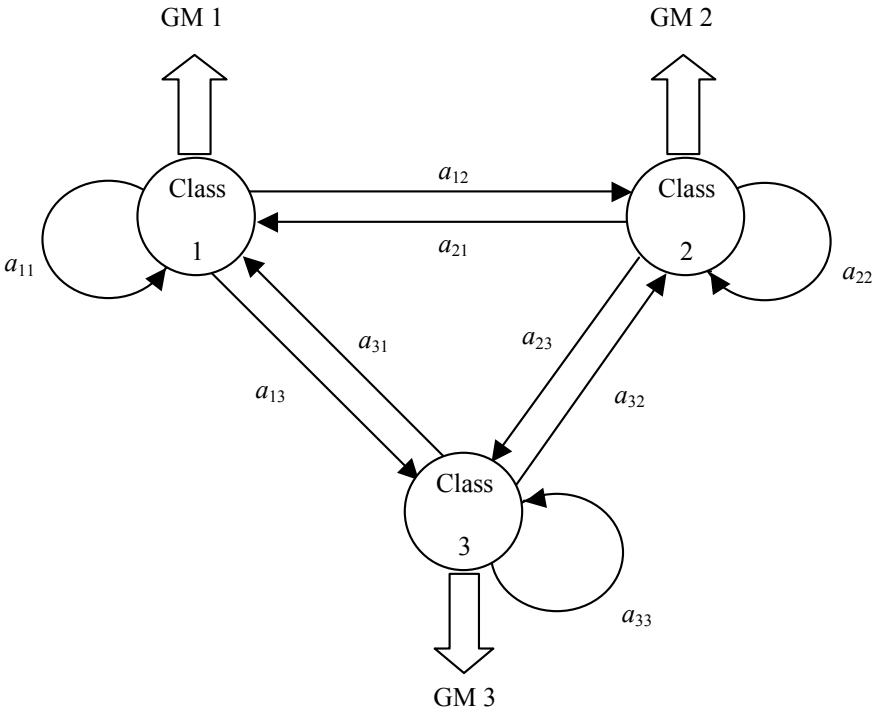


Figure 1. Structure of a classifier constituted by a single HMM, in which each state models a distinct class. Thin arrows connecting model states represent possible transitions, each of them being assigned a transition probability a_{ij}. Thick arrows stand for Gaussian Mixture probability density functions that model the probability distribution of observations for each class.

3. The Two Basic Problems for HMM-based classifiers

3.1. Definition of the Two Problems

Because of the new structure of the classification system, two basic problems need to be solved in order to train the model and to classify an unsupervised input signal. In particular, the procedure of training of the model consists in finding those model parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ that maximize the likelihood $P(\boldsymbol{Q}/\boldsymbol{O},\lambda)$, where $\boldsymbol{O}$ stands for the sequence of observations constituting the training set, and $\boldsymbol{Q}$ the correct sequence of states (i. e. classes) of the model.

The initial state probability vector is set so that $\pi_i=1/N$, $\forall i$ between 1 and N, resulting in the fact that

$$P(q_1=S_i)=b_i(\boldsymbol{o}_1)/N \tag{7}$$

On the other hand, the problem of classifying a sequence of observations $\boldsymbol{O}$ requires to find the sequence of states $\boldsymbol{Q}=q_1q_2\ldots q_T$ that maximizes the likelihood $P(\boldsymbol{Q}/\boldsymbol{O},\lambda)$, where λ is the model obtained by applying the training procedure. An effective way to identify the sequence $\boldsymbol{Q}$ is obtained thanks to the Viterbi algorithm described in [1].

The following training algorithm is proposed. It estimates the model parameters in two steps: first of all the procedure estimates the probability density function of the observation vectors in the V states which are part of the HMM model, and then it modifies its transition matrix $\boldsymbol{A}$.

3.2. The Training of the Model: Estimating the Statistics of the Observations

The first step of the training algorithm, as stated, corresponds to a generic clustering problem, with the aim of approximating the statistical distribution of a multivariate random variable, represented by observation vectors $\boldsymbol{O}$, with a Gaussian Mixture (GM) pdf. The solution can be analyzed for observation vectors belonging to a single class.It estimate the Gaussian mixture parameters relative to the corresponding state (i.e., each class).

The expression for a GM pdf is of the form

$$b(\boldsymbol{x})=\sum_{m=1}^{M} c_m \Gamma_m(\boldsymbol{x}) \tag{8}$$

where $\boldsymbol{x}$ is a vector in a D-dimensional space, $\Gamma_m(\boldsymbol{x})$, $1\le m\le M$, are Gaussian pdf and c_m, $1\le m\le M$, are the mixture weights. Each Gaussian pdf is of the form

$$\Gamma_m(\boldsymbol{x})=\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_m|^{1/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right\} \tag{9}$$

where $\boldsymbol{\mu}_m$ are the mean vectors and $\boldsymbol{\Sigma}_m$ the covariance matrixes.

In order to approximate an arbitrary probability density by means of a GM pdf, the following two problems are addressed

1) Given the number M of mixtures , determine the parameter set ($\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$, c_m), $1\le m\le M$, that better approximates the training data.

2) establish the optimal number M of mixtures that constitute the GM pdf; this second problem is usually said “validity problem”, and is described in [2].

In order to analyze a solution to problem 1, suppose to have a set $\boldsymbol{H}$ of n vectors $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$, which must be approximated by means of a GM pdf in the space $\mathfrak{R}^D$. A way to do this consists in clustering the n vectors in M disjoint subset, in such a way that vectors belonging to the same cluster result somehow more similar than those in different clusters. In particular the sum-of-squared-error criterion is selected as an objective function and then the partition that maximizes the objective function. A typical clustering procedure is obtained thanks to the ISODATA algorithm [2], which comprises the following steps:

1. Choose some initial values for the means $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_M$.

2. Classify the n vectors by assigning them to a cell (identified by its mean) having the closest mean.

3. Recompute the means $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_M$ as the average of the vectors which were attached to any one cell.

4. If any mean changed value, go to step 2; otherwise, stop.

At the end of this iterative procedure, the expression of the GM pdf can be obtained by evaluating the mean vectors and the covariance matrixes of the M disjoint subset, and setting the weights of the mixtures proportionally to the number of vectors in the corresponding subset.

3.3. The Training of the Model: Setting the Transition Matrix

Consider now to have a sequence of observation vectors $\boldsymbol{O}=\boldsymbol{o}_1\boldsymbol{o}_2\ldots\boldsymbol{o}_T$ and the corresponding sequence of states resulting from a supervised classification of those vectors $\boldsymbol{Q_0}=q_1q_2\ldots q_T$. Let us also assume that the probability density functions of the N states of the model are already assigned by means of training algorithms such as ISODATA (see section 3.2).

This following procedure implements an iterative gradient-based maximization technique, which at each iteration improves the similarity between the model and the sequence of observations. Denoting with $\boldsymbol{Q}_R$ the sequence of states in which the model passes – i.e. $P(\boldsymbol{Q}_R/\boldsymbol{O})\geq P(\boldsymbol{Q_0}/\boldsymbol{O})$, if $\boldsymbol{Q}_R\neq\boldsymbol{Q_0}$. The objective is on one hand to increase the probability of obtaining the sequence of states given by $\boldsymbol{Q_0}$ (the manual classification) and on the other hand to reduce the probability of having $\boldsymbol{Q}_R\neq\boldsymbol{Q_0}$ as a result of the classification. From this important property, the training procedure is said to be mutually discriminant, because it tries to discriminate the correct situations from the mistaken ones.

The procedure works as follows:

1. Initialization: Set the N elements π_i of the initial state probability vector to $1/N$; execute a training procedure (such as ISODATA) in order to determine the GM pdf of each state of the model; set initially the N^2 transition probability a_{ij} to $1/N$, $1\leq i,j\leq N$.

2. Denoting by λ the current model, calculate the sequence $\boldsymbol{Q}_R=r_1r_2\ldots r_T$ that maximizes the probability $P(\boldsymbol{Q}_R/\boldsymbol{O},\lambda)$; this operation can be executed by means of Viterbi algorithm.

3. Evaluate the variable

$$\begin{aligned}\alpha &= \frac{P(\mathbf{Q}_0/\mathbf{O},\lambda)}{P(\mathbf{Q}_R/\mathbf{O},\lambda)} = \frac{P(\mathbf{Q}_0,\mathbf{O}/\lambda)}{P(\mathbf{Q}_R,\mathbf{O}/\lambda)} \\ &= \frac{\pi_{q_1} b_{q_1}(\mathbf{o}_1)\cdot a_{q_1q_2} b_{q_2}(\mathbf{o}_2)\cdot\ldots\cdot a_{q_{T-1}q_T} b_{q_T}(\mathbf{o}_T)}{\pi_{r_1} b_{r_1}(\mathbf{o}_1)\cdot a_{r_1r_2} b_{r_2}(\mathbf{o}_2)\cdot\ldots\cdot a_{r_{T-1}r_T} b_{r_T}(\mathbf{o}_T)}\end{aligned} \tag{10}$$

4. Modify the values of coefficients a_{ij}, $1\leq i,j\leq N$, using a gradient-based technique, following the direction of maximum increasing of variable α. This step will be discussed in greater detail later, and it leads to a modified model $\lambda'=(\boldsymbol{\pi},\boldsymbol{A'},\boldsymbol{B})$.

5. Calculate the new sequence of reached states, called $\boldsymbol{Q_R'}$, that maximizes the probability $P(\boldsymbol{Q_R'}/\boldsymbol{O},\lambda')$.

6. Evaluate the value of the variable

$$\alpha' = \frac{P(\mathbf{Q}_0/\mathbf{O},\lambda)}{P(\mathbf{Q}_R'/\mathbf{O},\lambda')} \tag{11}$$

7. If α' results grater than α, assign $\lambda=\lambda'$, $\alpha=\alpha'$ and $\boldsymbol{Q_R}=\boldsymbol{Q_R'}$; then return to step 4. Otherwise, if $\alpha'<\alpha$, halves the value of the gradient step ratio η (see hereafter): if η results less than a threshold value η_{min}, the procedure stops, otherwise go back to step 4.

Now getting back to step 4, in which the transition probabilities a_{ij} are updated in order to maximize the value of α expressed by (10). First of all rewrite (10) as

$$\alpha = \frac{\pi_{q_1}}{\pi_{r_1}} \cdot \frac{a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}}{a_{r_1 r_2} a_{r_2 r_3} \ldots a_{r_{T-1} r_T}} \cdot \frac{b_{q_1}(\boldsymbol{o}_1) b_{q_2}(\boldsymbol{o}_2) \ldots b_{q_T}(\boldsymbol{o}_t)}{b_{r_1}(\boldsymbol{o}_1) b_{r_2}(\boldsymbol{o}_2) \ldots b_{r_T}(\boldsymbol{o}_t)} \tag{12}$$

Modify the N^2 transition probabilities a_{ij} in order to maximize α; thus the objective function can be reduced to the second factor of the previous expression, which will be denoted by β:

$$\beta = \frac{a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}}{a_{r_1 r_2} a_{r_2 r_3} \ldots a_{r_{T-1} r_T}} \tag{13}$$

Now rewrite the equation (13) as

$$\begin{aligned} \beta = & \left[(a_{11})^{s_{11}} (a_{12})^{s_{12}} \ldots (a_{1N})^{s_{1N}} \right] \cdot \left[(a_{21})^{s_{21}} (a_{22})^{s_{22}} \ldots (a_{2N})^{s_{2N}} \right] \\ & \cdot \ldots \cdot \left[(a_{N1})^{s_{N1}} (a_{N2})^{s_{N2}} \ldots (a_{NN})^{s_{NN}} \right] \end{aligned} \tag{14}$$

where s_{ij} are integer values. It is useful to isolate all a_{ij} coefficients for each fixed i, defining the N variables

$$\beta_i = \left[(a_{i1})^{s_{i1}} (a_{i2})^{s_{i2}} \ldots (a_{iN})^{s_{iN}} \right] \qquad 1 \le i \le N \tag{15}$$

obtaining from (14) and (15)

$$\beta = \prod_{i=1}^{N} \beta_i \tag{16}$$

Now it is possible to separate the maximization into N independent maximization of the variables β_i, for each i from 1 to N. Therefore the gradient of β_i with respect to the N variables a_{ij} is computed, obtaining the vector

$$\boldsymbol{g}_i = \nabla \beta_i = \left[g_{i1} \quad g_{i2} \quad \ldots \quad g_{iN} \right] \tag{17}$$

where each g_{ij} is of the form

$$g_{ij} = s_{ij} \frac{\beta_i}{a_{ij}} \qquad 1 \le i, j \le N \tag{18}$$

The respect of the stochastic properties of the transition matrix $\boldsymbol{A}$ requires to correct the values of gradients g_{ij}, using in the re-estimation formulas corrected gradients h_{ij} given by the relation

$$h_{ij} = g_{ij} - \frac{1}{N} \sum_{l=1}^{N} g_{il} \qquad 1 \le i, j \le N \tag{19}$$

so as to reach the final expression for the modified variables a_{ij} :

$$a_{ij}^{new} = a_{ij}^{old} + \eta \cdot h_{ij} \qquad 1 \le i, j \le N \tag{20}$$

The initial value for η can be set according to the (absolute) maximum value of the gradient h_{ij}. In order to respect the non-negativity properties of the transition probabilities a_{ij}, a threshold check is performed, followed by a normalization.

4. HMM-based Audiovisual Classification

4.1. Audio Classification

In order to perform audio classification a partition of the audio stream into four classes of signals (music, silence, speech and background noise) is carried out. First of all the audio signal is split into clips of length T_{clip} seconds, each of those being time-shifted respect the

previous of T_{shift} seconds. Then extract 10 features from each clip, both in time and frequency domains, obtaining a sequence $\boldsymbol{O}$ of observation vectors. The structure of the HMM-based audio classifier is shown in Figure 2, and the procedure of classification consists in finding the maximum-likelihood sequence of states $\boldsymbol{Q}$ that maximizes the probability $P(\boldsymbol{Q}/\boldsymbol{O})$. It can be implemented by using the Viterbi algorithm. In this way the model associates an audio class to each signal clip, reaching a temporal segmentation of the audio signal with a resolution of T_{shift} seconds.

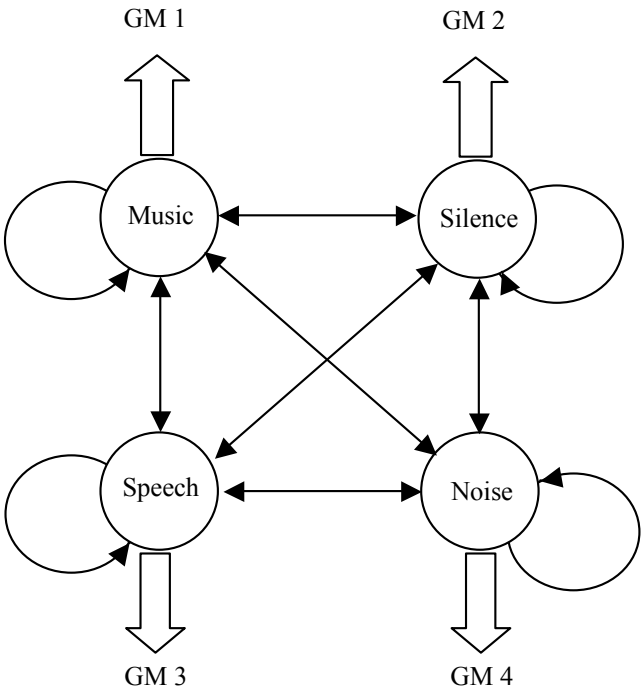


Figure 2. Structure of the audio classifier: the four states are associated to the classes to identify.

4.2. Video classification

The aim of video classification consists in partitioning the input signal in shots, i.e. sequences of adjacent frames that present a similar video content. In other words we want to identify those instants in which the image content changes, abruptly or continuously. The video classifier extracts a five-component feature vector from each couple of adjacent frames, obtaining a sequence $\boldsymbol{O}$ of observations; then it looks for the maximum likelihood sequence $\boldsymbol{Q}$ of states, using the Viterbi algorithm. The video classifier is composed of two states: state 1 stands for "no change in video content", while state 2 stands for "change": This way we recognize the shot changing in those instants in which the system passes through state 2. Figure 3 shows the structure of the video classifier.

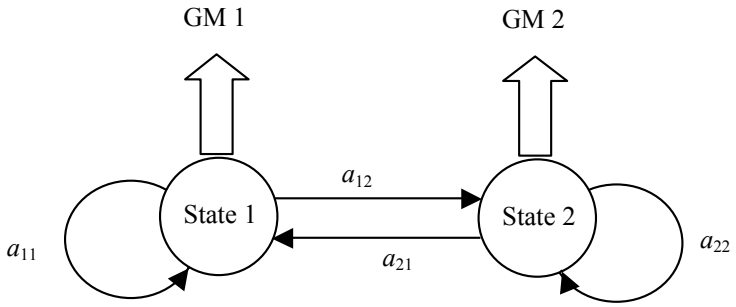


Figure 3. Structure of the video classifier.

It is alternatively possible to implement a between-shot correlation analysis, extracting feature vectors from couples of representative frames of non-adjacent shots, and then looking for the maximum likelihood sequence of states of a HMM whose transition matrix has any elements equal to $1/N$.

4.3. Performance

We trained both classifiers using a training set composed of 30 minutes of audiovisual signal taken from MPEG–7 Video Content Set, and then we tested the performances over 60 minutes of signal; results are shown in Table 1 and Table 2.

	Rec. music	Rec. Silence	Rec. Speech	Rec. Noise
Music	**80.5%**	3.6%	11.4%	4.5%
Silence	2.4%	**95.8%**	1.8%	0%
Speech	11.3%	2.3%	**85.4%**	1%
Noise	3.1%	2.9%	2.5%	**91.5%**

Table 1. Performance of the HMM-based audio classifier.

	Rec. state 1	Rec. state 2
State 1	95%	5%
State 2	1.1%	98.9%

Table 2. Performance of the HMM-based video classifier.

Figure 4 shows the results of the classifications of three TV programmes representing respectively 3 minutes of a talkshow, of a music program, and of a scientific documentary. Different statistical evolution of audio and video indices can be used to infer the semantic content of the analyzed signals, as it can be noticed from the chronograms shown in Figure 4.

5. Conclusion

A HMM based framework for the classification of audio and video information from audio-visual documents has been proposed. The classification results are quite satisfactory, and it appears that a joint visualization of audio and visual properties provide reach insights on the audio-visual programme content (see Figure 4). In terms of complexity the proposed scheme is quite efficient once the training phase has taken place. Further research is needed to assess the robustness of the classification procedure,to determine the adequacy of the proposed HMM model with respect to other classification procedures.

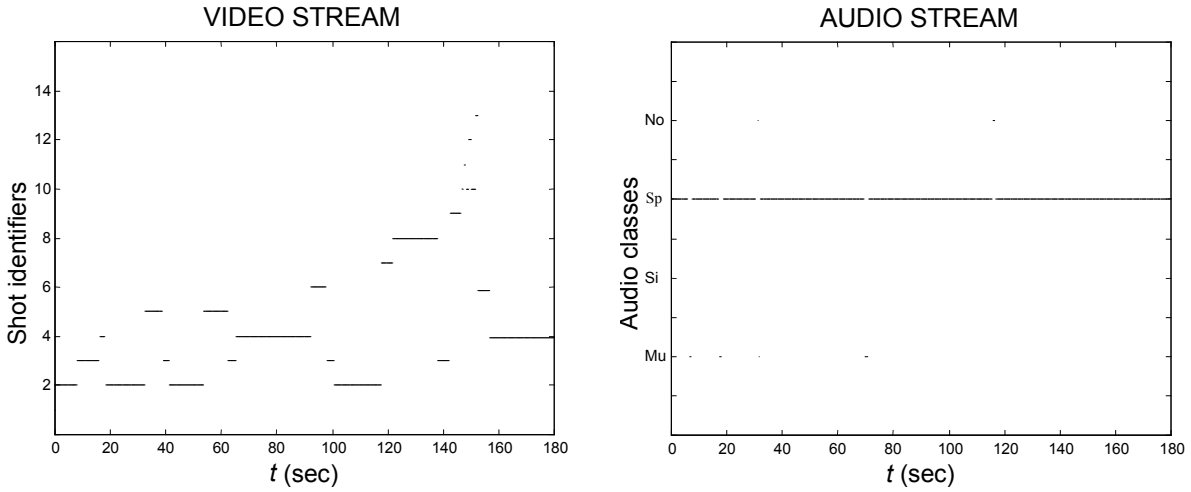
Bibliography

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

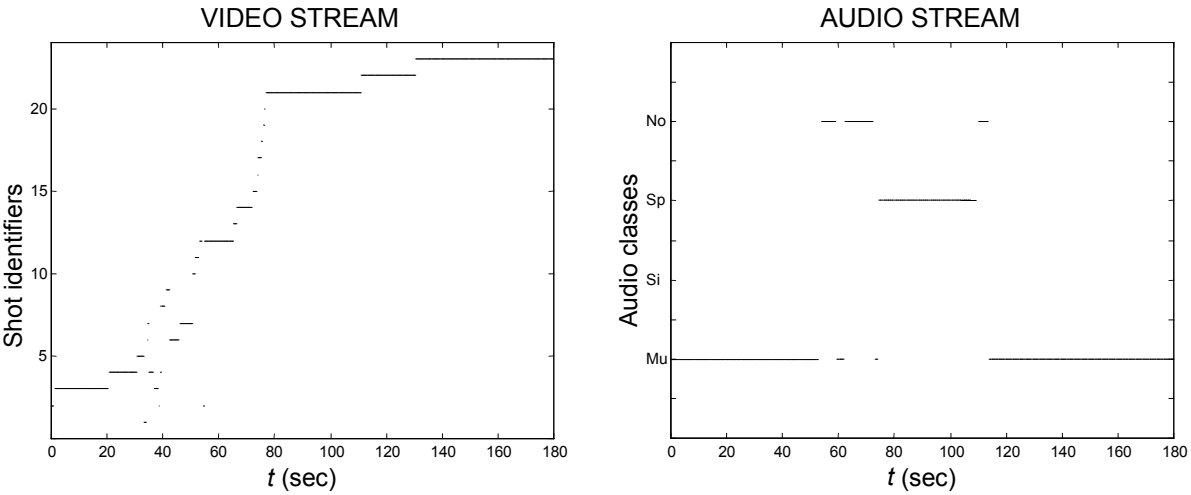
[2] R. O. Duda, P. E. Hart, "Pattern Classification and Scene Analysis", John Wiley and Sons, 1973.

[3] C. Saraceno, R. Leonardi, "Identification of Story Units in Audio-Visual Sequences by Joint Audio and Video Processing", Proceedings of ICIP '98, pp.363-367, 1998.

(a) Talkshow



(b) Music program



(c) Scientific Documentary

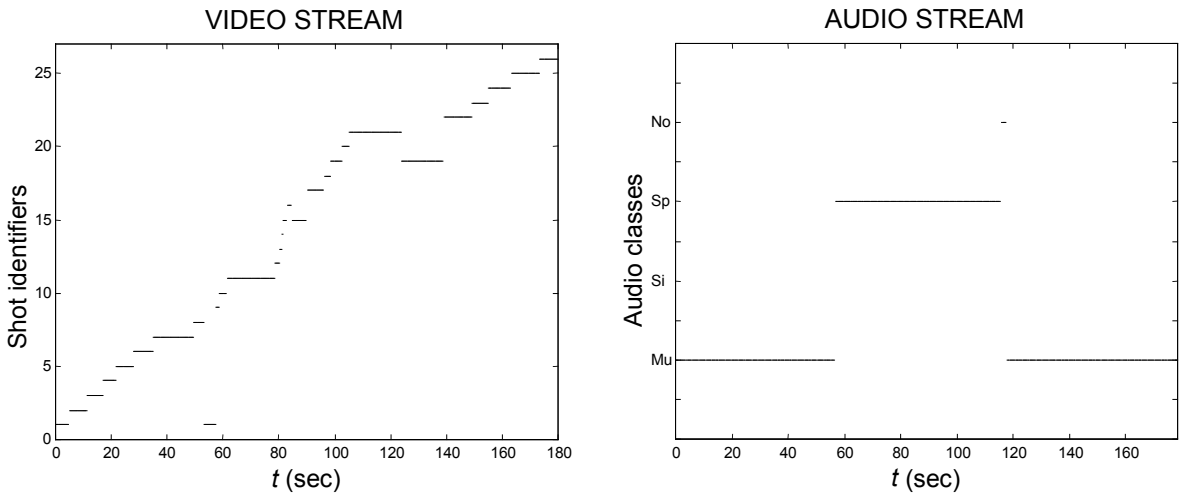


Figure 4. Audio and video classification results of 3 minute of a talkshow, a music program and a scientific documentary. In the video stream diagram, the shot label is associated to the visual content of the relative shot, in such a way that shots with similar visual content are denoted by the same label.