PAPER Special Section on Robust Speech Processing in Realistic Environments

Evaluation of a Noise-Robust Multi-Stream Speaker Verification Method Using F_0 Information

Taichi ASAMI[†*a)], *Nonmember*, **Koji IWANO**[†], *Member*, *and* **Sadaoki FURUI**[†], *Fellow*

SUMMARY We have previously proposed a noise-robust speaker verification method using fundamental frequency (F_0) extracted using the Hough transform. The method also incorporates an automatic stream-weight and decision threshold estimation technique. It has been confirmed that the proposed method is effective for white noise at various SNR conditions. This paper evaluates the proposed method in more practical in-car and elevator-hall noise conditions. The paper first describes the noise-robust F_0 extraction method and details of our robust speaker verification method using multi-stream HMMs for integrating the extracted F_0 and cepstral features. Details of the automatic stream-weight and threshold estimation method for multi-stream speaker verification framework are also explained. This method simultaneously optimizes stream-weights and a decision threshold by combining the linear discriminant analysis (LDA) and the Adaboost technique. Experiments were conducted using Japanese connected digit speech contaminated by white, in-car, or elevator-hall noise at various SNRs. Experimental results show that the F_0 features improve the verification performance in various noisy environments, and that our stream-weight and threshold optimization method effectively estimates control parameters so that FARs and FRRs are adjusted to achieve equal error rates (EERs) under various noisy conditions.

key words: *speaker verification, F_0 information, multi-stream HMMs, stream-weight and threshold optimization, Adaboost*

1. Introduction

Increasing noise-robustness is one of the key issues for constructing real-world speaker verification systems. Since F_0 features are less sensitive to channel distortions or additive noise than spectral features, they are expected to be useful for increasing the robustness of speaker recognition. Various methods using fundamental frequency (F_0) in combination with spectral features have been proposed to achieve high-performance speaker recognition systems [1]–[9]. Carey et al. proposed a robust speaker recognition method using F_0 features to cope with the effect of handset variation on telephone speech [2]. Kyung and Lee showed that F_0 features increased robustness of VQ-based speaker identification against additive noise [4]. We have proposed a noise-robust speaker verification method using multi-stream HMMs for integrating F_0 and spectral information [10]**. The F_0 extraction method and multi-stream HMM construction method used in the verification scheme are based on our previous work on noise-robust speech recognition [12],[13]. The Hough transform [14], a robust image processing technique, was used for reliably extracting F_0 values. In [10], we have confirmed that the F_0 features yield improvement of speaker verification performance in white noise at various SNR conditions.

In order to construct practical multi-stream speaker verification systems, optimum system parameters, such as stream-weights of multi-stream HMMs and decision thresholds, need to be estimated before the verification process [15],[16]. We have proposed an automatic stream-weights and threshold estimation method for multi-stream speaker verification [17], using the linear discriminant analysis (LDA) [18] and the Adaboost technique [19]. In this method, the stream-weights and the threshold are automatically optimized according to the noise conditions of a development set. Since the optimum threshold of the multi-stream speaker verification is variable according to the setting of stream-weights, the threshold and the stream-weights need to be simultaneously estimated. The threshold is estimated so that the false acceptance rate (FAR) is equal to the false rejection rate (FRR) in the proposed method. In [17], we have confirmed that this method can accurately optimize the stream-weights and the threshold in white noise at various SNR conditions.

In this paper, additional experiments in more practical in-car and elevator-hall noise conditions are conducted for evaluating performance of our proposed multi-stream speaker verification method incorporating an automatic stream-weights and threshold optimization technique.

This paper is organized as follows. Section 2 explains a speaker verification method using multi-stream HMMs, which integrates spectral and noise-robust F_0 features. In Sect. 3, an automatic stream-weight and threshold optimization method based on the LDA and the Adaboost is explained. Experimental results are presented in Sect. 4, and Sect. 5 concludes this paper.

2. Speaker Verification Using F_0 Information

We have proposed a noise-robust F_0 extraction method and an effective method for integrating spectral and F_0 information, and effectiveness of these methods has been confirmed for speech recognition [11],[12]. These methods have also been implemented for speaker verification [10]. Details of

Manuscript received June 29, 2007.
Manuscript revised September 15, 2007.

[†]The authors are with the Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152–8552 Japan.
[*]Presently, with Cyber Space Laboratories, Nippon Telegraph and Telephone Corporation.
a) E-mail: asami.taichi@lab.ntt.co.jp

DOI: 10.1093/ietisy/e91–d.3.549

**Wark et al. used multi-stream HMMs for fusing audio and visual information in multi-modal speaker recognition, and confirmed its effectiveness [11].

the methods are described below.

2.1 Noise-Robust F_0 Extraction Method Using Hough Transform

Cepstral peaks extracted independently for each short period of speech have been widely used to extract F_0 values. This method often causes errors, including half pitch, double pitch and drop outs, for noisy speech. Since F_0 contours have temporal continuity in voiced periods, the Hough transform [14], taking advantage of its continuity, applied to time-cepstrum images is expected to have robustness in extracting pitch in the noisy environment. The Hough transform is an image processing technique to robustly extract parametric patterns, such as lines, and ellipses, from a noisy image [20].

Speech waveforms are sampled at 16 kHz and transformed to a sequence of 256-dimensional cepstral vector. A 32 ms-long Hamming window is used to extract each frame at every 10 ms. A nine-frame moving window is applied at every frame interval to extract a time-cepstrum image used for line information detection. The time-cepstrum image is used as the pixel brightness image for the Hough transform. An F_0 value is obtained from a cepstrum index at the center point of the detected line. Since the moving window has nine frames, time continuity over 90 ms is taken into account in this method. More details of the F_0 extraction method is explained in [13].

By using this method, F_0 extraction error can be significantly reduced in comparison with the conventional method in which cepstral peaks are chosen at each frame independently [13]. It has also been confirmed that the Hough transform-based F_0 extraction method yields better performance than the ESPS get_f0 function† in various noise conditions [21]. Although the Hough transform-based method requires 10 times more computational cost than the conventional method, the computational cost can be reduced by adjusting the threshold value to control the number of points to be used for the transformation.

2.2 Noise-Robust Speaker Verification Using Multi-Stream HMMs

2.2.1 Japanese Connected Digit Speech

The proposed method is evaluated using four-connected-digit speech in Japanese. In Japanese connected digit speech, two consecutive digits usually make one prosodic phrase. Figure 1 shows an example of an F_0 contour of four-connected-digit speech. The first two digits, "1" and "4", make the first prosodic phrase, and the latter two digits, "3" and "8", make the second prosodic phrase. The transition of F_0 is represented by CV syllabic units, and each CV syllable can be prosodically labeled as "rising" or "falling".

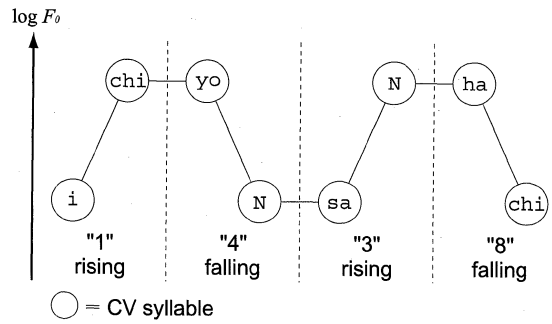


Fig. 1 An example of F_0 contour of four-connected-digit speech in Japanese.

2.2.2 Integration of Segmental and Prosodic Features

Each segmental feature vector has 25 elements consisting of 12MFCC (Mel Frequency Cepstral Coefficients), their delta and the delta log energy. The window length is 25 ms and the frame interval is 10 ms. Cepstral mean subtraction (CMS) is applied to each utterance.

Simultaneously, two kinds of prosodic features are extracted: $\log F_0$ and $\Delta \log F_0$. It has been confirmed that speaker verification performance is improved by using both prosodic features in comparison with the method using either of the features [10]. A segmental-prosodic feature vector is built by combining the segmental and prosodic feature vectors at each frame.

2.2.3 Integration of Segmental and Prosodic Models

Since F_0 transitions are easily represented by using CV syllabic units in Japanese connected digit speech, syllabic unit HMMs are used for modeling segmental and prosodic features in this study. This method can also be applied to other languages by changing the modeling units depending on the characteristics of the F_0 transitions of the target language.

An integrated syllable HMM denoted by "SP-HMM (Segmental-Prosodic HMM)" is modeled by taking both the syllable context and the F_0 transitions into account. Each Japanese digit uttered continuously with other digits can be modeled by a concatenation of two-syllable units (morae). Even "2" (/ni/) and "5" (/go/) can be modeled by two syllables, since their final vowel is usually lengthened as /ni:/ and /go:/. The syllable context is considered only within each digit in our experiment. Therefore, the SP-HMM can be denoted by either a left-context dependent syllable "`LC-SYL,PM`" or a right-context dependent syllable "`SYL+RC,PM`", where "`PM`" indicates an F_0 transition pattern which is either rising (`U`) or falling (`D`). For example, the first syllable /i/ of "1" (/ichi/) which has rising F_0 transition is denoted as "`i+chi,U`". Each SP-HMM has a standard left-to-right topology with $n \times 3$ states, where n is the number of

†The ESPS get_f0 function is one of the standard robust F_0 extraction methods. It uses the normalized cross-correlation function and dynamic programming [22].

Table 1 The list of integrated models (SP-HMMs).

digit	model	
0	ze+ro,U	ze+ro,D
/zero/	ze-ro,U	ze-ro,D
1	i+chi,U	i+chi,D
/ichi/	i-chi,U	i-chi,D
2	ni+i,U	ni+i,D
/ni:/	ni-i,U	ni-i,D
3	sa+N,U	sa+N,D
/saN/	sa-N,U	sa-N,D

digit	model	
4	yo+N,U	yo+N,D
/yoN/	yo-N,U	yo-N,D
5	go+o,U	go+o,D
/go:/	go-o,U	go-o,D
6	ro+ku,U	ro+ku,D
/roku/	ro-ku,U	ro-ku,D
7	na+na,U	na+na,D
/nana/	na-na,U	na-na,D

digit	model	
8	ha+chi,U	ha+chi,D
/hachi/	ha-chi,U	ha-chi,D
9	kyu+u,U	kyu+u,D
/kyu:/	kyu-u,U	kyu-u,D
	sil	sp

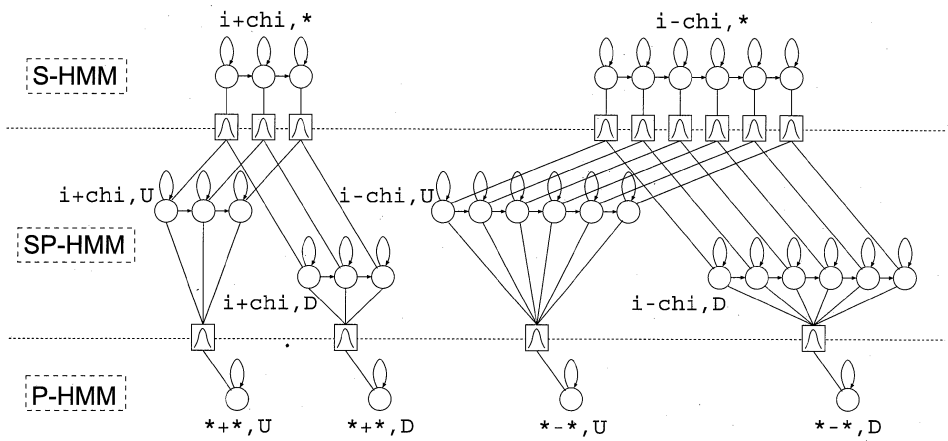


Fig. 2 Building SP-HMMs using a tied-mixture technique. S-HMMs and P-HMMs are trained using segmental and prosodic features, respectively.

phonemes in the syllable. Table 1 shows a list of integrated models. "sil" denotes a long pause that appears at the beginning and end of an utterance, and "sp" denotes a short pause that appears between digits. The "sil" model has 3 states, and the "sp" model has 1 state.

SP-HMMs are modeled as multi-stream HMMs. In recognition, the log-probability $b_j(\boldsymbol{O}^t_{sp})$ of generating the t-th frame segmental-prosodic observation $\boldsymbol{O}^t_{sp}$ at state j is calculated by:

$$b_j(\boldsymbol{O}^t_{sp}) = \lambda_s b_j(\boldsymbol{O}^t_s) + \lambda_p b_j(\boldsymbol{O}^t_p) \quad (1)$$

where $b_j(\boldsymbol{O}^t_s)$ is the log-probability of generating a segmental feature vector $\boldsymbol{O}^t_s$, and $b_j(\boldsymbol{O}^t_p)$ is the log-probability of generating a prosodic feature vector $\boldsymbol{O}^t_p$. λ_s and λ_p are weighting factors for the segmental and prosodic streams respectively.

Syllable HMMs for segmental and prosodic feature vectors are separately made and combined to build SP-HMMs using a tied-mixture technique as follows:

i. "S-HMMs (Segmental HMMs)" are trained by segmental features only. They can be denoted by either "LC-SYL,*" or "SYL+RC,*". Here, "*" (wild card) means that HMMs are built without considering the F_0 transitions, "U" and "D". The total number of S-HMM states is the same as the number of SP-HMM states.
ii. Training utterances are segmented into syllables by the forced-alignment technique using the S-HMMs, and one of the F_0 transition labels, "U" or "D", is given to each segment according to the actual F_0 pattern.
iii. "P-HMMs (Prosodic HMMs)" are trained by prosodic feature vectors within these segments, according to the F_0 transition label. Six separate models, "*-*,U", "*+*,U", "*-*,D", "*+*,D", "sil", and "sp" are made. Each P-HMM has a single state.
iv. The S-HMMs and P-HMMs are combined to make SP-HMMs. Gaussian mixtures in the segmental stream of SP-HMMs are tied with corresponding S-HMM mixtures, while the mixtures in the prosodic stream are tied with corresponding P-HMM mixtures. Figure 2 shows the integration process. In this example, the mixtures of SP-HMM "i+chi,U" are tied with S-HMM "i+chi,*" and P-HMM "*+*,U".

2.2.4 Verification Score

A verification score after observing a feature set $\boldsymbol{O}$ is denoted by $q(\boldsymbol{O})$, which is calculated as

$$q(\boldsymbol{O}) = l(\boldsymbol{O}|C) - l(\boldsymbol{O}|G) \quad (2)$$

where $l(\boldsymbol{O}|C)$ is a frame-averaged log-likelihood value with claimed speaker's SP-HMM C and $l(\boldsymbol{O}|G)$ is a frame-averaged log-likelihood value with general speaker's SP-HMM G. The likelihood values are calculated under the assumption that each speaker can utter arbitrary four connected digits. This method can be extended to a text-prompted speaker verification system, in which a sequence of four digits to be spoken is prompted by the system and verification decision is made by combining a recognized

word sequence and voice similarity to the claimed speaker.

The log-likelihood values for a segmental-prosodic feature vector $\boldsymbol{O}_{sp}$ are defined using Eq. (1) as follows:

$$l(\boldsymbol{O}_{sp}|C) = \lambda_s l(\boldsymbol{O}_s|C) + \lambda_p l(\boldsymbol{O}_p|C), \quad (3)$$

$$l(\boldsymbol{O}_{sp}|G) = \lambda_s l(\boldsymbol{O}_s|G) + \lambda_p l(\boldsymbol{O}_p|G). \quad (4)$$

Then, the verification score $q(\boldsymbol{O}_{sp})$ is calculated as

$$q(\boldsymbol{O}_{sp}) = \lambda_s q(\boldsymbol{O}_s) + \lambda_p q(\boldsymbol{O}_p). \quad (5)$$

If the score is larger than a threshold value θ, the speaker is accepted as the claimed speaker. Therefore, the discriminant function is $z = q(\boldsymbol{O}_{sp}) - \theta$. If z is positive, the speaker is accepted, and if it is less than or equal to 0, the speaker is rejected as being an imposter. In the experiments in this paper, the stream-weights are constrained by

$$\lambda_s + \lambda_p = 1. \quad (6)$$

3. Automatic Stream-Weight and Threshold Optimization Methods

In real-world applications, the stream-weights λ_s, λ_p and the decision threshold θ parameters need to be estimated before verification [15],[16]. In this section, our proposed method for automatically optimizing these parameters using the LDA and the Adaboost techniques [17] is explained.

3.1 Estimation by the LDA

As described in the previous section, speaker verification by SP-HMM uses the following discriminant function z:

$$z = q(\boldsymbol{O}_{sp}) - \theta \quad (7)$$

$$= \lambda_s q(\boldsymbol{O}_s) + \lambda_p q(\boldsymbol{O}_p) - \theta. \quad (8)$$

Since z is a linear function, the stream-weights and the threshold can be estimated as coefficients of a linear function obtained by the LDA. The estimation process is as follows. First, segmental and prosodic scores, $q(\boldsymbol{O}_s)$ and $q(\boldsymbol{O}_p)$, calculated from both claimed speaker's and imposter's data included in the development set are plotted in a two-dimensional space composed of $q(\boldsymbol{O}_s)$ and $q(\boldsymbol{O}_p)$. Then, the LDA is applied to the space so as to obtain the discriminant function z which distinguishes score distribution of claimed speakers from that of imposters.

Since the obtained function $z = a_s q(\boldsymbol{O}_s) + a_p q(\boldsymbol{O}_p) - b$ does not satisfy $a_s + a_p = 1$, it is transformed so that the sum of the coefficients becomes 1 according to the constraint Eq. (6). The estimated values of the stream-weights and the threshold are

$$\hat{\lambda}_s = \frac{a_s}{a_s + a_p}, \ \hat{\lambda}_p = \frac{a_p}{a_s + a_p}, \ \hat{\theta} = \frac{b}{a_s + a_p}. \quad (9)$$

Thus, all the parameters are estimated according to the LDA criterion which maximizes discrimination performance between claimed speakers and imposters.

In the boosting process described in the next subsection, multiple discriminant functions obtained from boosting iterations are integrated. The normalization by Eq. (9) is necessary for correctly weighting these discriminant functions in the integration process.

There is no guarantee that the denominator $a_s + a_p$ does not become 0, and if it becomes 0, some special processing such as flooring needs to be applied. However, almost all the actual a_s and a_p values estimated in Sect. 4.5 and 4.6 were positive, and only a few negative values were observed in several low SNR conditions. Hence, practically $a_s + a_p$ never became 0.

3.2 Optimization by the Adaboost

The Adaboost, a class of boosting algorithms, constructs a high performance classifier by sequentially combining trained simple classifiers [19]. In our optimization method, the linear discriminant functions obtained by the LDA are used as simple classifiers for the Adaboost. By doing so, we can estimate more accurate weights and thresholds than those obtained by only using the LDA. In this paper, the stream-weights and the threshold are optimized to achieve equal error rates (EERs) at which FAR is equal to FRR.

Details of the optimization algorithm are as follows, where n represents the number of data in the development set and K represents the number of iterations. Let $\{x_i\}(i = 1, \cdots, n)$ be the development data plotted into the two-dimensional space composed of $q(\boldsymbol{O}_s)$ and $q(\boldsymbol{O}_p)$, and $\{w_i^{(k)}\}$ $(i = 1, \cdots, n$, and $k = 1, \ldots, K)$ be the weights of each data in k-th iteration.

i. Initialize the weights of data $w_i^{(1)} = 1/n$.
ii. Iterate the following processes for $k = 1, \cdots, K$.

a. Choose n samples from $\{x_i\}$ allowing duplications, using $\{w_i^{(k)}\}$ as a probability distribution to make the re-sampled data $\{x_i'^{(k)}\}$.
b. Obtain a linear discriminant function

$$z_k = \lambda_s^{(k)} q(\boldsymbol{O}_s) + \lambda_p^{(k)} q(\boldsymbol{O}_p) - \theta^{(k)} + \delta_{k-1} \quad (10)$$

by applying the LDA to the re-sampled data $\{x_i'^{(k)}\}$, where δ_{k-1} is an offset value for adjusting FAR and FRR to EER. The offset value is obtained from the $\{k-1\}$-th boosting iteration, where the initial value δ_0 is set at 0.
c. Classify all the data in the development set $\{x_i\}$ using z_k, and calculate FAR, FRR, and the weighted discriminant errors ϵ_k, ϵ_{FA}, and ϵ_{FR}:

$$FAR_k = \frac{\sum_{i_{FA}} 1}{\sum_{i_{imp}} 1}, \quad (11)$$

$$FRR_k = \frac{\sum_{i_{FR}} 1}{\sum_{i_{cs}} 1}, \quad (12)$$

$$\epsilon_k = \sum_{i_{miss}} w_i^{(k)}, \quad (13)$$

$$\epsilon_{FA} = \frac{\sum_{i_{FA}} w_i^{(k)}}{\sum_{i_{imp}} w_i^{(k)}}, \tag{14}$$

$$\epsilon_{FR} = \frac{\sum_{i_{FR}} w_i^{(k)}}{\sum_{i_{cs}} w_i^{(k)}}, \tag{15}$$

where $\sum_{i_{FA}}$ and $\sum_{i_{FR}}$ are the summations for all i with the results of FA and FR, respectively, and $\sum_{i_{miss}}$ is the summation for all i with the classification error, FA or FR, in the classification of $\{x_i\}$. $\sum_{i_{imp}}$ and $\sum_{i_{cs}}$ are the summations for all i belonging to imposters and claimed speakers, respectively. The offset value δ_k is determined by the following equation:

$$\delta_k = \alpha \cdot \frac{1 - FAR_k/FRR_k}{1 + FAR_k/FRR_k}, \tag{16}$$

where α is a scaling factor of the offset value.

d. Calculate u_k as the weight of z_k by the following equations:

$$u_k = c(\epsilon_k) \cdot c(d_k), \tag{17}$$

$$d_k = \frac{|FAR_k - FRR_k|}{2}. \tag{18}$$

where the cost function $c(x)$ is defined by the following equation:

$$c(x) = \frac{1}{2} \log \frac{1-x}{x}. \tag{19}$$

e. Update $\{w_i^{(k)}\}$ by the following formula:

$$w_i^{(k+1)} = \begin{cases} w_i^{(k)} e^{-c(err_k)} (i\text{:classify } x_i \text{ accurately}) \\ w_i^{(k)} e^{c(err_k)} \quad (i\text{:misclassify } x_i) \end{cases} \tag{20}$$

where err_k is a weighted error rate in k-th iteration calculated as follows:

$$err_k = \frac{\epsilon_{FA} + \epsilon_{FR}}{2}. \tag{21}$$

f. Normalize $\{w_i^{(k+1)}\}$ to meet $\sum_{i=1}^{n} w_i^{(k+1)} = 1$.

iii. Let the conclusive classifier z be the weighted majority vote of z_k:

$$z = \sum_{k=1}^{K} (u_k z_k). \tag{22}$$

iv. Normalize the coefficients of z so that the sum of them becomes 1.

v. Set the normalized coefficients as the estimated stream-weights and the threshold.

In the original Adaboost algorithm, the conclusive classifier z is defined by $z = \sum_{k=1}^{K} \{u_k \times sign(z_k)\}$. However, this cannot be directly used for stream-weight estimation, since its form is not a linear discriminant function. Thus, we approximate z by $z = \sum_{k=1}^{K} (u_k z_k)$ as shown in Eq. (22).

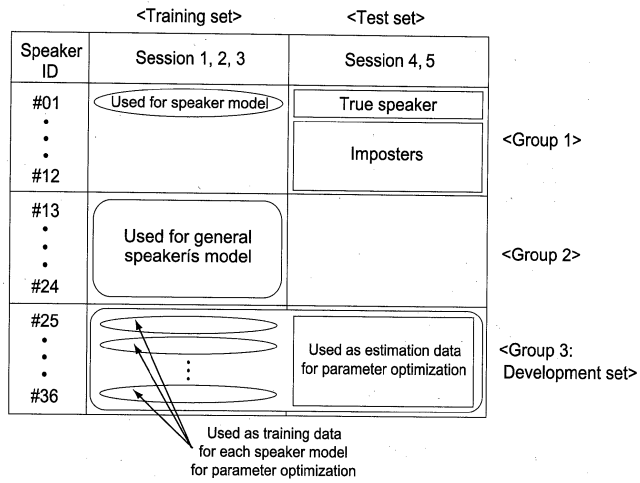


Fig. 3 Training, testing and development sets for the verification experiment when the speaker #01 is used as the claimed speaker.

4. Experiments

4.1 Experimental Conditions

Speech data were recorded in 5 sessions separated by intervals of approximately one month. The data were collected from 36 male speakers and sampled at 16 kHz with a 16 bit resolution. Each speaker uttered 50 four-connected-digit strings in Japanese at each session. Each digit appeared the same number of times in this set of 50 randomly arranged four digit strings.

For each speaker, 150 strings recorded at sessions 1 ~ 3 were used for training and 100 strings recorded at sessions 4 and 5 were used for testing.

The database was separated into three groups in terms of speakers as shown in Fig. 3. The figure shows the case where speaker #01 was used as the claimed speaker. The general speaker's model was trained using utterances by all the speakers in the speaker group 2, which did not include the claimed speaker, and the stream-weights and the threshold were optimized using the data of speaker group 3. Each speaker model of speaker group 3 was trained using utterances from sessions 1 ~ 3 for the parameter optimization. Utterances from sessions 4 and 5 of group 3 were used as estimation data for the parameter optimization. Another experiment was conducted in which the general speaker's model was trained by the data of speaker group 3 and the weights and the threshold were optimized by the data of speaker group 2. Next, group 1 & 3 are used for training and development for testing group 2, and finally group 1 & 2 are used for training and development for testing group 3. There are six combinations of the training set, the development set and the testing set. The result averaged over the six experiments was used for evaluation.

White noise was added to the training set at a 30 dB SNR level to increase robustness against noisy speech, and the development and testing sets were contaminated with

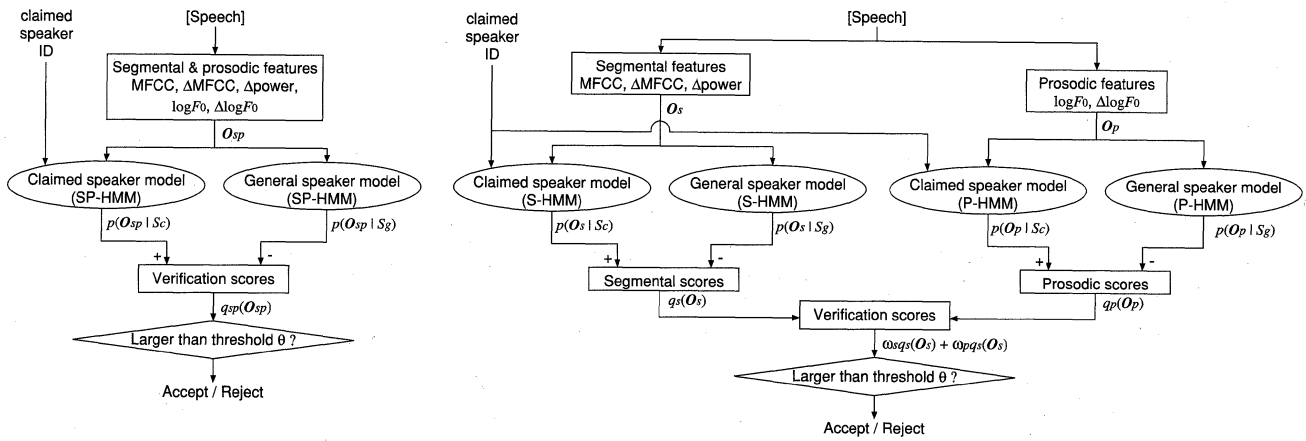


Fig. 4 Flow of speaker verification process. (a): The verification score $q_{sp}(\boldsymbol{O}_{sp})$ is calculated from a segmental-prosodic feature vector $\boldsymbol{O}_{sp}$. (b): The scores $q_s(\boldsymbol{O}_s)$ and $q_o(\boldsymbol{O}_p)$ are separately calculated from segmental and prosodic feature vectors, $\boldsymbol{O}_s$ and $\boldsymbol{O}_p$, and integrated afterward.

white, in-car or elevator-hall noise at 5, 10, 15, 20 and 30 dB SNR conditions. The latter two noises are included in the noise database distributed by the Japan Electronic Industry Development Association (JEIDA) [23]. The development set contaminated with the same noise at the same SNR as the testing set was used for automatically optimizing the stream-weights and the decision threshold.

In our preliminary experiments using S-HMMs with a white noise SNR of 30 dB, the most accurate verification performance was achieved using S-HMMs with four mixtures. Thus four mixture S-HMMs were also used in the following experiments. Each mixture component for all the HMMs (S-HMMs, P-HMMs, and SP-HMMs) was modeled using a diagonal-covariance Gaussian distribution.

4.2 Effectiveness of Multi-Stream HMMs

We first investigated the effectiveness of using multi-stream HMMs to integrate segmental and prosodic information. A two-dimensional prosodic feature vector, consisting of $\log F_0$ and $\Delta \log F_0$ extracted by the Hough transform, was used for this experiment.

The EERs of the following two cases were compared:

(a) The case where the $q_{sp}(\boldsymbol{O}_{sp})$ obtained from SP-HMMs is used as the verification score.
(b) The case where $\omega_s q_s(\boldsymbol{O}_s) + \omega_p q_p(\boldsymbol{O}_p)$ which is the weighted sum of $q_s(\boldsymbol{O}_s)$ and $q_p(\boldsymbol{O}_p)$ obtained separately from S-HMMs and P-HMMs is used as the verification score.

The subscript m of q_m represents the model from which the score is calculated, S-HMMs, P-HMMs or SP-HMMs. In this experiment, m was either s, p or sp. (a) and (b) in Fig. 4 show the flow charts of the speaker verification process in each case. The integration weights, ω_s and ω_p, and the stream-weights, λ_s and λ_p, were manually optimized in the 0 to 1 range using the testing set ex post facto for each noise condition. The number of mixtures in prosodic models was optimized for each case at the white noise 30 dB SNR condition; the best number of mixtures was four in both cases.

Table 2 Comparison of the EERs (%) of the case where the S-HMMs and P-HMMs are separately used with the case where the SP-HMMs are used.

Noise	SNR	S-HMM & P-HMM	SP-HMM
white noise	30 dB	0.66	0.63
	20 dB	3.42	3.28
	15 dB	9.40	8.74**
	10 dB	17.6	16.1**
	5 dB	23.8	23.5
in-car noise	30 dB	11.3	10.5**
	20 dB	11.1	10.9
	15 dB	11.2	11.2
	10 dB	11.7	11.9
	5 dB	13.3	13.6
elevator-hall noise	30 dB	3.71	3.61
	20 dB	5.07	5.10
	15 dB	9.86	9.60
	10 dB	19.7	18.7**
	5 dB	28.9	28.5

The results in the white noise, in-car noise and elevator-hall noise conditions are shown in Table 2. The symbol "**" indicates that the EER obtained by the feature level integration implemented by SP-HMMs is significantly better at 1% significance level than that obtained when the scores obtained separately from S-HMMs and P-HMMs are integrated afterward.

These results show that the speaker verification method using multi-stream HMMs for integrating segmental and prosodic information is effective in most of the noise conditions, except for a small number of conditions where a minor degradation is observed. Our previous research on speech recognition [12] showed that using multi-stream HMMs which integrated segmental and prosodic features yielded better time alignment of digits in noisy conditions. Probably this is also the reason why EERs are decreased by using SP-HMMs in speaker verification.

Table 3 Comparison of the EERs when using the prosodic feature vector with/without the Hough transform.

Noise	SNR	S-HMM	SP-HMM (**NH-LD**)	SP-HMM (**H-LD**)
white noise	30 dB	0.88	0.63	0.63
	20 dB	4.91	3.77	3.28**
	15 dB	14.7	10.9	8.74**
	10 dB	27.1	20.9	16.1**
	5 dB	37.5	31.1	23.5**
in-car noise	30 dB	15.9	12.4	10.5**
	20 dB	15.9	12.3	10.9**
	15 dB	15.7	12.3	11.2**
	10 dB	16.0	13.0	11.9**
	5 dB	18.4	15.1	13.6**
elevator-hall noise	30 dB	4.48	3.69	3.61
	20 dB	5.79	5.02	5.10
	15 dB	11.7	9.66	9.60
	10 dB	24.8	19.9	18.7**
	5 dB	40.2	32.3	28.5**

4.3 Effectiveness of the Hough Transform

For examining the effect of the Hough transform on verification performance, a two-dimensional prosodic feature vector was extracted for each frame without using the Hough transform; it consisted of $\log F_0$, extracted by choosing highest cepstral peaks, and $\Delta \log F_0$, computed by linear smoothing of the $\log F_0$ values within a 90 ms window.

The comparisons of the EERs when using the feature vector with using the Hough transform (**H-LD**) and without using it (**NH-LD**) are shown in Table 3. SP-HMMs were used for this experiment. Stream-weights were manually optimized in the 0 to 1 range using the testing set ex post facto for each noise condition. In the tables, the EERs with "**" indicate that results with the Hough transform are significantly better than that without the transform at 1% significance level.

These results indicate that the Hough transform is effective in F_0 extraction for noise-robust speaker verification.

4.4 Effects of the Prosodic Stream Weight

Figure 5 shows the EERs as a function of the prosodic stream weight λ_p at each SNR in the white noise environment. Improvements from baseline ($\lambda_p = 0$) are observed over a wide range: $0.0 < \lambda_p < 0.9$ in all SNR conditions. Since the performance is not sensitive to the change of the stream weight in the range of $0.0 < \lambda_p < 0.9$ in all SNR conditions, it is confirmed that the proposed speaker verification method is reliable against the change of the stream weight.

4.5 Effectiveness of Adaboost-Based Stream-Weights and the Threshold Optimization Method

The stream-weight and threshold optimization method was evaluated under various noise conditions. In this experiment, the number of iterations and the scaling factor α to determine the offset value δ_k were experimentally set to 200 and 0.005, respectively.

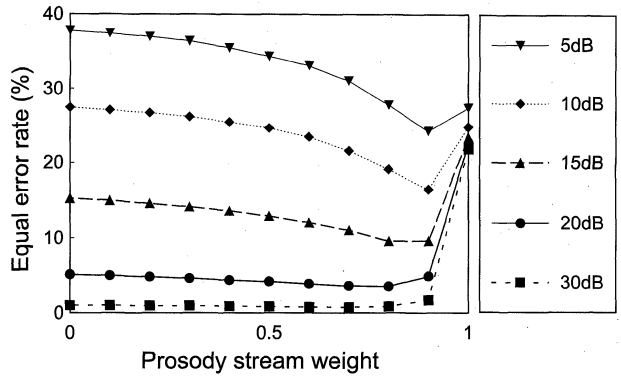


Fig. 5 The EERs as a function of the prosodic stream weight (λ_p) at each SNR in the white noise environment.

Table 4 shows decision costs $C = 0.5 \cdot FAR + 0.5 \cdot FRR$ obtained by using the proposed optimization method in each noise condition. Since the target of the FAR and the FRR is an EER in this experiment, the weights of both the FAR and the FRR are 0.5. The results in "LDA only" indicate that using the stream-weights and the threshold estimated by only the LDA; and, the results in "Adaboost" were obtained by the proposed optimization method combining the boosting technique. The bottom line shows the target decision costs which are obtained when the stream-weights and the threshold were manually optimized for the testing set ex post facto so that the EERs were minimized.

The results show that the Adaboost-based optimization method outperforms the method using only the LDA in most of the SNR conditions with white and elevator-hall noise. In an in-car noise environment, since the method using only the LDA is effective enough in adjusting the FARs and FRRs to the target EERs, the Adaboost-based method does not further improve them.

4.6 Effects of the Number of the Adaboost Iterations

Figures 6 (a), (b), and (c) show FARs and FRRs for the development set as a function of the number of Adaboost iterations[†] at 15 dB SNR condition in white, in-car, and elevator-hall noise environment, respectively.

FARs and FRRs smoothly converges to the EERs. These results indicate that the proposed method is stable over a variable number of Adaboost iterations.

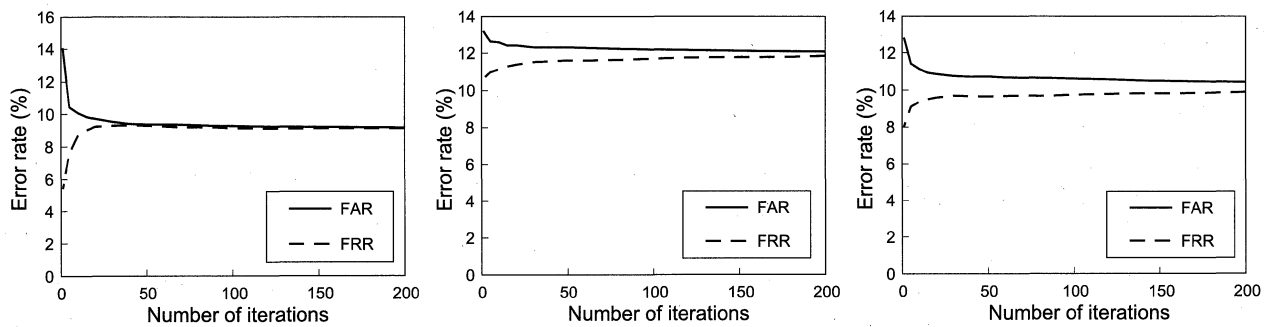
5. Conclusions

This paper has evaluated a speaker verification method using multi-stream HMMs which combine segmental and prosodic features in various noise conditions. This method uses an automatic stream-weight and threshold optimization method based on the LDA and the Adaboost approaches.

[†]They have been denoted by "FAR_k" and "FRR_k" in Eqs. (11) and (12).

Table 4 Comparison of the decision costs with different stream-weights and threshold optimization methods in various SNR conditions.

Optimization method	white noise					in-car noise					elevator-hall noise				
	30 dB	20 dB	15 dB	10 dB	5 dB	30 dB	20 dB	15 dB	10 dB	5 dB	30 dB	20 dB	15 dB	10 dB	5 dB
LDA only	1.74	4.50	10.0	17.1	24.7	11.5	11.9	12.3	13.0	14.7	4.30	5.58	10.7	19.6	30.3
Adaboost	1.10	3.60	9.05	16.3	24.7	11.7	11.8	12.1	12.9	15.1	4.08	5.25	10.0	19.2	30.3
Manual optimization (Target)	0.63	3.28	8.74	16.1	23.5	10.5	10.9	11.2	11.9	13.6	3.61	5.10	9.60	18.7	28.5



(a) White noise 15 dB. (b) In-car noise 15 dB. (c) Elevator-hall noise.

Fig. 6 Transition of FAR and FRR when the number of Adaboost iterations increases at 15 dB SNR condition. ((a) white noise, (b) in-car noise, and (c) elevator-hall noise)

The prosodic features are extracted by an F_0 feature extraction technique based on the Hough transform. Experimental results using Japanese connected digit utterances show that: 1) the Hough transform is effective for increasing robustness in extracting the F_0 features at various noise conditions; 2) the multi-stream verification method is robust against the change of the stream weight; 3) the Adaboost-based parameter optimization method is effective in white and elevator-hall noise conditions; and 4) the optimization method is stable over the variation of the number of Adaboost iterations.

Our future work include: 1) investigating prosodic features other than F_0-based features, such as durations; 2) improving the SP-HMM topology; 3) using voiced/unvoiced information; 4) improving the stream-weight and threshold optimization algorithm so that FAR-FRR ratio can be freely adjusted; 5) evaluating performance when applying the proposed optimization method to multi-stream speaker verification systems using a larger number of streams; 6) investigating a parameter optimization method using a testing set without having labelled speaker IDs, instead of using a development set; 7) generalizing the proposed method so that it can be applied to speaker verification tasks using arbitrary words or sentences; and 8) evaluating performance of the proposed method on real-world data recorded under actual noisy environment.

Acknowledgment

This research is supported by the Ministry of Education, Culture, Sports, Science, and Technology (MEXT), Grant-in-Aid for Young Scientists (B), 17700141, 2005.

References

[1] T. Matsui and S. Furui, "Text-independent speaker recognition using vocal tract and pitch information," Proc. ICSLP1990, vol.1, pp.137–140, Kobe, Japan, Nov. 1990.
[2] M.J. Carey, E.S. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," Proc. ICSLP1996, vol.3, pp.1800–1803, Philadelphia, Pennsylvania, USA, Oct. 1996.
[3] M.K. Sonmez, L. Heck, M. Weintraub, and E. Shriberg, "A lognormal tied mixture model of pitch for prosody-based speaker recognition," Proc. Eurospeech1997, vol.3, pp.1391–1394, Rhodes, Greece, Sept. 1997.
[4] Y.-J. Kyung and H.-S. Lee, "Text independent speaker recognition using micro-prosody," Proc. ICSLP1998, vol.1, pp.157–160, Sydney, Australia, Dec. 1998.
[5] Y. Cheng and H.-C. Leung, "Speaker verification using fundamental frequency," Proc. ICSLP1998, vol.1, pp.161–164, Sydney, Australia, Dec. 1998.
[6] K.P. Markov and S. Nakagawa, "Text-independent speaker recognition using multiple information sources," Proc. ICSLP1998, vol.1, pp.173–176, Sydney, Australia, Dec. 1998.
[7] C. Miyajima, Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Text-independent speaker identification using Gaussian mixture models based on multi-space probability distribution," IEICE Trans. Inf. & Syst., vol.E84-D, no.7, pp.847–855, July 2001.
[8] F. Weber, L. Manganaro, B. Peskin, and E. Shriberg, "Using prosodic and lexical information for speaker identification," Proc. ICASSP2002, vol.1, pp.141–144, Orlando, Florida, USA, May 2002.
[9] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition," Proc. ICASSP2003, vol.4, pp.784–787, Hong Kong, China, April 2003.
[10] K. Iwano, T. Asami, and S. Furui, "Noise-robust speaker verification using F_0 features," Proc. ICSLP2004, vol.2, pp.1417–1420, Jeju Island, Korea, Oct. 2004.

[11] T. Wark, S. Sridharan, and V. Chandran, "The use of temporal speech and lip information for multi-modal speaker identification via multi-stream HMM's," Proc. ICASSP2000, vol.4, pp.2389–2392, Istanbul, Turkey, June 2000.

[12] K. Iwano, T. Seki, and S. Furui, "Noise robust speech recognition using F_0 contour extracted by Hough transform," Proc. ICSLP2002, vol.2, pp.941–944, Denver, Colorado, USA, Sept. 2002.

[13] K. Iwano, T. Seki, and S. Furui, "Noise robust speech recognition using F_0 contour information," IEICE Trans. Inf. & Syst., vol.E87-D, no.5, pp.1102–1109, May 2004.

[14] P.V.C. Hough, "Method and means for recognizing complex patterns," U.S. Patent #3069654, 1962.

[15] T. Matsui, T. Nishitani, and S. Furui, "Robust methods of updating model and a priori threshold in speaker verification," Proc. ICASSP1996, vol.1, pp.97–100, Atlanta, GA, USA, May 1996.

[16] J. Lindberg, J. Koolwaaij, H.-P. Hutter, D. Genoud, J.-B. Pierrot, M. Blomberg, and F. Bimbot, "Techniques for a priori decision threshold estimation in speaker verification," Proc. RLA2C, pp.89–92, Avignon, France, April 1998.

[17] T. Asami, K. Iwano, and S. Furui, "A stream-weight and threshold estimation method using adaboost for multi-stream speaker verification," Proc. ICASSP2006, SS.7-4, Toulouse, France, May 2006.

[18] R.A. Fisher, "The use of multiple measurements in taxonomic problems," Annals of Eugenics, vol.7, pp.179–188, 1936.

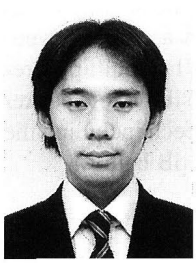
[19] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol.55, no.1, pp.119–139, Aug. 1997.

[20] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognit., vol.13, no.2, pp.111–122, 1981.

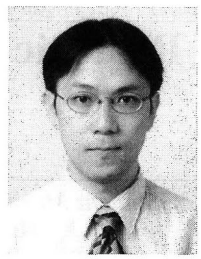
[21] T. Seki, K. Iwano, and S. Furui, "Robust pitch extraction for noisy environments using Hough transformation," IPSJ SIG Technical Reports, vol.2001, no.100, pp.9–14, Oct. 2001.

[22] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in Speech Coding and Synthesis, ed. W.B. Kleijn and K.K. Paliwal, pp.495–518, Elsevier, Amsterdam, NL, 1995.

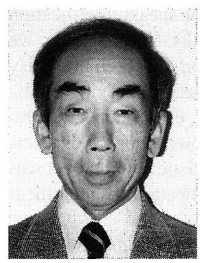
[23] S. Itahashi, "Recent speech database projects in Japan," Proc. ICSLP1990, vol.2, pp.1081–1084, Kobe, Japan, Nov. 1990.



Taichi Asami received the B.E. and M.E. degrees in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2004 and 2006, respectively. He has been with Cyber Space Laboratories, Nippon Telegraph and Telephone Corporation, Kanagawa, Japan, since 2006. He is a member of the Acoustical Society of Japan (ASJ).



Koji Iwano received the B.E. degree in information and communication engineering in 1995, and the M.E. and Ph.D. degrees in information engineering respectively in 1997 and 2000 from the University of Tokyo. He is currently an Assistant Professor at Tokyo Institute of Technology, Department of Computer Science. His research interests are in speech information processing, such as speech recognition, speaker verification, and speech synthesis. He is a member of the IEEE, International Speech Communication Association (ISCA), the Information Processing Society of Japan (IPSJ), and the Acoustical Society of Japan (ASJ).



Sadaoki Furui is currently a Professor at Tokyo Institute of Technology, Department of Computer Science. He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human-computer interaction and has authored or coauthored over 700 published articles. He is a Fellow of the IEEE and the Acoustical Society of America. He has served as President of the Acoustical Society of Japan (ASJ) and the International Speech Communication Association (ISCA). He has served as a member of the Board of Governor of the IEEE Signal Processing (SP) Society and Editor-in-Chief of both the Transaction of the IEICE and the Journal of Speech Communication. He has received the Yonezawa Prize, the Paper Award and the Achievement Award from the IEICE (1975, 1988, 1993, 2003, 2003), and the Sato Paper Award from the ASJ (1985, 1987). He has received the Senior Award and Society Award from the IEEE SP Society (1989, 2006), the Achievement Award from the Minister of Science and Technology and the Minister of Education, Japan (1989, 2006), and the Purple Ribbon Medal from Japanese Emperor (2006). In 1993 he served as an IEEE SPS Distinguished Lecturer.