# A Hidden Markov Model Framework for Video Segmentation Using Audio and Image Features

*John S. Boreczky and Lynn D. Wilcox*
FX Palo Alto Laboratory
Palo Alto, CA 94304 USA

## ABSTRACT

This paper describes a technique for segmenting video using hidden Markov models (HMM). Video is segmented into regions defined by shots, shot boundaries, and camera movement within shots. Features for segmentation include an image-based distance between adjacent video frames, an audio distance based on the acoustic difference in intervals just before and after the frames, and an estimate of motion between the two frames. Typical video segmentation algorithms classify shot boundaries by computing an image-based distance between adjacent frames and comparing this distance to fixed, manually determined thresholds. Motion and audio information is used separately. In contrast, our segmentation technique allows features to be combined within the HMM framework. Further, thresholds are not required since automatically trained HMMs take their place. This algorithm has been tested on a video data base, and has been shown to improve the accuracy of video segmentation over standard threshold-based systems.

## 1. INTRODUCTION

An important aspect of video indexing is the ability to segment video into meaningful segments. One type of segment is a shot, or a sequence of video frames from a single camera. In produced video such as television or movies, shots are separated by different types of transitions, or boundaries. The simplest transition is a cut, an abrupt shot change that occurs in a single frame. Gradual transitions between shots are more complex. Two common types that we consider in this work are fades and dissolves. Camera movement within a shot can sometimes be mistaken for gradual transitions. We model two types of camera movement, the pan and the zoom.

Shot boundaries are typically found by computing an image-based distance between adjacent frames of the video, and noting when this distance exceeds a certain threshold. The distance between adjacent frames can be based on statistical properties of pixels [4], compression algorithms [1], or edge differences [13]. The most widely used method is based on histogram differences. If the bin-wise difference between histograms for adjacent frames exceeds a threshold, a shot boundary is assumed. Zhang *et al.* [14] used this method with two thresholds in order to detect gradual transitions.

Audio and motion features have been used to improve shot boundary detection. Saraceno *et al.*[8] classify audio according to silence, speech, music, or noise and use this information to verify shot boundaries hypothesized by image-based features. In [5], speaker identification is used to cluster shots. Phillips and Wolf [6] use motion features alone or with histogram differences to improve boundary detection. Shahraray [9] combines motion features with pixel differences.

In this work, we combine information from features that are based on image differences, audio differences, and motion for segmenting video. Hidden Markov models provide a unifying framework for jointly modeling these features. In Wolf [12], HMMs are used to build scenes from video which has already been segmented into shots and transitions. Here, HMMs are used to perform segmentation directly based on multiple features.

States of the HMM consist of the various segments of a video, namely the shots themselves, the transitions between them: cuts, fades, and dissolves, and camera motion: pans and zooms. The HMM contains arcs between states showing the allowable progressions of states. The parameters of the HMM are learned using training data in the form of the frame-to-frame distances for a video labeled with shots, transition types, and motion. Once the HMM is trained, it can be used for segmenting video into its component shots and transitions by applying the Viterbi algorithm to determine the most likely sequence of states through the HMM.

## 2. Features

We consider three types of features for use in video segmentation. The first is a standard histogram distance, which measures the difference between adjacent frames of video based on a gray-scale histogram. The second is an audio distance measure, which computes the distance between the audio in intervals just before and just after the frames. The third feature is based on an estimate of object motion between two adjacent video frames.

### 2.1 Image Features

The histogram feature measures the distance between adjacent video frames based on the distribution of luminance levels. It is simple, easy to compute, and works well for most types of video [2]. The luminance of a pixel $L_{pixel}$ is computed from the 8-bit red (R), green (G), and blue (B) components as

$$L_{pixel} = .3008(R)+.5859(G)+.1133(B).$$

$H$ is a 64 bin histogram computed by counting the number of pixels in each bin of 4 gray levels, thus

$$H[k]=\#\ of\ pixels\ where\ k=L_{pixel}/4,\ 0<=k<=63.$$

The histogram feature $D_H$ is the absolute bin-wise difference of the histograms of adjacent frames. $D_H$ is computed as

$$D_H = \Sigma|H[k] - H_{prev}[k]|,\ 0<=k<=63.$$

The image distance is computed between each pair of adjacent frames at the frame rate of 30 times per second.

## 2.2 Audio Features

In contrast to other approaches to using audio features to aid in video segmentation, we do not attempt to categorize audio into classes such as speech, silence, music, and noise [8], or to identify speakers [5]. Rather, we take the approach used with the video feature and compute an audio distance measure. This distance is computed between two adjacent intervals of audio ***X*** and ***Y*** (see Figure 1). In order for this distance to accurately reflect differences in the type of audio (speech, silence, etc.) it is necessary to use a relatively long interval. This is because speech is composed of short (approximately 30 ms) intervals containing either silence, periodic voiced signal (typically vowels), or noise. Silence can also appear as pauses in speech. Thus using short analysis windows for audio could show large differences in purely speech audio. In our work, we compute audio distances based on sliding two-second intervals.

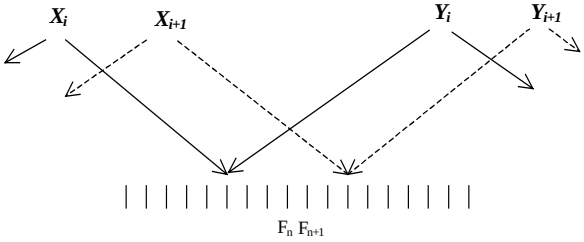


Figure 1. $F_n$ and $F_{n+1}$ are adjacent frames of video sampled at 30 frames per second. $\boldsymbol{X_i}$ and $\boldsymbol{Y_i}$ are 2 second intervals used for audio analysis.

Audio is first converted to a sequence of 12-dimensional cepstal vectors $v_i$, computed every 20 ms. [7]. Thus the two-second interval ***X*** consists of a sequence of 100 cepstral vectors $\boldsymbol{X}=(v_1,\ldots,v_{100})$. Let ***X*** and ***Y*** be two such intervals, and let ***Z*** be the four-second interval obtained by concatenating ***X*** and ***Y***, ***Z*** = ***(X,Y)***.

The audio distance measure is similar to the likelihood ratio measure first suggested by Gish [3] and used in [10] for audio segmentation. Let $H_0$ denote the hypothesis that ***X*** and ***Y*** are the same audio type, and let $H_1$ denote the hypothesis that they are different types of audio. Let $L(\boldsymbol{X}:\theta_X)$ be the likelihood of the ***X*** interval. We assume the cepstral vectors in the interval are independent, and are characterized by a Gaussian mixture distribution consisting of 16 Gaussians. $\theta_X$ represents the parameters of this distribution, namely the 16 mixture coefficients, the 12-dimensional mean vector, and the covariance matrix, which we assume is diagonal. Let $L(\boldsymbol{Y}:\theta_Y)$ and $L(\boldsymbol{Z}:\theta_Z)$ be defined similarly. The likelihood $L_1$ that the two intervals are different audio types is then $L_1 = L(\boldsymbol{X}:\theta_X)\ L(\boldsymbol{Y}:\theta_Y)$. The likelihood that the two intervals are the same audio type is $L_0 = L(\boldsymbol{Z}:\theta_Z)$. Thus the likelihood ratio for testing the hypothesis that the intervals represent the same sound type is $\lambda_L = L_0/L_1$. This ratio is large when $H_0$ is true, that is when ***X*** and ***Y*** have the same statistics, and small otherwise. Thus the distance measure between intervals ***X*** and ***Y*** is set to $d(\boldsymbol{X},\boldsymbol{Y}) = -log(\lambda_L)$, which is large when ***X*** and ***Y*** are different.

The audio feature is computed at a rate of 5 times per second by sliding the intervals shown in Figure 1 by 10 cepstral vectors. Since the video difference is computed 30 times per second, we replicate each audio distance 6 times so that the audio feature is computed at the same rate.

## 2.3 Motion Features

The motion feature detects motion of objects between frames, and is useful for identifying camera movement such as pans and zooms. Motion vectors are computed using an exhaustive-search block-matching algorithm in a 24x24 window for nine evenly distributed 40x40 pixel blocks. Large blocks are used to minimize the effect of object motion. The presence of motion is detected using two features computed from the coherence of the motion vectors. The first feature is the magnitude of the average of the nine motion vectors. The second feature is the average magnitude of the nine motion vectors. The combination of the two features allows pans and zooms to be detected. When both features have a high value, a pan is indicated. A zoom occurs when the first feature is small but the second feature is large. When the second feature is small, no camera motion is occurring. Figure 2 shows some sample motion vector fields and the feature values that result

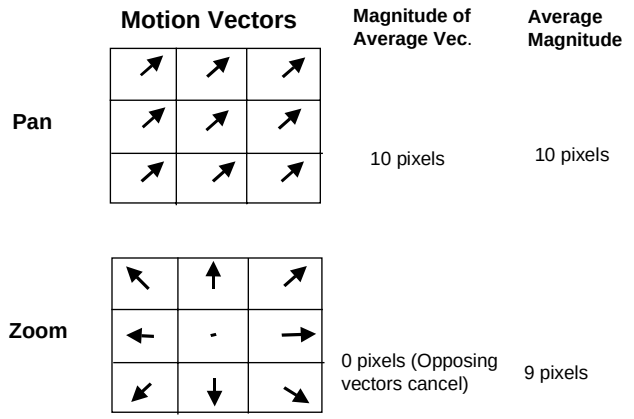


Figure 2. Motion features for pan and zoom.

# 3. HMM Segmentation

Figure 3 shows the hidden Markov model (HMM) used for video segmentation. The shot state models segments of the video within a single shot. We use separate states to model camera motion, namely pan and zoom. The other states model the transition segments between shots, namely cuts, fades, and dissolves. The arcs between states model the allowable progressions of segments. Thus from the shot state it is possible to go to any of the transition states, but from a transition state it is only possible to return to the shot state. This assures that only a single transition segment occurs between shots. Similarly, the pan and zoom state can only be reached from the shot state, since they are in fact subsets of the shot. The arcs from a state to itself model the length of time the video is in that particular state.

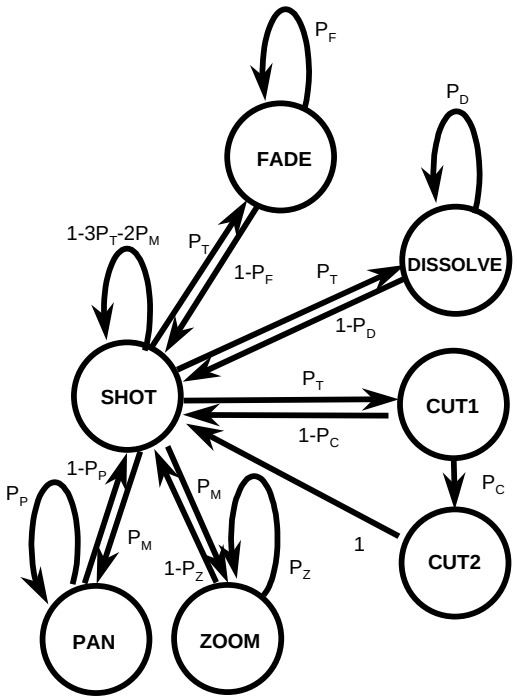


Figure 3. Hidden Markov model for video segmentation.

A probability is associated with each of the arcs in Figure 3. The probability $P_T$ is the probability that a transition occurs. We assume for simplicity that each of the transition types, namely cut, fade, and dissolve, are equally likely. The probability $P_M$ is the probability of camera motion, namely the probability of a pan or zoom. The probability $1\text{-}3P_T - 2P_M$ is the probability of staying in a shot. The probability $P_F$ is the probability of staying in the fade state, and models the duration of a fade. The probability $1\text{-}P_F$ is the probability of returning from a fade back to a shot. The other probabilities $P_D$, $P_Z$ and $P_P$ are defined similarly.

A cut is modeled by two states in the HMM, shown as CUT1 and CUT2 Figure 3. When an analog video consisting of fields is digitized into frames, the cuts may fall across two frames. Instead of producing one large frame-to-frame distance value, two lower values are produced. By explicitly modeling the noisy nature of some cuts, we avoid mislabeling these cuts as short gradual transitions. By using a two-state cut model rather than a self-transition, we correctly model the known duration of cuts.

Each state of the hidden Markov model (HMM) has an associated probability distribution that models the distribution of the image, audio, and motion features conditioned on the state. We assume that the features are independent, so that the joint distribution is the product of the probability distributions for each of the features. In this work, we use either a single Gaussian or a Gaussian mixture distribution for the state-conditional probability distributions. Since fewer parameters are required, a single Gaussian is used to model most state conditional feature distributions. One exception is the image feature distribution for the fade state, which uses a mixture of 2 Gaussians. This is because the histogram difference during a fade can have both large and small values.

The parameters of the HMM, namely the transition probabilities $P_T$, $P_M$, $P_F$, $P_C$, $P_D$, $P_Z$, and $P_P$, as well as the means and variances of the various Gaussian distributions, are learned during a training phase. Data for training consists of features computed for a collection of video, labeled according to whether there is a shot, a cut, a fade, a dissolve, a pan or a zoom. Given this data, a standard algorithm for training hidden Markov model parameters, namely Baum-Welch re-estimation [7], is applied. It is important that sufficient data, in terms of both quantity and variety, be used in training so that the resulting parameters can be applied to arbitrary video.

Once the parameters are trained, segmenting the video into its shots, camera motions, and transitions is performed using the Viterbi algorithm, a standard technique for segmentation and recognition with HMMs [7]. Given a sequence of features, the Viterbi algorithm produces the sequence of states most likely to have generated these features. The state sequence is time-aligned with the feature sequence, so that the video is segmented according to the times corresponding to shots, cuts, fades, dissolves, pans, and zooms.

# 4. Experiments

## 4.1 HMM vs Manual Threshold

The algorithm has been tested on portions of a video data base containing a variety of video, including television shows, news, movies, commercials, and cartoons [2]. The training data for our experiments was a six-minute cartoon with 64 cuts, 4 dissolves, and 12 fades, for a total of 80 transitions. The results are given using three measures. Recall is the percentage of shot boundaries that were correctly detected by the model. Precision is the percentage of claimed shot boundaries that are actually shot boundaries. Classify is the percentage of shot boundaries that are correctly labeled (e.g., cut or fade). We started with a single video feature, the gray-scale histogram difference. The first experiment used our video model without the pan and zoom states. The model was tested on the training data (Bugs), 30 minutes of a motion picture (Raiders), 30 minutes of a television news program with commercials (CNN), and 30 minutes of a television drama with commercials (Babylon). Table 1 shows the results of the model compared to using a simple threshold on the histogram difference feature. The threshold data results from selecting a threshold that produces a recall value close to that of the HMM. No classify data is given for the threshold method, which is unable to label gradual transitions.

It is clear from this experiment that for a fixed recall, our model gives higher precision than using a simple threshold selection. In addition, it is able to classify transition types using the simple histogram feature, and does not require tailoring to specific videos. Other video features that are useful in classifying gradual transitions would likely produce better results.

| | HMM | | | Threshold | |
|---|---|---|---|---|---|
| Data Set | Recall | Precision | Classify | Recall | Precision |
| Bugs | .950 | .809 | .950 | .938 | .636 |
| Raiders | .970 | .793 | .843 | .964 | .644 |

| | | | | | |
|---|---|---|---|---|---|
| CNN | .907 | .829 | .828 | .902 | .644 |
| Babylon | .928 | .865 | .878 | .928 | .727 |

Table 1: Comparison of HMM segmentation to threshold methods for the image difference feature.

### 4.2 Addition of Audio and Motion Features

Our second experiment used a feature vector consisting of the histogram feature and the audio feature combined with the video model without the pan and zoom states. Table 2 shows the results of applying this model to the training data and the Raiders data set. The model made use of the different audio characteristics of action sequences and gradual transitions to do a better job of classifying transitions than the simple image feature at the cost of slightly worse recall and precision.

Our third experiment used the full video model and a feature vector consisting of the histogram feature and the motion features. Table 2 shows the results of applying this model to the training data and the Raiders data set. This model was able to use the motion feature to avoid mistaking pans and zooms for gradual transitions and therefore produced a higher precision measure.

| | With Audio Feature | | | With Motion Feature | | |
|---|---|---|---|---|---|---|
| Data set | Recall | Precision | Classify | Recall | Precision | Classify |
| Bugs | .963 | .778 | .963 | .950 | .873 | .925 |
| Raiders | .964 | .708 | .901 | .970 | .843 | .828 |

Table 2: HMM segmentation results for image and audio and image and motion features.

## 5. SUMMARY

Use of hidden Markov models for video segmentation eliminates two standard problems. The first is the setting of thresholds for the frame-to-frame distances for detecting cuts and gradual transitions. In the past, this has been done manually. However, with HMMs these parameters can be learned automatically. The second problem is how to use multiple features, such as histogram differences, motion vectors, and audio differences, to aid in the video segmentation. The HMM framework allows any number of features to be included in a feature vector. Our results show that the HMM segmentation algorithm gives higher precision for similar recall values when used with the image difference feature alone. Additional features allow tradeoff between precision and classification accuracy.

While these techniques have been tested on a database of produced video, they also apply to raw, unedited video. We are applying this method to segment informal video taken to record usage studies and meetings [11]. While the detection of fades and dissolves does not apply to this type of data, the explicit modeling of pans and zooms allows segmentation based on camera movement.

## 6. REFERENCES


[1] Arman, F., Hsu, A., Chiu, M-Y., "Image Processing on Encoded Video Sequences", Multimedia Systems (1994) Vol. 1, No. 5, pp. 211-219.

[2] Boreczky, J. and Rowe, L., "Comparison of Video Shot Boundary Detection Techniques", Proc.SPIE Conference on Storage and Retrieval for Still Image and Video Databases IV, San Jose, CA, February, 1996, pp. 170-179.

[3] Gish, H., Siu, M., Rohlicek, R. "Segmentation of Speakers for Speech Recognition and Speaker Identification", Proc. Int. Conf. Acoustics, Speech, and Signal Processing, vol. 2, IEEE, Toronto, Canada, May 1991, pp. 873-876.

[4] Kasturi, R., Jain, R., "Dynamic Vision", in *Computer Vision: Principles*, Kasturi R., Jain R., Editors, IEEE Computer Society Press, Washington, 1991.

[5] Nam, J., Cetin, E., and Tewfik, A. "Speaker Identification and Video Analysis for Hierarchical Video Shot Classification", Proc. Int. Conf. Image Processing, Santa Barbara, CA, October, 1997.

[6] Phillips, M. and Wolf, W., "Video Segmentation Techniques for News," in Multimedia Storage and Archiving Systems, C.-C. Jay Kuo, Editor, Proc. SPIE 2916, 243-251 (1996).

[7] Rabiner, L, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. IEEE, vol. 77, No. 2, February 1989, pp. 257-285.

[8] Saraceno, C. and Leonardi, R., "Audio as a Support to Scene Change Detection and Characterization of Video Sequences", Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Munich, Germany, April 1997, pp. 2597-2600.

[9] Shahraray, B., "Scene Change Detection and Content-Based Sampling of Video Sequences", in Digital Video Compression: Algorithms and Technologies, Rodriguez, Safranek, Delp, Eds., Proc. SPIE 2419, Feb 1995, pp. 2-13.

[10] Wilcox, L., Kimber, D., Chen, F., "Audio Indexing Using Speaker Identification", Proc. SPIE Conference on Automatic Systems for the Inspection and Identification of Humans, San Diego, CA, July, 1994, pp. 149-157.

[11] Wilcox, L, and Boreczky, J., "Annotation and Segmention in Multimedia Indexing and Retrieval", to appear in HICSS, Jan 1998.

[12] Wolf, W., "Hidden Markov Model Parsing of Video Programs", Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Munich, Germany, April 1997, pp. 2609-2611.

[13] Zabih, R., Miller, J., Mai, K., "A Feature-based Algorithm for Detecting and Classifying Scene Breaks", Proc. ACM Multimedia 95, San Fransisco, CA, November, 1995, pp. 189-200.

[14] Zhang, H.J., Kankanhalli, A., Smoliar, S.W., "Automatic Partitioning of Full-motion Video", Multimedia Systems (1993) Vol. 1, No. 1, pp. 10-28.