Adaptive Speaker Identification with AudioVisual Cues for Movie Content Analysis

Ying Li, Shri Narayanan, and C.-C. Jay Kuo [a]

[a]Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-2564

E-mail:{yingli,shri,cckuo}@sipi.usc.edu

An adaptive speaker identification system which employs both audio and visual cues is proposed in this work for movie content analysis. Specifically, a likelihood-based approach is first applied for speaker identification using pure speech data, and techniques such as face detection/recognition and mouth tracking are applied for talking face recognition using pure visual data. These two information cues are then effectively integrated under a probabilistic framework for achieving more robust results. Moreover, to account for speakers' voice variations along time, we propose to update their acoustic models on the fly by adapting to their incoming speech data. An improved system performance (80% identification accuracy) has been observed on two test movies.

Keywords: Movie content analysis; Speech segmentation and clustering; Mouth detection and tracking; Adaptive speaker identification; Unsupervised speaker model adaptation

1. Introduction

A fundamental task in video analysis is to organize and index multimedia data in a meaningful manner so as to facilitate user's access such as browsing and retrieval. This work proposes to extract an important type of information, the *speaker identity*, from feature films for the content indexing and browsing purpose.

So far, a large amount of speaker identification work has been reported on standard speech databases. For instance, Chagnolleau et al. (1999) adopted a likelihood-based speaker detection approach to estimate target speakers' segments with Hub4 broadcast news, which is the benchmark test set provided by NIST (National Institute of Standards and Technology). However, while acceptable results were reported in its one-target-speaker case, the system performance degraded dramatically in two-target-speaker case. Similar work was also reported by Rosenberg et al. (1998) where the NBC Nightly News broadcasts were used. Johnson (1999) addressed the problem of labeling speaker turns by automatically segmenting and clustering a continuous audio stream. A frame-based clustering approach was proposed, and an accuracy of 70% was obtained on the 1996 Hub4 development data.

Recently, with the increase of the accessibility to other media sources, researchers have attempted to improve system performance by integrating knowledge from all available media cues. For instance, Tsekeridou and Pitas (2001) proposed to identify speakers by integrating cues from both speaker recognition and facial analysis schemes. This system is, however, impracticable for generic video types since it assumes there is only one face in each video frame. Similar work was also reported by Li et al. (2001), where TV sitcoms were used as test sequences. In (Li et al., 2002), a speaker identification system was proposed for indexing the movie content, where both speech and visual cues were employed. This system, however, has certain limitations since it only identifies speakers in movie dialogs.

From the other point of view, most existing work in this field deals with supervised identification problems, where speaker models are not allowed to change once they are trained. Two drawbacks arise when we apply supervised identification techniques to feature films.

1. *The insufficiency of training data.* A speaker's voice can have dramatic variations along time, especially in feature films. Therefore, a model built with limited training data (*e.g.* 40- or 50-second speech) cannot model a speaker well for a long video sequence. Moreover, to manually transcribe training data is a very time-consuming task.

2. *The decrease of system efficiency.* Because we have to go through movies at least once to collect and transcribe training data before the actual identification process can be started, it wastes time and decreases system efficiency.

An adaptive speaker identification system is thus proposed in this work which aims to offer a better solution to speaker identification for movie content analysis. Specifically, a set of initial speaker models are first constructed during the system initialization stage; then we keep updating them on the fly by adapting to speakers' newly incoming data. By doing so, we are able to capture the speakers' voice variations along time, thus to achieve a higher identification accuracy. Both audio and visual sources are exploited in the identification process, where the audio content is analyzed to recognize speakers using a likelihood-based approach, while the visual content is parsed to recognize talking faces using face detection/recognition and mouth tracking techniques.

The rest of the paper is organized as follows. Section 2 will elaborate on the proposed identification scheme which includes speech segmentation and clustering, mouth detection and tracking, audiovisual-based speaker identification and unsupervised speaker model adaptation. Experimental results obtained on two test movies are reported and discussed in Section 3. Finally, concluding remarks and possible future extensions are given in Section 4.

2. Adaptive speaker identification

Figure 1 shows the proposed system framework that consists of the following six major modules: (1) shot detection and audio classification, (2) face detection, recognition and mouth tracking, (3) speech segmentation and clustering, (4) initial speaker modeling, (5) audiovisual (AV)-based speaker identification, and (6) unsupervised speaker model adaptation. As shown, given an input video, we first split it into audio and visual streams, then perform a shot detection on the visual source. Following this, a shot-based audio classification is carried out which categorizes each shot into either environmental sound, silence, music, or speech. Next, with non-speech shots being discarded, all speech shots are further processed in the speech segmentation and clustering module where speeches from the same person are grouped into one cluster. Meanwhile, a face detection/recognition and mouth tracking process is also performed on speech shots for recognizing talking faces. Both of the speech and face cues are then effectively integrated to recognize speakers in the audiovisual-based identification module, under the assistance of either initial or updated speaker models. Finally, the identified speaker's model is updated in the unsupervised model adaptation module, which becomes effective in the next round of the identification process.

2.1. Shot Detection and Audio Classification

The first step towards visual content analysis is shot detection. In this work, a color histogram-based approach is employed to fulfill this task. Specifically, once a distinct peak is detected in the frame-to-frame histogram difference, we declare it as a shot cut (Li and Kuo, 2000).

In the second step, we analyze the audio content of each shot and classify it into one of the following four classes: *silence*, *speech* (including speech with music), *music*, and *environmental sound*. Five different audio features including short-time energy function, short-time average zero-crossing rate (ZCR), short-time fundamental frequency (SFuF), energy band ratio (EBR) and silence ratio (SR), are extracted for this purpose. Specifically, silence is detected by thresholding the energy and ZCR features; speech is recognized by exploiting the inter-relationship between energy, ZCR and SFuF features, as well as

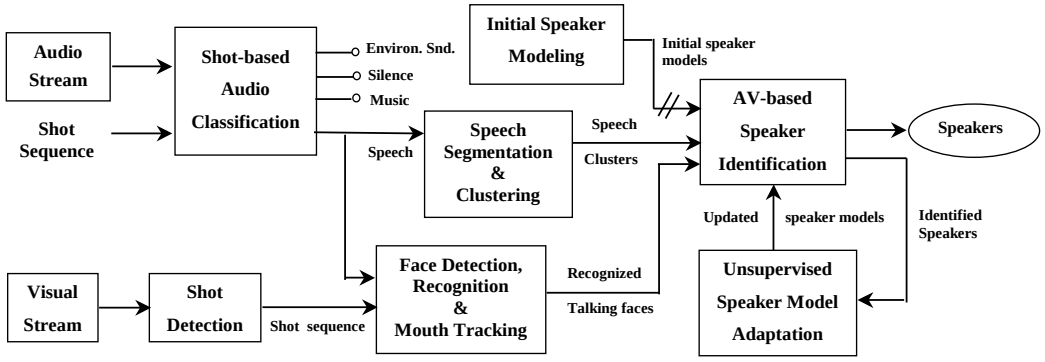


Figure 1. Block diagram of the proposed adaptive speaker identification system.

by checking EBR and SR values; finally, music is distinguished from other audio signals by measuring both ZCR and SFuF features (Zhang and Kuo, 1999).

2.2. Face Detection, Recognition and Mouth Tracking

The goal of this module is to detect and recognize talking faces in speech shots.

2.2.1. Face Detection and Recognition

The face detection and recognition library we used was developed by Hewlett-Packard Research Labs based on a neural network technology (HP, 1998). Currently, this library can only detect upright frontal faces or faces rotated by plus or minus 10 degrees from the vertical. To speed up the process, we will only carry out face detection on speech shots as shown in Figure 1. Figure 2(a) shows a detection example where the detected face is bounded by a rectangle and two eyes are indicated by crosses. To facilitate the subsequent recognition process, we organize the detection results into a set of face sequences, where all frames within each sequence contain the same number (nonzero) of human faces.

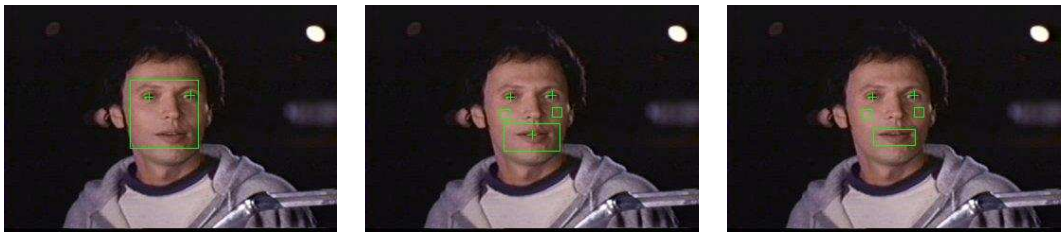
The face database used for face recognition is constructed as follows. During the system initialization, we first ask users to select their N interested casts (also called *target speakers*) by randomly choosing video frames that contain the casts' faces. These faces are then detected, associated with the casts' names, and added into the face database.

During the face recognition process, each detected face in the first frame of each face sequence is recognized. The result is returned as a face vector $\vec{f} = [f_1, \ldots, f_N]$, where f_i is a value in [0, 1] which indicates the confidence of being target cast i.

2.2.2. Mouth detection and tracking

At this step, we will first apply a weighted block matching approach to detect the mouth for the first frame of a face sequence, then we track it for the rest of its frames. Note that if more than two faces are present in the sequence, we will virtually split it into a number of sub-sequences with each focusing on one face.

There has been some work reported on mouth detection and tracking for video indexing and speaker detection purposes. For example, Tsekeridou and Pitas (2001) employed a mouth-template tracking process to assist in talking face detection. In (Sobottka and Pitas, 1998), a facial feature extracting and tracking system was reported, where features including eyebrows, eyes, nostrils, mouth and chin were determined by searching the minima and maxima in a topographic grey-level relief. A block matching mechanism was further applied to track the mouth in the sequence. This approach, however, is not



(a) (b) (c)

Figure 2. (a) A detected human face, (b) the coarse mouth center, the mouth search area, and two small squares for skin-color determination, and (c) the detected mouth region.

suitable for our research since the block matching method will not work well in talking mouth tracking.

1. Mouth detection

Figure 3 shows three abstracted face models where (a) gives a model of an upright face, and (b), (c) give models for rotated faces. According to the facial biometric analogies, we know that there is a certain ratio between the interocular distance and the distance *dist* between eyes and mouth. Thus, once we obtain the eyes positions from the face detector, which are denoted by (x_1, y_1) and (x_2, y_2) in the figure, we can subsequently locate the coarse mouth center (x, y) for an upright face.

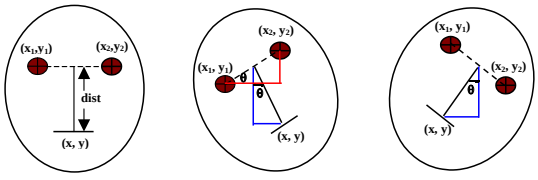


Figure 3. Three abstracted face models for (a): upright face, (b) and (c): rotated faces.

When a face is rotated as shown in Figure 3(b) or (c), we compute its mouth center as

$$\begin{cases} x = \frac{x_1+x_2}{2} \pm dist \times \sin(\theta), \\ y = \frac{y_1+y_2}{2} + dist \times \cos(\theta), \end{cases} \quad (1)$$

where θ is the head rotation angle.

We then expand the coarse mouth center (x, y) into a rectangular mouth search area as shown in Figure 2(b), and perform a weighted block-matching process to locate the target mouth. The intuition we used to detect the mouth is that it presents the largest color difference from the skin color, which is determined from the average pixel color in the two small under-eye squares as shown in Figure 2(b).

Now, denote the size of the mouth search area by (SW, SH), and the size of the desired mouth by (MW, MH), its upper-left corner is located as

$$(\hat{i}, \hat{j}) = \arg\max_{(i,j)} \left\{ c_{i,j} \times \sum_{w=i}^{i+MW} \sum_{h=j}^{j+MH} \text{dist}(w, h) \right\}, \quad (2)$$

where

$$dist(w, h) = dist(YUV(w, h), YUV(\text{skin-color})),$$

$0 \le i \le (SW - MW)$, and $0 \le j \le (SH - MH)$. $dist()$ is a plain Euclidean distance in YUV color space, and $c_{i,j}$ is a Gaussian weighting coefficient. An example of a correctly detected mouth is shown in Figure 2(c). For the rest of the discussion, we denote the detected mouth center by (cx, cy).

2. Mouth tracking

To track the mouth for the rest of frames, we assume that for each subsequent frame, the centroid of its mouth mask can be derived from that of the previous frame as well as from its eye positions. Moreover, we assume that the distance between the coarse mouth center (x, y) and the detected mouth center (cx, cy) remains the same for all frames.

Now, assume that we have obtained all feature data for frame f_i including the eye positions, the coarse and final mouth centers (x, y) and (cx, cy), frame f_{i+1}'s mouth centroid (cx', cy') can be computed as

$$\begin{cases} cx' = cx - (x - x'), \\ cy' = cy - (y - y'), \end{cases}$$

where x, x', y and y' are calculated via Equation (1).

Figure 4 shows the mouth detection and tracking results, as marked by rectangles, on a face sequence containing ten consecutive frames.

Finally, we apply a color histogram-based approach to determine if the tracked mouth is talking. Particularly, if the normalized accumulated histogram difference in the mouth area of the entire or part of the face sequence f exceeds a certain threshold, we label it as a talking mouth; and correspondingly, we mark sequence f as a talking face sequence.

2.3. Speech segmentation and clustering

For each speech shot, the two major speech processing tasks are speech segmentation and speech clustering. In the segmentation step, all individual speech segments are separated from the background noise or silence. In the clustering step, we group the same speaker's segments into homogeneous clusters so as to facilitate successive processes.

2.3.1. Speech segmentation

A general solution to separate speech from the background, or equivalently, to detect silence from the voice segment, is to apply a global energy/zero-crossing ratio thresholding scheme (Zhang and Kuo, 1999). However, while a global threshold works well on static audio content, it is not suitable for movies which have complex audio background.

In this work, we propose to determine the threshold by adapting it to dynamic audio content. Particularly, given the audio signal of a speech shot, we first segment it into 15ms-long audio frames, then we sort them into an array based on their energies computed in the dB scale. Next, for all frames whose energy values are greater than a preset threshold $Discard\_thresh$, we quantize them into M bins where bin_1 has the lowest and bin_M has the highest average energy. Now that both speech and silence signals are present in the shot, the threshold that separate them must be a value between these two extremes. Specifically, we determine it from the sums of the first and last three bins, as well as a predefined $Speech\_thresh$ that gives the minimum dB difference between the two signals. Currently, we set $Discard\_thresh$ to be 30.0, $Speech\_thresh$ to be 9.0, and M to be 10.

Next, a 4-state transition diagram is employed to separate speech segments from the background. As shown in Figure 5, the diagram has four states: *in-silence*, *in-speech*, *leaving-silence* and *leaving-speech*. Either in-silence or in-speech can be the starting state, and any state can be a final state. The input of this state machine is a sequence of frame energy, and the output is the beginning and ending frame indices of detected speech segments. The transition conditions are labeled on each edge with the corresponding actions described in parentheses. Here, $Count$ is a frame counter, E denotes the frame energy, and L indicates the minimum length of a silence or speech segment. As we can see, this state machine basically groups blocks of continuous silence/speech frames as a silence/speech segment while removing impulsive noises.

This algorithm works best when it is performed within a shot range since the background can be assumed to be quasi-stationary in this case. A segmentation example is shown in the upper part of Figure 8 where the detected speech segments are bounded by the passbands of a pulse curve. As we can see, all speech fragments have been successfully isolated from the impulse noise. The loud background sounds are removed as well.

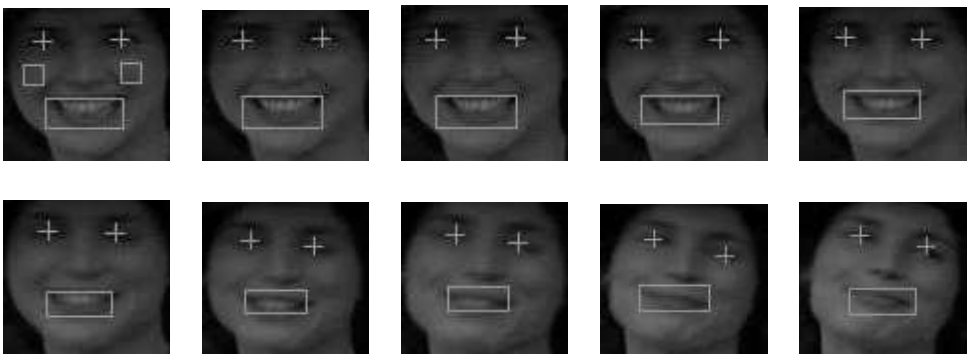


Figure 4. Mouth detection and tracking results on ten consecutive video frames, where eyes are indicated by crosses and mouthes are bounded by rectangles.

2.3.2. Speech clustering

Speech clustering has been studied for a long time, and many approaches have been proposed. Among them, the Kullback-Leibler distance metric, Generalized Likelihood Ratio criterion (GLR), Bayesian Information Criterion (BIC), GMM likelihood measure, and VQ distortion criterion are the most popularly applied (Siegler et al., 1997), (Mori et al., 2001), (Chen et al., 1998). In this work, we use BIC to measure the similarity between two speech segments.

When comparing two segments using the BIC, the distance measure can be stated as a model selection criterion where one model is represented by two separate segments X_1 and X_2, and the other model represents the joined segment $X = \{X_1, X_2\}$. The difference between these two modeling approaches is given by (Chen et al., 1998)

$$\Delta BIC(X_1, X_2) = \frac{1}{2}(M_{12} \log |\Sigma_{12}| - M_1 \log |\Sigma_1| - M_2 \log |\Sigma_2|) - \frac{1}{2}\lambda(d + \frac{1}{2}d(d+1)) \log M_{12}, \quad (3)$$

where Σ_1, Σ_2, Σ_{12} are X_1, X_2 and X's covariance matrices, and M_1, M_2, M_{12} are their feature vector numbers, respectively. λ is a penalty weight and equals 1 in this case. d gives the dimension of the feature space. According to the BIC theory, when $\Delta BIC(X_1, X_2)$ is negative, the two speech segments can be considered from the same speaker.

Now, assume cluster C contains n homogeneous speech segments, then given an incoming speech segment X, we compute their distance as

$$Dist(X, C) = \sum_{i=1}^{n} w_i \times \Delta BIC(X, X_i), \quad (4)$$

where w_i is the weighting coefficient of segment X_i, and is computed as $w_i = M_i / \sum_{j=1}^{n} M_j$. Finally, if $Dist(X, C)$ is less than 0, we merge X to cluster C; otherwise, if none of existing clusters is matched, a new cluster will be initialized.

2.4. Initial speaker modeling

To bootstrap the identification process, we need initial speaker models as shown in Figure 1. This is achieved by exploiting the inter-relation between the face and speech cues. Specifically, the following two steps are applied.

First, for each target cast A, we identify its 1-face shot where a 1-face shot is a speech shot that contains only 1 face in most of its frames. When multiple 1-face shots are identified for cast A, we choose the one having the highest face recognition rate and the longest shot length. Moreover, to make sure that the selected 1-face shot only contains A's speech, we can return it to the user for further verification.

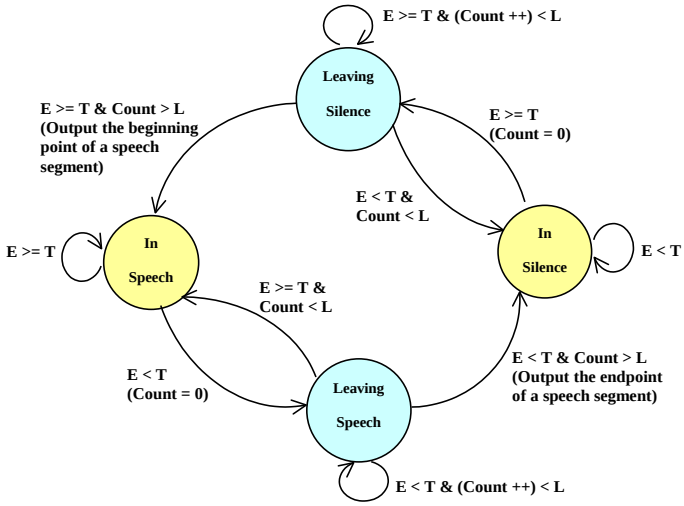


Figure 5. A state transition diagram for speech-silence segmentation, where T stands for the derived adaptive threshold, E denotes the frame energy, *count* is a frame counter and L indicates the minimum speech/silence segment length.

Next, we collect A's speech segments from its 1-face shot as described in Section 2.3.1, and build its initial model. Currently, the Gaussian Mixture Models (GMM) are employed to model speakers which will be introduced in the next section. Note that at this stage, each model will only contain one Gaussian component with its mean and covariance computed as the global ones due to the limitation of training data.

2.5. Likelihood-based speaker identification

At this stage, we will identify speakers based on pure speech information. Specifically, given a speech signal, we first decompose it into a set of overlapped audio frames; then 14 Mel-frequency cepstral coefficients (MFCC) are extracted from each frame to form an observation sequence X. Next, we calculate the likelihood $L(X; M_i)$ between X and all speaker models M_i based on the multivariate analysis. Finally, we obtain a speaker vector $\vec{v}$ with its ith component indicating the confidence of being target speaker i.

2.5.1. Likelihood calculation

Due to its successful usage in both speech and speaker recognition, MFCC coefficients are chosen as the audio features in this work. Moreover, considering that there are various noises in movie data, we have also performed a cepstral mean normalization on all cepstral coefficients (Young et al., 2000).

To model speakers, we choose to use GMM (Gaussian Mixture Model) since the Gaussian mixture density can provide a smooth approximation to the underlying long-term sample distribution of a speaker's utterances (Reynolds and Rose, 1995). A GMM model M can be represented by the notation $M = \{p_j, \vec{\mu}_j, \Sigma_j\}, j = 1, \ldots, m$. where m is the total number of components in M, and p_j, $\vec{\mu}_j$, Σ_j are the weight, mean vector and covariance matrix of the ith component, respectively.

Now, let M_i be the GMM model corresponding to the ith enrolled speaker with $M_i = \{p_{ij}, \vec{\mu}_{ij}, \Sigma_{ij}\}$, and let X be the observation sequence consisting of T cepstral vectors $\vec{x}_t$, $t = 1, \ldots, T$, under the assumption that all observa-

tion vectors are independent, the likelihood between X and M_i can be computed as

$$L(X; M_i) = \sum_{j=1}^{m} p_{ij} \times L(X; \vec{\mu}_{ij}, \Sigma_{ij}), \quad (5)$$

where $L(X; \vec{\mu}_{ij}, \Sigma_{ij})$ is the likelihood of X's belonging to M_{ij}. Based on the multivariate analysis in Mardia et al. (1979), the log likelihood of $L(X; \vec{\mu}_{ij}, \Sigma_{ij})$ can be computed as

$$\begin{aligned} \ell(X; \vec{\mu}_{ij}, \Sigma_{ij}) &= -\frac{T}{2} \log |2\pi\Sigma_{ij}| - \frac{T}{2} tr(\Sigma_{ij}^{-1} S) \\ &- \frac{T}{2} (\bar{X} - \vec{\mu}_{ij})' \Sigma_{ij}^{-1} (\bar{X} - \vec{\mu}_{ij}), \end{aligned}$$

where S and $\bar{X}$ are X's covariance and mean, respectively. Note that when S equals Σ_{ij}, the above log likelihood becomes the Mahalonobis distance.

Now, based on this identification scheme, given any speech cluster C, we will assign it a speaker vector $\vec{v} = [v_1, \ldots, v_N]$, where v_i is a value in [0, 1] which equals the normalized log likelihood value $\ell(X; \vec{\mu}_{ij}, \Sigma_{ij})$, and indicates the confidence of being target speaker i.

2.6. Audiovisual integration for speaker identification

In this step, we will finalize the speaker identification task for cluster C (in shot S) by integrating the audio and visual cues obtained in Sections 2.2, 2.3 and 2.5. Specifically, given cluster C and all recognized talking face sequences F in shot S, we examine if there is a temporal overlap between C and any sequence F_i. If yes, and also the overlap ratio exceeds a preset threshold, we assign F_i's face vector $\vec{f}$ to C; otherwise, we set its face vector to null. However, if C is overlapped with multiple F_i due to speech clustering or talking face detection errors, we choose the one with the highest overlap ratio. Finally, to accommodate for detection errors, we can extend the talking face sequence to both ends by a certain number of frames during the overlap checking. A slightly better performance has been gained from this extra effort.

Now, we determine the speaker's final identity in cluster C as

$$speaker(C) = \arg \max_{1 \le j \le N} (w_1 \cdot f[j] + w_2 \cdot v[j]), \quad (6)$$

where $\vec{f}$ and $\vec{v}$ are C's face and speaker vectors, respectively. w_1 and w_2 are two weights that sum up to 1.0. Currently we set them to be equal in the experiment. In fact, instead of choosing the top speaker in the above equation, we can also obtain a sorted list of possible speakers for cluster C, which can be used to smooth the identification results over neighboring shots.

The detected speaker's model is then updated using his or her current speech data as will be detailed in the next section. However, one thing worth mentioning here is that if the current shot contains music (*i.e.* it is a speech with music shot), then no model adaptation is carried out since otherwise, it will corrupt the model.

2.7. Unsupervised Speaker Model Adaptation

Now, after we identify speaker P for cluster C, we will update P's model using C's data in this step. Meanwhile, a background model will be either initialized or updated to account for all non-target speakers. Specifically, when there is no a priori background model, we use C's data to initialize it if the minimum of $L(C; M_i), i = 1, \ldots, N$ is less than a preset threshold. Otherwise, if the background model produces the largest likelihood, we denote the identified speaker as "unknown", and use C's data to update the background model.

The following three approaches have been investigated to update the speaker model: *Average-based model adaptation*, *MAP-based model adaptation*, and *Viterbi-based model adaptation*.

2.7.1. Average-based model adaptation

With this approach, the P's model is updated in the following three steps.

Step 1: Compute the BIC distances between cluster C and all of P's mixture component b_i. Denote the component that gives the minimum distance as b_0.

Step 2: If the minimum distance is less than an empirically determined threshold, we consider C to be acoustically close to b_0, and use C's data to update this component. Specifically, let $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$ be C and b_0's Gaussian models, respectively, we update b_0's mean and

covariance as (Mokbel, 2001)

$$\mu_2' = \frac{N_1}{N_{12}}\mu_1 + \frac{N_2}{N_{12}}\mu_2, \quad (7)$$

$$\Sigma_2' = \frac{N_1}{N_{12}}\Sigma_1 + \frac{N_2}{N_{12}}\Sigma_2 + \frac{N_1 N_2}{N_{12}^2}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T, \quad (8)$$

where N_1 and N_2 are the number of feature vectors in C and b_0, respectively. In practice, because the mean can be easily biased by different environment conditions, the third item in Equation (8) is actually discarded in our implementation, which results in a slightly better performance.

Otherwise, if the minimum distance is larger than the threshold, we will initialize a new mixture component for P, with its mean and covariance equaling to μ_1 and Σ_1. However, once the total number of P's components reaches a certain value, which is currently set to be 32, no more components will be initialized and only component adaptation is allowed. This is adopted to avoid having too many Gaussian components in one model.

Step 3: Update the weight p_i for each of P's mixture component. Specifically, p_i is proportional to the number of feature vectors contributing to component b_i.

2.7.2. MAP-based model adaptation

MAP adaptation has been widely and successfully used in speech recognition, yet it has not been well explored in speaker recognition. In this work, due to the limited speech data, only Gaussian means will be updated. Specifically, given P's model M_p, we update its component b_i's mean via

$$\mu_i' = \frac{L_i}{L_i + \tau}\bar{\mu} + \frac{\tau}{L_i + \tau}\mu_i, \quad (9)$$

where τ defines the "adaptation speed", and is currently set to 10.0. L_i gives the occupation likelihood of the adaptation data to component b_i, and is defined as

$$L_i = \sum_{t=1}^{T} p(i|\vec{x}_t, M_p), \quad (10)$$

where $p(i|\vec{x}_t, M_p)$ is the *a posteriori* probability of $\vec{x}_t$ to b_i, and is computed as

$$p(i|\vec{x}_t, M_p) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{m} p_k b_k(\vec{x}_t)}. \quad (11)$$

Finally, $\bar{\mu}$ in Equation (9) gives the mean of observed adaptation data, and is defined as

$$\bar{\mu} = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, M_p)\vec{x}_t}{\sum_{t=1}^{T} p(i|\vec{x}_t, M_p)}. \quad (12)$$

Unlike the previous method, this MAP adaptation is applied to every component of P based on the principle that every feature vector has a certain possibility of belonging to every component. Thus, MAP adaptation provides a *soft* decision on which feature vector belongs to which component.

Now that we can no longer say which feature vector occupies which component, we must define a new way to update the number of feature vectors belonging to each component. Specifically, if cluster C contains M frames, then the number of b_i's feature vectors shall be increased by $M \times L_i / \sum_{j=1}^{m} L_j$.

2.7.3. Viterbi-based Model Adaptation

Widely used in speech recognition, the Viterbi algorithm performs an alignment of training observations to the mixture component that gives the highest probability within a certain state (Young et al., 2000). As a result, every observation will be associated with one single mixture component. This is the key concept that we employ in this approach.

Similar to the MAP-based approach, this approach also allows different feature vectors belonging to different components. Nevertheless, while the MAP approach gives a *soft* decision, this approach implies a *hard* one, *i.e.* for any one particular feature vector $\vec{x}_t$, it will either occupy component b_i or not. Therefore, the probability function $p(i|\vec{x}_t, M_p)$ in Equation (10) will now be replaced by an indicator function which is either 0 or 1. Thus given any feature vector $\vec{x}_t$, the mixture component it occupies will be determined by

$$m_0 = \arg \max_{1 \le i \le m} p(i|\vec{x}_t, M_p). \quad (13)$$

Finally, we use Equations (7) and (8) to update P's components after we assign every feature vector to its belonged component. Clearly, this approach is a compromise between the previous two methods.

To summarize, based on the proposed model adaptation approaches, a speaker model will grow from 1 Gaussian mixture component up to 32 components as we go through the entire movie sequence. Strictly speaking, the GMMs generated in this way are not the same as original GMMs, and they are also less accurate than the models trained using the EM (Expectation Maximization) algorithm. However, compared to EM, this approach is computationally simpler. Moreover, since no iteration is needed, it can better meet the real-time processing goal.

3. Experimental results

To evaluate the performance of the proposed adaptive speaker identification system, studies have been carried out on two movies, each of which is approximately 1-hour long. The first movie is a comedic drama (*"When Harry Met Sally"*) with many conversation scenes while the second one is a tragic romance (*"The Legend of the Fall"*) with fewer dialogs but more background music.

3.1. Results for Movie 1

Three interested characters were chosen for Movie 1, and totally 952 speech clusters were generated. An average of 90% clustering purity was achieved which is defined as the ratio between the number of segments from the dominant speaker and the total number of segments in a cluster.

Regarding the talking face detection, we have achieved 83% precision and 88% recall rates on 425 detected face sequences. However, the ratio of talking face-contained frames over the total number of video frames is as low as 11.5%. This is because movie casts are always in constant moving status, thus making it difficult to detect their faces.

The identification results for all obtained speech clusters are reported in the form of a confusion matrix as shown in Table 1. The three speakers are indexed by A, B, C, and their corresponding movie characters are denoted by A', B' and C'. "Unknown" is used for all non-target speakers. The number in each grid, say grid (A', B), indicates the number of speech segments where character A' is talking yet actor B is identified. Obviously, the larger the number in the diagonal, the better the performance. Three parameters, namely, *false acceptance* (FA), *false rejection* (FR) and *identification accuracy* (IA) are calculated to evaluate the system performance. Particularly, for each cast or character, we have

$$FR = \frac{\text{sum of off-diagonal numbers in the row}}{\text{sum of all numbers in the row}},$$

$$FA = \frac{\text{sum of off-diagonal numbers in the column}}{\text{sum of all numbers in the column}},$$

$$IA = 1 - FR.$$

Table 1(a) gives the identification result when the average-based model adaptation is applied. An average of 75.3% IA and 22.3% FA are observed. Result obtained from the MAP-based approach is given in Table 1(b) where we have an average 78.6% IA and 21% FA. This result is slightly better than that in (a), yet at the cost of a higher computation complexity. The performance improvement of applying MAP adaptation to speaker recognition has also been reported by Ahn et al. (2000), where a speaker verification was carried out on a Korean speech database.

Table 1(c) shows the result for the Viterbi-based approach. As we can see, this table presents the best performance with an average 82% IA and 20% FA. The fact that this approach outperforms the MAP approach may imply that, for speaker identification, a hard decision would be good enough.

By carefully studying the results, we found two major factors that degrade the system performance: (a) imperfect speech segmentation and clustering, and (b) inaccurate facial analysis results. Due to the various sounds/noises existing in movies, it is extremely difficult to achieve perfect speech segmentation and clustering results. Besides, incorrect facial data can result in mouth detection and tracking errors, which will further affect the identification accuracy.

Table 1
Adaptive speaker identification results for Movie 1 using: (a) the average-based, (b) the MAP-based, and (c) the Viterbi-based model adaptation approaches.

	A	B	C	Ukn	FR	IA
A'	228	42	6	32	26%	74%
B'	37	281	35	24	25%	75%
C'	10	9	115	16	23%	77%
Ukn	10	15	4	88		
FA	20%	19%	28%			

(a)

	A	B	C	Ukn	FR	IA
A'	239	25	22	22	22%	78%
B'	59	302	5	11	20%	80%
C'	10	8	117	15	22%	78%
Ukn	19	16	7	75		
FA	27%	14%	22%			

(b)

	A	B	C	Ukn	FR	IA
A'	246	29	13	20	20%	80%
B'	41	317	13	6	16%	84%
C'	10	10	123	7	18%	82%
Ukn	18	22	14	63		
FA	22%	14%	24%			

(c)

Moreover, to examine the robustness of the three set of speaker models (denoted by AVG, MAP and VTB) obtained from the three adaptation processes, we carried out a supervised speaker identification based on these models. The identification results are shown in Table 2, and a slightly degraded system performance is observed. This is because, when generating these models during the adaptation process, we have gradually adapted them to the later part of the movie data. Thus, when we apply them in supervised speaker identification, they may not model speakers well for the entire movie. Nevertheless, this table still presents acceptable results, especially for the Viterbi-based approach which presents 80% IA and 23% FA. These results are also comparable to those reported by other supervised identification work in an adverse audio environment (Chagnolleau et al., 1999), (Yu and Gish, 1993).

Table 2
Supervised speaker identification results.

Model set	IA			FA		
	A'	B'	C'	A	B	C
AVG	68%	74%	70%	23%	21%	33%
MAP	71%	81%	72%	25%	21%	26%
VTB	74%	90%	76%	24%	17%	28%

To determine the upper limit of the number of mixture components in each speaker model, we examined the average identification accuracy in terms of 32 and 64 components for all three adaptation methods and plotted them in Figure 6(a). As shown, except for the average-based method where a similar performance is observed, the use of 32 Gaussian mixture components has produced a better performance.

Finally, the average identification accuracy obtained by using or without using face cues is compared in Figure 6(b). Clearly, without the assistance of face cue, the system performance has been significantly degraded, especially for the average-based adaptation approach. This indicates that the face cue plays an important role in model adaptation.

3.2. Results for Movie 2

We also tested our algorithms on Movie 2, which is a tragic romance with fewer dialogs. Four casts were chosen as target speakers, and the speech clustering process resulted in 738 speech clusters. An average 83% clustering purity was achieved.

As for the talking face detection, similar precision and recall rates were achieved. However, the percentage of talking face-contained frames is only around 3.6% in this movie. Therefore, compared to Movie 1, this time we got less help from the face cue.

The identification results for all three adapta-

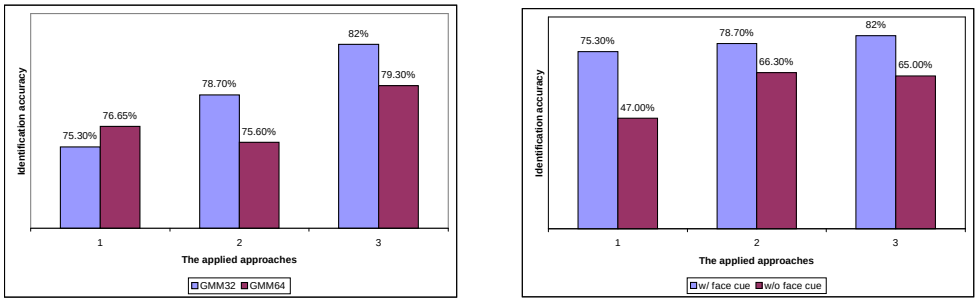


Figure 6. Identification accuracy comparison for the average-based, the MAP-based, and the Viterbi-based approaches with: (a) 32-component vs. 64-component for speaker models, and (b) using vs. without using face cues.

tion approaches are tabulated in Table 3. As shown, they have yielded similar results. However, the system performance has significantly degraded from that in Table 1 due to the following reasons: 1) the fewer dialogs in Movie 2 have made the identification errors more costly since now the target speakers have much fewer speech data; 2) the frequent use of background music in this movie brings a big challenge to our speech segmentation and clustering scheme; and 3) the casts' strong emotions have resulted in a wide variation in their talking rates, volumes and other related acoustic characteristics, which also brings a challenge to the current system.

In order to find the optimal number of Gaussian mixture components for initial speaker models, we examined the average identification accuracy in terms of 1-component, 2-component and 4-component for all three adaptation methods and plotted them in Figure 7(a). As shown, for all three cases, the use of 1 component gives the best performance. This means that when the amount of training data is very limited, *e.g.* the speech collected from one single shot, the use of multiple components tends to bring worse performance.

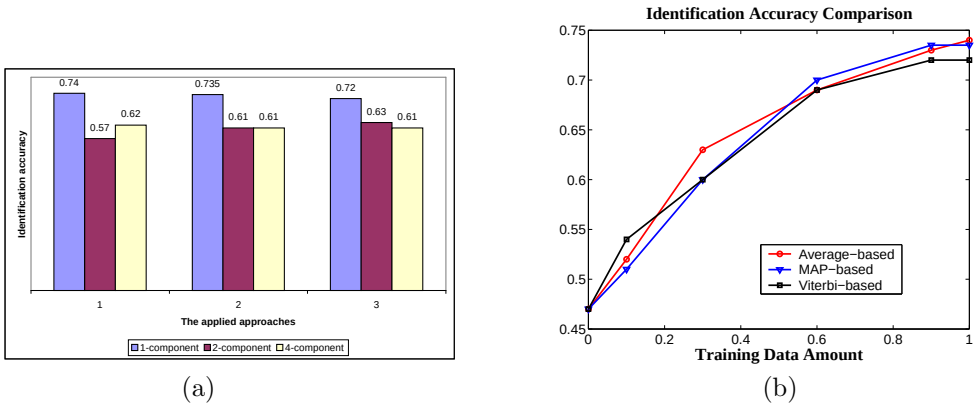
Another experiment was carried out to examine the performance change when the speaker models were updated with different amount of training data. Specifically, we measured the identification accuracy when the first 0%, 10%, 30%, 60%, 90%, and 100% of the entire movie data were used to update the models. The system performance evolvement for the average-based, MAP-based and Viterbi-based approaches are plotted in Figure 7(b) using circle-, triangle- and square-marked curves, respectively.

We see from this figure that all curves have shown a continuous performance improvement with the increase of the training data volume. This indicates the effectiveness of the proposed adaptive identification scheme. Moreover, we observe that there is a significant performance increase when the data amount rises from 10% to 30%, and from 30% to 60%, for all three cases. However, the identification accuracy remains relatively stable when the data amount increases from 90% to 100%. Similarly, the performance gain is relatively minor when the speaker models are updated using the first 10% of data, except for the Viterbi-based approach. This is because that, the initial speaker models are trained using the data collected from randomly distributed shots, thus they may not work well at the very beginning of the movie.

Figure 8 gives a detailed description of a speaker identification example. Specifically, the upper part shows the waveform of an audio signal recorded from a speech shot where two speakers take turns to talk. The superimposed pulse curve illustrates the speech-silence separation result

Table 3
Adaptive speaker identification results for Movie 2 using: (a) the average-based, (b) the MAP-based, and (c) the Viterbi-based model adaptation approaches.

Method	IA				FA			
	A'	B'	C'	D'	A	B	C	D
AVG-based	73%	80%	79%	64%	27%	43%	26%	13%
MAP-based	66%	78%	80%	70%	30%	34%	19%	28%
VTB-based	63%	82%	79%	65%	26%	36%	28%	32%



(a) (b)

Figure 7. Comparison of identification accuracy for the average-based, the MAP-based, and the Viterbi-based approaches with: (a) 1, 2 and 4 components in initial speaker models, and (b) different amount of model training data.

where all detected speech segments are bounded by the passband of the curve. The speaker identity for each speech segment is given right below this sub-figure, where speakers A and B are represented by dark- and light-colored blocks, respectively. The likelihood-based speaker identification result is given right below the ground truth, where only the toppest speaker's identity is shown. Besides, since two of the speech segments are too short (indicated by the circles) for speaker identification, we will disregard them in later processes. As shown, there are two false alarms in this result where B is falsely recognized as A twice. The talking face detection result is shown in the next sub-figure including both talking and non-talking cases. Finally, the last sub-figure shows the ultimate identification result obtained by integrating both speech and face cues as discussed in Section 2.6. As shown, although the first error still exists as the face cue cannot offer help, the second error has been corrected.

4. CONCLUSION AND FUTURE WORK

In this research, an adaptive speaker identification system was proposed which recognizes speakers in a real-time fashion for movie content indexing and annotation purposes. Both audio and visual cues were exploited where the audio source was analyzed to recognize speakers using a likelihood-based approach, and the visual source was parsed to recognize talking faces using face detection/recognition and mouth tracking techniques. Moreover, to better capture speakers'

voice variations along time, we update their models on the fly. Extensive experiments were carried out on two test movies of different genres, and encouraging results have been achieved.

As our future work, we plan to continue our research on face detection and tracking module since the current face detector has certain difficulty in locating non-upright faces which turns out to be very common in feature films. Moreover, the performance of the speech clustering module needs to be further improved.

References

Chagnolleau, I., Rosenberg, A., Parthasarathy, S., Detection of target speakers in audio databases, ICASSP'99, Phoenix, 1999.

Rosenberg, A., Chagnolleau, I., Parthasarathy, S., Huang, Q., Speaker detection in broadcast speech databases, ICSLP'98, 1339-1342, Sydney, Australia, 1998.

Johnson, S., Who spoke when ? - automatic segmentation and clustering for determining speaker turns, Eurospeech'99, 1999.

Yu, G., Gish, H., Identification of speakers engaged in dialog, ICASSP'93, 383-386, 1993.

Tsekeridou, S., Pitas, I., Content-based video parsing and indexing based on audio-visual interaction, IEEE Transactions on Circuits and Systems for Video Technology, 11(4), 522-535, 2001.

Li, D., Wei, G., Sethi, I., Dimitrova, N., Person identification in TV programs, Journal of Electronic Imaging, 10(4), 930-938, 2001.

Li, Y., Narayanan, S., Kuo, C., Identification of speakers in movie dialogs using audiovisual cues, ICASSP'02, Orlando, May 2002.

Li, Y., Kuo, C., Real-time segmentation and annotation of MPEG video based on multimodal content analysis I & II, Technical Report, University of Southern California, 2000.

Zhang, T., Kuo, C., Audio-guided audiovisual data segmentation, indexing and retrieval, Proc. of SPIE, 3656, 316-327, 1999.

HP Labs, Computational Video Group, The HP face detection and recognition library, User's Guide and Reference Manual, Version 2.2, December 1998.

Sobottka, K., Pitas, I., A novel method for automatic face segmentation, facial feature extraction and tracking, Image Communication, 12(3), 263-281, 1998.

Chen, S., Gopalakrishnan, P., Speaker, environment and channel change detection and clustering via the Bayesian information criterion, Proc. of DARPA Broadcast News Transcription and Understanding Workshop, 1998.

Siegler, M., Jain, U., Raj, B., Stern, R., Automatic segmentation, classification, and clustering of broadcast news, Proc. of Speech Recognition Workshop, Chantilly, Virginia, February 1997.

Mori, K., Nakagawa, S., Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition, ICASSP'01, 2001.

Reynolds, D., Rose, R., Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Transactions on Speech and Audio Processing, 3(1), 72-83, 1995.

Mardia, K., Kent, J., Bibby, J., Multivariate Analysis, Academic Press, San Diego, 1979.

Mokbel, C., Online adaptation of HMMs to real-life conditions: A unified framework, IEEE Transactions on Speech and Audio Processing, 9(4), 342-357, May 2001.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., HTK Book, Version 3.0, Downloaded from http://htk.eng.cam.ac.uk/index.shtml, July 2000.

Ahn, S., Kang, S., Ko, H., Effective speaker adaptations for speaker verification, ICASSP'00, 2, 1081-1084, 2000.

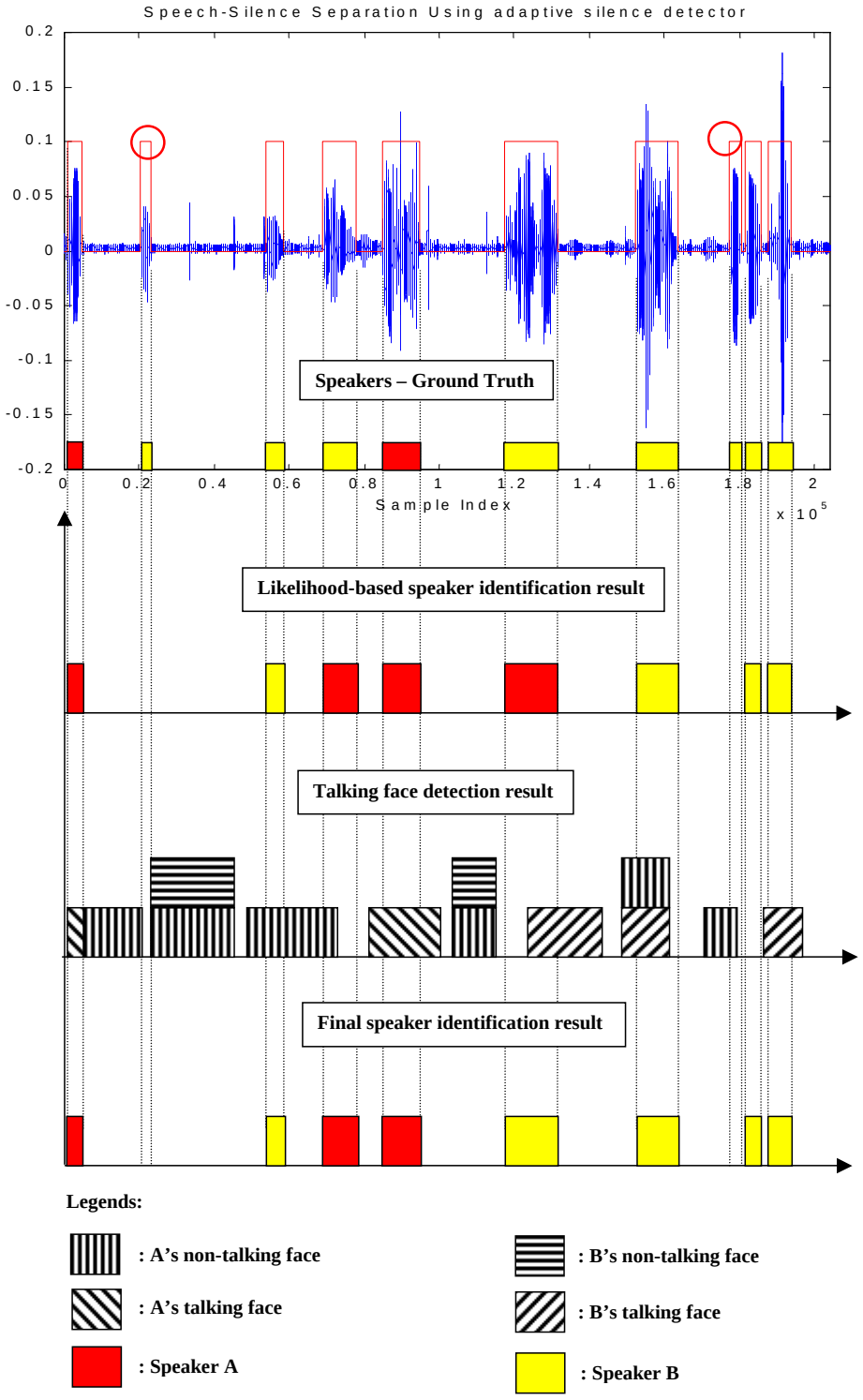


Figure 8. A detailed description of a speaker identification example.