Discovering Video Structure Using The Pseudo-Semantic Trace

Cuneyt M. Taskiran, Charles A. Bouman, and Edward J. Delp

Video and Image Processing Laboratory (*VIPER*)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285

ABSTRACT

In this paper we describe a framework of analyzing programs belonging to different TV program genres using Hidden Markov Models and pseudo-semantic features derived from video shots. Clustering using Gaussian mixture models is used to determine the order of the models. Results for initial genre classification experiments using two simple features derived from video shots are given.

Keywords: video database, hidden markov model, pseudo-semantic feature

1. INTRODUCTION

With the proliferation of video material, it has become important to be able to automatically analyze and index video data and retrieve its relevant parts. Consequently, there has been a great interest in designing and building systems that organize and search video data based on its content.[1] It is evident that, for an information source as rich as video, efficient indexing and retrieval require some form of automatic analysis of semantic content. Since the smallest meaningful unit of video is the shot, this may be achieved by giving semantic labels to the shots. The purpose of semantic labeling of a shot is to classify or label the shot using a high level semantic description. True semantic labels such as "young girl running," "blue dress," and "park scene" characterize a shot based on its content. Ideally, such semantic labels might provide the most useful descriptions for indexing and searching video databases. Currently, however, automatic extraction of such truly semantic features is not possible. One way to circumvent this problem is to define features that correlate well with high level semantic labels and hence are useful in bridging the gap between low-level image features and semantic labels. We call such features *pseudo-semantic features*.

Previously we have described an integrated video database system known as *ViBE* (video indexing and browsing environment) for managing large amounts of video.[2–5] In *ViBE* a variety of algorithms and techniques for processing, representing, and managing video are tightly integrated into a single system which can be scaled to large database sizes and extended to a wide variety of functionalities. Figure 1 illustrates the four major components of *ViBE*: shot boundary detection, hierarchical shot representation, pseudo-semantic shot labeling, and active browsing.

In this paper we will concentrate on the pseudo-semantic shot labeling component of *ViBE* system. We will describe initial experiments in using low level image features derived from the compressed video stream to generate pseudo-semantic labels for shots. We will also investigate the use of a Hidden Markov Model (HMM) based approach to building stochastic models for different types of program genres such as soap operas and talk shows. These models can be used both to classify sequences into different program genres and to analyze their structure.

HMMs are versatile tools to analyze time series whose statistical properties may change with time. Their applications range from spoken word recognition[6] to analysis of DNA sequences.[7] In a HMM an underlying and unobserved sequence of states follows a Markov chain with finite state space, and the probability distribution of the observation at any time is determined only by the current state of that Markov chain.[8]

Recently a number of researchers have applied HMMs to different aspects of the video analysis task. Boreczky and Wilcox[9] used a HMM trained on audio features in addition to the motion features derived from adjacent frames to detect shot boundaries. Eickeler and Muller[10] use image features derived from difference images to train a HMM which in turn is used both to detect shot boundaries and classify types of shots for news sequences. Using shot labels such as *medium shot* or *close-up*, Wolf[11] has trained a HMM to detect dialogs in video sequences. Liu et al[12] have used various features derived from audio to classify TV programs using a discrete HMM.

This work was supported by a grant from the Indiana 21st Century Research and Technology Fund. Address all correspondence to E. J. Delp, `ace@ecn.purdue.edu`, `http://www.ece.purdue.edu/~ace` or telephone: +1 765 494 1740.

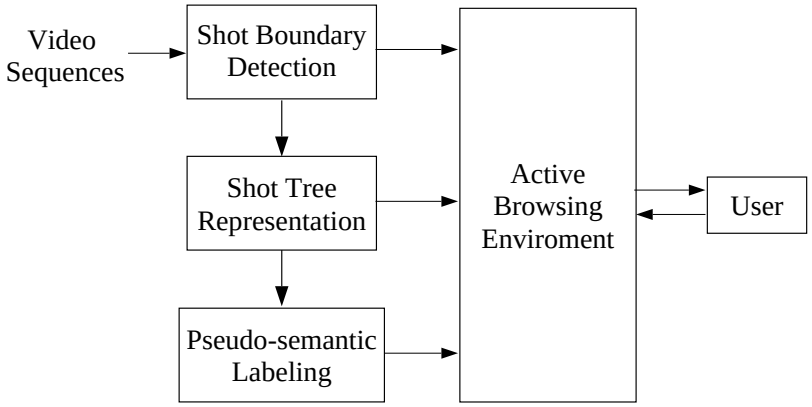


Figure 1. A schematic view of the *ViBE* system

2. EXTRACTION OF PSEUDO-SEMANTIC FEATURES

As pointed above, we want the pseudo-semantic features derived from a shot to correlate well with the semantic labels. On the other hand, we do not want the features to be complex, they should be easily derivable from the compressed video stream with reasonable computational burden. In *ViBE* pseudo-semantic shot labeling is performed as follows: First the DC sequence is extracted from the compressed video sequence. The DC sequence is formed from the DC coefficients of the DCT used in MPEG. While the DC coefficients are directly available for I frames, they must be estimated for P and B frames. We have used the method described in[13] for estimating the DC coefficients. The shot boundaries are determined using a regression tree classifier as described in.[14]

After shot boundaries are determined, a number of pseudo-semantic features are extracted from each shot in a video sequence. We have been experimenting with a number of such features such as "face" and "indoor-outdoor" for a shot,[2.15] However, for this paper, we shall report results for two simpler features: the length of the shot and the average number of macroblocks with motion vectors within the shot. The shot length feature is an indication of the editing pattern used in the video sequence which is different for different video genres. The number of macroblocks with motion vectors is a rough indication of the amount of motion taking place within frames. Note that even if there is little motion, if te image content contains textures which are fairly uniform we can get a lot of macroblocks with motion vectors dur to the block matching process in MPEG coding. Shot length and some indication of average shot activity have been shown to be useful in discriminating between different types of movie trailers.[16]

Let b_n and e_n denote the start and end frame numbers of the n^{th} shot in a video sequence. The components of the feature vector, Y_n, for this shot are then defined as

$$Y_{n1} = \frac{1}{e_n - b_n + 1} \sum_{k=e_n}^{b_n} m_k \quad (1)$$

$$Y_{n2} = e_n - b_n + 1 \quad (2)$$

where $m_k = (\#\,\text{forward MB}) + (\#\,\text{backward MB}) + 2(\#\,\text{forward-backward MB})$ for the k^{th} frame for P and B frames and is zero for I frames.

We call the sequence of vetors Y_n the *pseudo-semantic trace* for the video sequence. The components of the pseudo-semantic trace for a sequences from three different program classes are plotted in Figure 2.

3. FEATURE CLUSTERING AND DETERMINATION OF MODEL ORDER

After the feature vectors are extracted from each shot, they are modeled using a Gaussian mixture model. We use an agglomerative clustering algorithm to estimate the parameters of the Gaussian mixture model and the number of clusters from training data. The estimated number of clusters is then taken to be the order of the HMM to be built, as described in Section 4.

Let $Y_1, Y_2, \ldots, Y_N$ be N feature vectors derived from the shots in the training set which are assumed to belong to K clusters. Furthermore, assume that the the random variables $X_1, X_2, \ldots, X_N$, denote the clusters that the feature

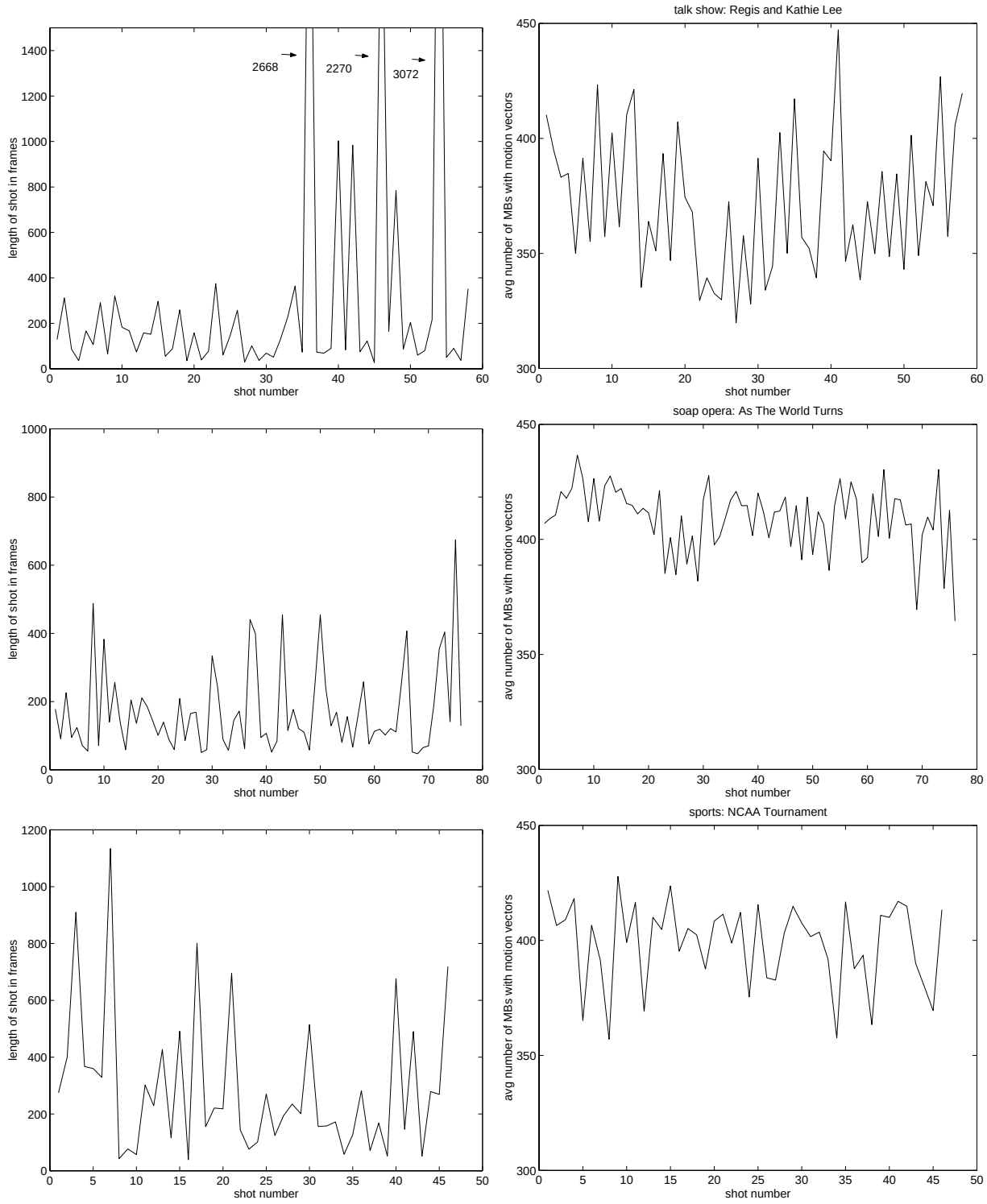


Figure 2. Components of the pseudo-semantic traces for sequences from three different program classes. (a) Lengths of shots in frames. (b) The average number of macroblocks with motion vectors in a shot.

vectors belong to. Then, assuming that each cluster has a multivariate Gaussian distribution, the probability density function for the feature vector Y_n given that it belongs to the k^{th} cluster is given by

$$p_{y_n|x_n}(y_n|k,\theta) = \frac{1}{(2\pi)^{M/2}}|R_k|^{-1/2}\exp\left\{-\frac{1}{2}(y_n-\mu_k)^T R_k^{-1}(y_n-\mu_k)\right\} \tag{3}$$

where $\theta = (\pi, \mu, R)$ is the complete set of parameters, M is the dimensionality of the feature vectors, and μ_k and R_k are the mean vector and covariance matrix for cluster k, respectively. The log-likelihood for the entire sequence $Y = \{Y_n\}_{n=1}^N$ may then be written as

$$\log p_y(y|K,\theta) = \sum_{n=1}^{N}\log\left(\sum_{k=1}^{K} p_{y_n|x_n}(y_n|k,\theta)\pi_k\right) \tag{4}$$

where π_k is the probability that Y_n belongs to cluster k. The parameters θ can then be estimated using the maximum likelihood (ML) estimate given by

$$\hat{\theta}_{ML} = \arg\max_{\theta} \log p_y(y|K,\theta) \tag{5}$$

It is not possible to estimate the number of clusters, K, using a similar approach. The ML estimate for K is not well-defined because the likelihood may always be made better by increasing the number of clusters. Methods for estimating model order generally generally require the addition of an extra term which penalizes over-fitting of higher order models. One such technique is to minimize the Akaike Information Criterion[17]

$$AIC(K,\theta) = -2\log p_y(y|K,\theta) + 2L \tag{6}$$

where L is the number of real valued numbers required to specify the parameters of the model, θ. In our case, we have

$$L = K(1 + M + \frac{M(M+1)}{2}) - 1 \tag{7}$$

Unfortunately the above criterion does not lead to a consistent estimator. We have used another criterion, called the minimum description length (MDL) criterion,[18] which has better asymptotic convergence properties and is defined as

$$MDL(K,\theta) = -\log p_y(y|K,\theta) + \frac{1}{2}L\log(NM) \tag{8}$$

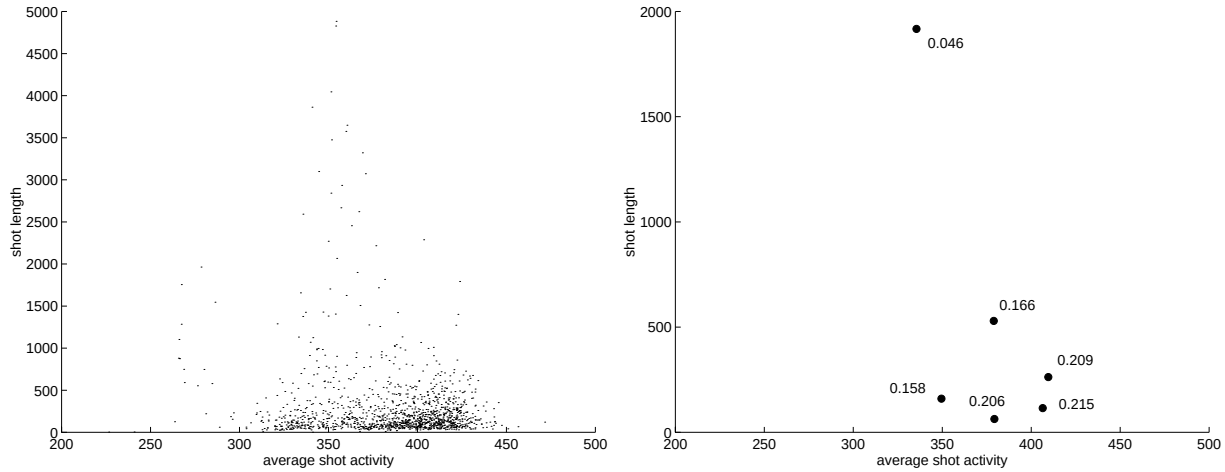
The minimization of the above criterion is performed iteratively using the expectation-maximization (EM) algorithm. We start with a high number of initial clusters, usually 2-3 times the expected number of clusters, and at each step merge those clusters which cause the maximum decrease in the MDL criterion. This process is continued until only one cluster is left. Then, the number of clusters for which the minimum value of MDL was achieved is chosen as the estimate of the number of clusters.

The feature vectors for all shots in the database are plotted in Figure 3 together with the mean vectors of the Gaussian mixtures estimated by the clustering procedure described above. The estimated probabilities for the clusters are shown next to the means.

4. BUILDING HIDDEN MARKOV VIDEO GENRE MODELS

The Gaussian mixture densities found above may directly be used to build HMMs with continuous observation densities.[6] However, we have chosen to use HMMs characterized by discrete observation symbols chosen from a discrete alphabet. In this case, the symbol densities for each state will be discrete. A discrete-valued HMM is characterized by the following parameters

1. N, the number of states in the model. The state at time t is shown by $q_t \in \{S_1, \ldots, S_N\}$.
2. M, the number of distinct observation symbols per state. Individual symbols are denoted as $V = \{v_1, \ldots, v_M\}$.
3. $A = \{a_{ij}\}$, the state transition matrix, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$.
4. $B = \{b_j(k)\}$, the observation symbol probability distribution in state j, where $b_j(k) = P(v_k \text{ at } t | q_t = S_j)$.



(a) (b)

Figure 3. (a) Two dimensional feature vectors for all the shots in the database. (b) The mean vectors for the Gaussian mixtures estimated using the clustering procedure. The estimated cluster occurrence probabilities are shown next to the means.

5. $\pi = \{\pi_i\}$, the initial state distribution, where $\pi_i = P(q_1 = S_i)$.

The notation $\lambda = (A, B, \pi)$ is used to refer to the model.

The type of the HMM also needs to be chosen. One possibility is to use an *ergodic* or fully connected HMM, for which $a_{ij} > 0$ for all values of i and j and hence every state can be reached from any other. Another model, which is widely used in speech recognition, is the *left-right* model which has the properties that $a_{ij} = 0$ for $j < i$, i.e., no transitions are allowed to states whose indices are lower than the current state, and that the the sequence always starts in state 1 and ends in state N. In[10] a left-right HMM was used to segment and classify semantic components of news programs. However we agree with[12] in that an ergodic model may be more suitable for modeling video programs because they generally exhibit recurring characteristics. Hence, we have used an ergodic HMM for genre modeling.

We use the clustering scheme of Section 3 to estimate both the number of symbols and the order of the HMM, i.e., we take $N = M = \hat{K}$ where $\hat{K}$ is the estimated number of clusters. From Figure 3 we see that $\hat{K} = 6$ with one of the clusters being very unlikely to occur. Note this value of the order of the HMMs to describe video sequences from different genres agrees very well with the value found in[12] which was derived using ad hoc considerations of classifier accuracy.

The symbol corresponding to each shot is determined by the cluster the shot feature vector is most likely to belong, that is, given the feature vector, Y_n, for the shot we determine the corresponding symbol, v_n, using

$$v_n = \arg \max_{k \in [1, \hat{K}]} p_{y_n|x_n}(y_n|k, \theta) \tag{9}$$

After the symbol sequences are obtained for all the shots in the training set we train a HMM for each video genre using the shots in the training set belonging to that genre.

The models thus obtained are used to classify a given sequence to one of the genres. Given HMMs for the L genres, $\lambda_1, \ldots, \lambda_L$, a new sequence, with pseudo-semantic trace $Y_1, \ldots, Y_m$, is classified by choosing the model that is most likely to produce the given sequence. This is accomplished by first converting the feature vectors of the sequence to discrete symbols using Equation 9 and then estimating the probabilities $p(v_1, \ldots, v_m|\lambda_i)$ for all genres by using the forward-backward procedure.[6] The decision rule then becomes

$$\text{genre of } S = \arg \max_{i \in [1, L]} p(v_1, \ldots, v_m|\lambda_i)$$

5. RESULTS

5.1. The Experimental Data Set

All sequences in our database have been digitized at a rate of 1.5 Mb/sec in CIF format, i.e., 352×240 using MPEG-1 compression. Commercials in the sequences, if they exist, were edited out. The locations of all the shot transitions in these sequences were recorded by a human operator. These 27 sequences are distributed among six program genres as follows:

- Soap operas (6 sequences, 415 shots). Two episodes from *The Young And The Restless* and *As The World Turns* each, one episode from *The Bold And The Beautiful* and *Guiding Light*, These consist mostly of dialogue with cut transitions between them and have little camera motion.
- Talk shows (7 sequences, 447 shots). One sequence from *The Rosie O'Donnell Show*, two sequences from *Oprah Winfrey Show*, *Late Night With David Letterman*, and *Regis and Kathie Lee* each.
- Sports programs (6 sequences, 335 shots). One college and one NFL football sequence, two sequences from the NCAA Tournament, and two sequences with car races, one from NASCAR and one from the Texas 500 Car Race.
- Sequences from CSPAN (8 sequences, 168 shots). Obtained from CSPAN-I and CSPAN-II. Consists of long shots with very little camera or object motion.

Note that we never use more than one sequence from a given airing of a particular program, in order to achieve maximum content variation.

5.2. Computation of Model Distances Between Genres

It is interesting to compare the HMMs obtained for different video genres. However, this cannot be done by inspection of the model parameters. As pointed out in,[6] the parameters for two HMMs, λ_1 and λ_2 may look very different, yet the HMMs may be equivalent in the sense that the statistical properties for the observation symbols are the same, i.e., $E\{O_t = v_k|\lambda_1\} = E\{O_t = v_k|\lambda_2\}$.

The distance between two HMMs, λ_i and λ_j may be defined as

$$d(\lambda_i, \lambda_j) = \frac{D(\lambda_i, \lambda_j) + D(\lambda_j, \lambda_i)}{2} \tag{10}$$

where

$$D(\lambda_i, \lambda_j) = \frac{1}{T}\left[\log P(O^{(i)}|\lambda_i) - \log P(O^{(i)}|\lambda_j)\right] \tag{11}$$

is the asymmetric measure of the difference between intermodel and intramodel matches for a sequence, and $O^{(i)} = O_1 O_2 \cdots O_T$ is a sequence of observations generated by model λ_i.

For this experiment we trained four HMMs, $\lambda_1, \ldots, \lambda_4$, for the different genres in the database by using all available sequences for each genre. Then we generated sequences of length $T = 5000$ using each model and computed pairwise distances using Equation 10. The resulting distances are given in Table 1. By examining the values for the distances we can see that the *cspan* class is well-separated from the other three classes. Also the models for the two classes *soap* and *sports* are close together, which will hinder the classification accuracy for these classes.

5.3. Genre Classification Experiments

To get an honest estimate of the performance of the HMMs that were built in classifying sequences, we have used the following procedure which is similar to a cross-validation scheme

for $i = 1$ to 6
  for each genre $G \in \{soap, talk, sports, cspan\}$
    randomly choose two sequences from G and place them in the test set
    use remaining sequences in G to train a HMM, λ_G, for G
  classify all sequences in the test set using the HMMs for genres, $\lambda_1, \ldots, \lambda_4$
average the six set of values to obtain the classification performance

	soap	talk	sports	cspan
soap	0	2.0329	1.7878	3.9908
talk		0	2.0938	3.2408
sports			0	3.4236
cspan				0

Table 1. Pairwise distances between HMMs of different genres for sequences of 5000 symbols.

	Classifier Output			
True Label	soap	talk	sports	cspan
soap	0.583	0.25	0.167	0.0
talk	0.0	0.833	0.167	0.0
sports	0.333	0.083	0.583	0.0
cspan	0.0	0.0	0.083	0.917

Table 2. Confusion matrix when classifying different genres using HMMs of order 6.

The results are shown in Table 2. From this table we can see that the results agree with the distance distribution between classes given in Table 1. The similarity between the models λ_{soap} and λ_{sports} cause many misclassifications between these two genres. Notice from Table 1 that the distance between *sports* and *talk*, $d(\lambda_{sports}, \lambda_{talk}) = 2.0938$ is greater than $d(\lambda_{soap}, \lambda_{talk}) = 2.0329$ which accounts for the fact that soap opera sequences are misclassified as talk shows more often than sports sequences.

6. CONCLUSION

In this paper we have outlined a method to analyze video sequences from different types of TV genres using Hidden Markov Models. The HMMs were built using a sequence of feature vectors, which we called the pseudo-semantic trace, derived from the shots in a video sequence. In this paper we have used two simple pseudo-semantic features for a shot, the average number of MBs with motion vectors for the frames in the shot and the length of the shot in frames. We have shown that agglomerative clustering of feature vectors using a Gaussian mixture model is an efficient way to determine the order of HMMs if an a priori model for video structure does not exist. We have also shown that the pairwise distances between HMMs is a valuable tool in predicting the performance of a classifier based on these HMMs.

Our genre classification experiments using the HMMs built using the above two features have shown that these features are able to distinguish between programs from CSPAN and talk shows successfully. However, the classification performance was low in discriminating between soap operas and sports programs. It is evident that more sophisticated features are called for to increase classification accuracy although the fact that the HMMs have the classification performance they have even using such simple features suggest that they are powerful tools in description of video content.

REFERENCES

1. E. J. Delp, "Video and image databases: Who cares?," *Proceedings of the SPIE/IS&T Conference on Storage and Retrieval for Image and Video Databases VII*, January 23-29 1999, San Jose, CA, pp. 274–277.

2. J.-Y. Chen, C. Taskiran, A. Albiol, E. J. Delp, and C. A. Bouman, "Vibe: A compressed video database structured for active browsing and search," *IEEE Transactions on Image Processing*, submitted to.
3. J.-Y. Chen, C. Taskiran, A. Albiol, C. A. Bouman, and E. J. Delp, "Vibe: A video indexing and browsing environment," *Proceedings of the SPIE Conference on Multimedia Storage and Archiving Systems IV*, vol. 3846, September 1999, Boston, MA, pp. 148–164.
4. C. Taskiran, J.-Y. Chen, A. Albiol, C. A. Bouman, and E. J. Delp, "A compressed video database structured for active browsing and search," *Proceedings of the IEEE International Conference on Image Processing*, October 1998, Chicago, IL.
5. J.-Y. Chen, C. Taskiran, C. A. Bouman, and E. J. Delp, "Vibe: A new paradigm for video database browsing and search," *Proceedings of the 1998 IEEE Workshop on Content-Based Access of Image and Video Databases*, June 21 1998, Santa Barbara, CA.
6. L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.
7. J. H. S.Salzberg and K. Fasman, "Finding genes in human dna with a hidden markov model," *Journal of Computational Biology*, vol. 4, no. 2, pp. 127–141, 1997.
8. I. L. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series*. London, UK: Chapman and Hall, 1997.
9. J. S. Boreczky and L. D. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, May 1998, Seattle, WA, pp. 3741–3744.
10. S. Eickeler, A. Kosmala, and G. Rigoll, "A new approach to content-based video indexing using hidden markov models," *Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services*, June 1997, Louvin-la-Neuve, Belgium, pp. 149–154.
11. W. Wolf, "Hidden markov model parsing of video programs," *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, April 1997, Munich, Germany, pp. 2609–2611.
12. Z. Liu, J. Huang, and Y. Wang, "Classification of tv programs based on audio information using hidden markov model," *IEEE Second Workshop on Multimedia Signal Processing*, December 1998, Redondo Beach, CA, pp. 27–32.
13. K. Shen and E. J. Delp, "A fast algorithm for video parsing using MPEG compressed sequences," *Proceedings of the IEEE International Conference on Image Processing*, October 26-29 1995, Washington, D.C., pp. 252–255.
14. C. Taskiran, C. Bouman, and E. J. Delp, "The vibe video database system: An update and further studies," *Proceedings of the SPIE/IS&T Conference on Storage and Retrieval for Media Databases 2000*, January 26-28 2000, San Jose, CA, pp. 199–207.
15. A. Albiol, C. A. Bouman, and E. J. Delp, "Face detection for pseudo-semantic labeling in video databases," *Proceedings of the IEEE International Conference on Image Processing*, October 25-28 1999, Kobe, Japan.
16. N. Vasconcelos and A. Lipman, "Towards semantically meaningful feature spaces for the characterization of video content," *Proceedings of IEEE International Conference on Image Processing*, October 1997, Santa Barbara, CA, pp. 78–89.
17. H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, December 1974.
18. J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals of Statistics*, vol. 11, no. 2, pp. 417–431, 1983.