IEICE TRANS. INF. & SYST., VOL.E92–D, NO.4 APRIL 2009

PAPER

Multi-Input Feature Combination in the Cepstral Domain for Practical Speech Recognition Systems

Yasunari OBUCHI[†a)] ***and*** **Nobuo HATAOKA**[††]**,** ***Members***

SUMMARY In this paper we describe a new framework of feature combination in the cepstral domain for multi-input robust speech recognition. The general framework of working in the cepstral domain has various advantages over working in the time or hypothesis domain. It is stable, easy to maintain, and less expensive because it does not require precise calibration. It is also easy to configure in a complex speech recognition system. However, it is not straightforward to improve the recognition performance by increasing the number of inputs, and we introduce the concept of variance re-scaling to compensate the negative effect of averaging several input features. Finally, we propose to take another advantage of working in the cepstral domain. The speech can be modeled using hidden Markov models, and the model can be used as prior knowledge. This approach is formulated as a new algorithm, referred to as Hypothesis-Based Feature Combination. The effectiveness of various algorithms are evaluated using two sets of speech databases. We also refer to automatic optimization of some parameters in the proposed algorithms.

key words: *robust speech recognition, microphone array, MFCC, feature combination, HMM*

1. Introduction

After a few decades of the run-up period dominated by researchers and engineers, speech recognition is now entering a second phase of commercialization. In its first phase, the performance had been presented and evaluated in terms of scientific research using "champion data." A test subject sat in a quiet room, wore a headset, had plenty of time to read the instructions and to relax for the task, pushed the trigger button, and finally uttered a given command or sentence. However, the second phase of commercialization requires something more complicated and difficult. The system must work properly even if the circumstance is far from such an ideal state. In fact, the second phase of speech recognition had already started, and the robustness of the speech recognition system in noisy environments is one of the hottest topics of this field. It is indeed true that noise is an important factor to speech recognition systems, but there are a number of factors that degrade speech recognition performance.

The ability to handle such degrading factors is generally called *ilities* [1], [2], because most of the words for such ability end with *-ility*. In this paper, we propose a feature combination framework for speech recognition, that has a lot of *ilities*. The basic idea of the framework is to combine the feature vectors obtained by a microphone array in the cepstral domain. This framework's *ilities* include

Stability: (a. k. a. robustness) The system can work effectively even in an unknown environment without elaborate adjustment. The large portion of noise effects of multiple channels are averaged out by feature combination.

Maintainability: The system's performance in the real filed is as good as in the laboratory. Even a slight miscalibration sometimes degrades the performance of phase-sensitive microphone array systems [3], but feature combination is not likely to be affected by it. Therefore, the system performance does not degrade even if we use it continually without daily maintenance.

Affordability: Since the feature vectors are calculated every 10 ms or so, the required precision of synchronization for a feature-based system is 1–2 ms. Therefore, the system can be constructed as an assembly of monaural or stereo audio systems, and it makes the system less expensive. This is another advantage over ordinary microphone array systems, in which all the input channels must be synchronized precisely (approx. 0.01–0.02 ms precision).

Configurability: The framework is independent from either preceding feature extraction or subsequent decoding (including acoustic and language models). Therefore, it is easy to develop the optimal system by assembling our feature combination module with existing feature extraction tools, decoder, acoustic and language models, etc.

Reliability: The feature combination framework is not based on any vulnerable assumption, such as the stability of the noise power spectrum, statistical independence of the input signals, etc. Therefore, it never performs unacceptably in an unpredicted situation.

In this paper, we start with a simple framework of MFCC averaging over a few microphone inputs, and expand it to various combination schemes. Although it works well with a few inputs, simple MFCC averaging becomes less effective by itself as we increase the number of inputs. This problem can be solved by introducing the idea of variance re-scaling, and we can put ***scalability*** on the pile of our *ilities* by solving this problem.

Furthermore, working in the cepstral domain has an additional advantage that speech can be modeled in this domain precisely using hidden Markov model (HMM), which can be used as the prior knowledge. This idea leads us to a modified MFCC averaging framework, referred to as Hypothesis-Based Feature Combination (HBFC), in which

Manuscript received June 11, 2008.
Manuscript revised October 28, 2008.

[†]The author is with Central Research Laboratory, Hitachi Ltd., Kokubunji-shi, 185–8601 Japan.
[††]The author is with the Department of Electronics and Intelligent Systems, Graduate School of Electronics, Tohoku Institute of Technology, Sendai-shi, 982–8577 Japan.
a) E-mail: yasunari.obuchi.jx@hitachi.com

DOI: 10.1587/transinf.E92.D.662

the speech model is processed in the cepstral domain to be combined with the input features.

The rest of this paper is organized as follows. In the next section, we look over preceding research works dealing with multiple microphone inputs. Next, we describe two databases which are used to evaluate the proposed framework throughout this paper. In the forth section, we propose to use the average of multiple MFCC feature vectors. Evaluation experiments reveals that there are some problems with simple MFCC average, and we bring up some modifications. In the fifth section, feature combination framework will be extended to two-path decoding scheme, in which we can take advantage of speech models as the prior knowledge. The effectiveness of HBFC will be proved through various evaluation experiments. Finally, conclusions and future works are presented in the last section.

2. A Quick Survey of Multi-Input Speech Recognition

The target of this paper is a speech recognition system which has more than one microphone for speech input. In this section, we present a brief overview of multi-input speech recognition.

2.1 Beamforming

When two closely-located microphones receive the sound signal coming from one direction, it is expected that their trajectories on the time axis have similar patterns and one has a slight latency against the other. Therefore, we can generate an output signal which has specific sensitivity to directions by controlling the gain and latency to each input and summing them up. Such a technique can be easily extended to more than two microphones, and the general framework is called beamforming. The simplest and most popular one is a delay-and-sum beamformer [4], which uses a fixed set of gain and latency parameters to enhance signals from one direction (typically front) and suppress the rest. More sophisticated one is an adaptive beamformer [5], which controls the gain and latency parameters so as to minimize a pre-defined cost function. It is also possible to combine beamforming with spectral subtraction [6].

Since beamforming is based on the small difference of arriving time, such as 0.1 ms with 3 cm replacement, it is important to maintain the synchronization of the speech inputs. It is also important to know what kind of sensitivity pattern must be designed to obtain good results, which is often not obvious in real environments.

2.2 Blind Source Separation

As mentioned above, setting the required sensitivity pattern is not always a trivial question. Blind source separation is a framework to obtain the desired signal by introducing statistical assumption of the signal instead of the geometrical knowledge. If we assume that the number of signal sources is the same as the number of microphones, and that all the signal sources are statistically independent from each other, then we can use Independent Component Analysis (ICA) [7] to separate them. It was shown that ICA provides excellent performance in a controlled situation, such as separating two voices from different directions in a nonechoic room. However, it is not as effective in a diffusive noise environment. ICA is also vulnerable if the room is echoic because the echo is not statistically independent from the original sound.

2.3 Parallel Decoding

In beamforming and blind source separation, multiple inputs are combined in a very early stage of speech recognition. In contrast, it is also possible to combine them in a very late stage of speech recognition. ROVER [8] is a well-known example of this category, in which multiple recognition hypotheses are aligned with each other to create a word transition network (WTN), and the best path is determined by voting. In fact, parallel decoding is useful even for isolated word recognition if we have a reliable selection criterion [10], [11].

In most cases, parallel decoding is used to make use of multiple decoders [12] and/or multiple models [13]. However, it is also applicable to multiple inputs. Since it expects the inputs to be nonuniform (otherwise its output is almost identical to the single input case), this framework is more effective in uncontrolled environments with various disturbing factors.

The problem of parallel decoding is that it is time-consuming. Naturally an N-input parallel decoding system consumes N times as much time as a single input system.

Another approach of parallel decoding is to incorporate the combination process into the decoding process. Yamada et al. proposed an algorithm based on a 3-D Viterbi search [9], which was proved to be effective to treat moving sound sources.

2.4 Feature Combination

There have been some previous works of multi-microphone speech recognition working on the feature domain. In [14], feature averaging in the cepstral domain was proposed. It is indeed the starting point of our work in this paper. Feature averaging approach of [14] was then expanded to feature regression approach of [15]. However, it was assumed in both works that the training data is available under the same condition of the recognition. In this paper, we start our work by applying these approaches to more general cases without matched training data, and solve various problems in such situations.

3. Databases

Throughout this paper, we use two databases to evaluate various implementations of feature combination and other competing algorithms. The CMU PDA speech database [10],

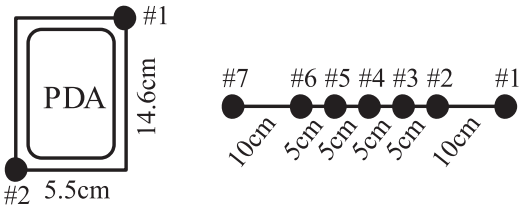


Fig. 1 Microphone alignment: CMU PDA speech database (left) and HITNAVI05 (right).

[11], [16] is an open database consisting of newspaper article sentences uttered in a laboratory. The other is our proprietary database [11] consisting of POI (point of interest) utterances in a car, referred to as HITNAVI05 in this paper.

3.1 CMU PDA Speech Database

This database was created in Carnegie Mellon University to develop speech-based interface of PDAs. It consists of two subsets, recorded by either single or multiple microphones respectively. In this paper, we use the multiple microphone subset (PDAm), which consists of 660 English utterances, uttered by 16 speakers in a noisy laboratory. The majority of the noise was made by computers and other electric tools (fan noise). Each utterance is a newspaper article whose average length is 16.2 words and perplexity is 64.35 (chosen from the test set of LDA's Wall Street Journal speech database) assuming a 5,000 word lexicon and a trigram language model. The original database includes five channels including a close-talk microphone, but we use only two of them. The alignment of the microphones is shown in Fig. 1 left.

3.2 HITNAVI05

This database was created in Hitachi Central Research Laboratory to develop speech-based interface of car navigation systems. It consists of 3,630 utterances, uttered by 18 speakers on the passenger seat of a moving car. Each speaker uttered one of 152 points of interest (POIs) in Japanese while the car is moving (or stopping due to heavy traffic) in the downtown area of Tokyo. Accordingly, the recognition task is isolated word recognition with a 152 word lexicon. The average SNR was estimated as −3.4 dB using NIST STNR tool. However, most of the noises are localized in the lower frequency range. After applying a high-pass filter with 400 Hz cut-off, it was improved to 10.0 dB. The remaining noises were attributed to the turn signal lever, road construction, etc., but the engine and road surface noises also have certain energy in middle to high frequency range. The utterances were recorded by seven microphones attached to the dashboard at the interval of 10 cm, 5 cm, 5 cm, 5 cm, 5 cm, and 10 cm (Fig. 1 right). These interval values were used to realize microphone pairs with various width. The shortest width of 5 cm corresponds to the 6.8 kHz sound wave, which is the upper limit of the important band for speech recognition. The longest width of 40 cm was determined by the width of the dashboard. Such an inhomogeneous arrangement can also be found in [17]. These microphones were labeled as #1 to #7 from the driver's side to the window's side, so #4 is the central microphone. More details of this database would be found in [18].

3.3 Acoustic and Language Models

Since we use two evaluation databases, one is English and the other is Japanese, we prepared two sets of acoustic models. In addition, since only the English database is made of continuous sentences, we prepared a trigram-based English language model. The acoustic and language models of English were trained using the training set of LDC's Wall Street Journal speech database. The acoustic models of Japanese were trained using our proprietary database, which is made of 16 hour clean speech of phonetically balanced sentences, uttered by 120 male and female speakers. All the training data were recorded in a quiet room using a close-talk microphone.

In each language, two kinds of acoustic model were trained. One is with cepstral mean normalization (CMN), and the other is with cepstral mean and variance normalization (MVN). Cepstral means and variances were calculated using the whole utterance. All the feature vectors in this paper are made of 13 dimensional MFCCs including 0-th coefficients. Feature combination is applied to these 13 dimensional MFCCs, but time-derivative operations to create delta and delta-delta parameters are applied to the combined feature. Hence the final feature vector to be decoded is made of 39 dimension elements. The sampling rate and frame rate are fixed at 16 kHz and 100 Hz respectively. In both languages, three state left-to-right triphone HMMs were used. English model is made of 2000 tied states, each of which has eight Gaussians. Japanese model is made of 1,614 states (without state-tying), each of which has six Gaussians.

4. Feature Combination by MFCC Averaging

4.1 MFCC Averaging and Accuracy Degradation

We started our feature combination evaluations with simple MFCC average over two inputs, using the CMU PDA speech database. Figure 2 shows the results. In these experiments, the combined feature vector was created by

$$\mathbf{x}_{ave} = (1 - W)\mathbf{x}_1 + W\mathbf{x}_2 \tag{1}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the simple input feature vectors and W is the weight parameter. Since we already proved that Delta-Cepstrum Normalization (DCN) [19] is effective for this database [11], $\mathbf{x}_1$ and $\mathbf{x}_2$ were normalized by DCN. The horizontal axis of Fig. 2 represents W, and the vertical axis represents the recognition accuracy defined as

$$\text{Accuracy} = 1 - \frac{S + I + D}{N} \tag{2}$$

where S, I, D are the numbers of substitution, insertion, and

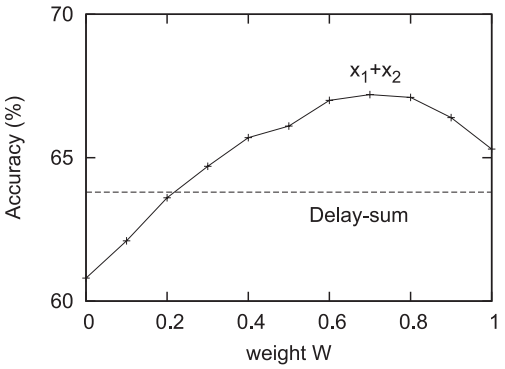


Fig. 2 MFCC averaging applied to CMU PDA speech database. All feature vectors were normalized by DCN before combined.

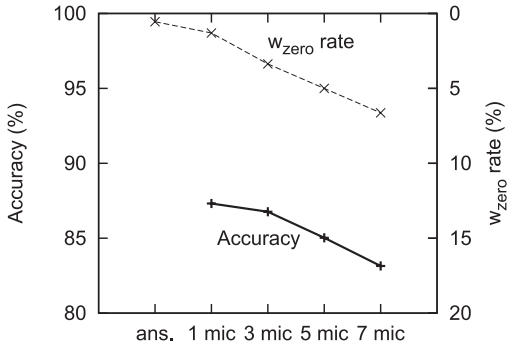


Fig. 3 MFCC averaging applied to HITNAVI05. All feature vectors were normalized by CMN before combined.

deletion errors respectively, and N is the number of words in the transcript.

Figure 2 revealed that the two input signals are quite different from each other, resulting in accuracies of 60.8% (Channel 1 only) and 65.3% (Channel 2 only). The difference could be attributed to various factors, including the distance and angle between the speaker and PDA, interference by the speaker's finger, and piece-to-piece characteristic variation of the microphones. As presented in [11], the CMU PDA speech database was recorded under two different conditions, and the average SNRs were estimated as 24.3 dB/15.2 dB (Channel 1) and 26.5 dB/18.6 dB (Channel 2). Therefore, the recognition accuracies were relatively low, in contrast with the higher recognition rate of 82.0% obtained by the close-talk microphone data.

A reference experiment using the delay-and-sum beamformer was also conducted, in which the original 16 kHz waveforms were upsampled to 64 kHz, the optimal delay was obtained by calculating correlation values, and the two inputs were added. Due to the large difference between two input signals, the accuracy of the delay-and-sum beamformer (dashed horizontal line in the figure) is lower than that of the better channel. In contrast, the average of two feature vectors ($\mathbf{x}_1 + \mathbf{x}_2$ in the figure) makes an upward convex curve, and we can obtain better results than choosing the better channel by setting the value of the weight parameter appropriately. It is also found that the MFCC averaging provides better results than the delay-and-sum beamformer in the wide range of W.

The remaining problem is how to find an appropriate value of W. We had proposed a solution for this problem [20], in which decoding results obtained with various Ws are compared in view of feature compensation efficiency. It is also possible to compare the likelihoods generated by the decoder. Our preliminary experiments showed that the accuracy was 65.2% (feature compensation efficiently) or 66.8% (likelihood), but these methods are very time-consuming due to their multiple decoding requirements.

Next, we applied MFCC averaging to HITNAVI05, in which the accuracy of isolated word recognition is simply defined as $1-S/N$. Since our preliminary experiments showed that the seven inputs had similar accuracies, we adopted a very simple averaging over N channels,

$$\mathbf{x}_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i. \tag{3}$$

where either CMN or MVN was applied to create $\mathbf{x}_i$ because DCN is not as effective for the short POI utterances of HITNAVI05 as for the long newspaper articles of CMU PDA speech database.

In Fig. 3, the results of MFCC averaging over 1 (microphone #4 only), 3 (#3, 4, 5), 5 (#2, 3, 4, 5, 6), and 7 (#1, 2, 3, 4, 5, 6, 7) microphones (with CMN) are plotted by the solid line. It was disappointing that the accuracy was not improved, and became even worse as we increased the number of microphones. However, we found from the error analysis that the recognition errors were mostly converged on a specific word, w_{err}, which is defined more specifically as

$$w_{err} = \underset{w\in D}{\text{argmax}}\, c(w) \tag{4}$$

where D is the recognition dictionary and $c(w)$ is the number of utterances in the test set which were recognized as w using seven microphones. Through the error analysis, we observed that the averaged feature vectors seems to have smaller magnitude than the original ones. Among 1,444,283 frames of HITNAVI05, 915,645 frames (62.4%) had the averaged feature vector with smaller magnitude than the single microphone (#4) feature vector. The average magnitude of all the averaged feature vectors was 7.2% larger than the average magnitude of the single microphone feature vectors. Accordingly, we tried an extreme example in which all the feature values are zero,

$$w_{\text{zero}} = \underset{w\in D}{\text{argmax}}\, \mathcal{L}(w|\mathbf{0}) \tag{5}$$

where $\mathcal{L}$ denotes the likelihood function given by the decoder, and $\mathbf{0}$ is the MFCC feature vector sequence whose elements are all zero. The result confirmed our prediction that $w_{err} = w_{zero}$. The dashed line in Fig. 3 represents the ratio of w_{zero} appearances in the recognition results over N.

Although w_{zero} was uttered only 20 times (0.55%), it appeared 240 times (6.63%) in the recognition results when we averaged seven input signals. It is clear that the accuracy degradation is proportional to the appearance rate of w_{zero}.

From more detailed analysis of the experimental results, we found that there were 445 utterances (12.3%) which were recognized differently between one microphone and seven microphone experiments. Among them, 44 utterances (1.2%) correspond to incorrect-to-correct transitions from one to seven microphones and 195 utterances (5.4%) correspond to correct-to-incorrect transitions. The rest of 206 utterances (5.7%) had inconsistent misrecognition results between two conditions. For the 43 utterances of the first group, at least one of the other microphones was recognized correctly[†]. A clear advantage of MFCC averaging is that an extreme noise effect on a specific microphone can be smoothed out if other microphones are less contaminated by the noise. For the majority of the 195 utterances of the second group, the misrecognition is caused by w_{zero}. Since MFCC averaging tends to make the feature vector smaller, w_{zero} appears more frequently if the MFCCs are averaged. From these analyses, we figured out that MFCC averaging has an advantage of noise smoothing and a disadvantage of convergence to w_{zero} at the same time.

4.2 Variance Re-Scaling

In the previous experiments, we found that most of the recognition errors were caused by w_{zero}. Since w_{zero} is tightly connected to all-zero feature vectors, we can expect that the recognition rate might be improved if we put the input feature vectors away from the origin by

$$\mathbf{x}_{\text{VR}} = \alpha \mathbf{x}_{\text{ave}} \tag{6}$$

where α is a scaling parameter. We call this modification *variance re-scaling*[††].

In Figs. 4 and 5, the solid lines labeled "MFCC-ave" show the experimental results of HITNAVI05 when we applied either CMN (Fig. 4) or MVN (Fig. 5) and variance re-scaling[†††]. When we applied CMN, MFCC averaging without variance re-scaling ($\alpha = 1.0$) resulted in the poor recognition accuracy of 83.15%, but it can be raised up to 89.78% by variance re-scaling. We also examined the effectiveness of variance re-scaling after MVN. MVN normalizes the variance of each cepstral dimension over the utterance, but we wanted to confirm if such normalization also reduces the variation over the microphones. The experimental results showed that the recognition rate is reasonable (89.06%) even without variance re-scaling, and can be improved to 90.00% with variance re-scaling.

The fact that the accuracy does not degrade by MVN and MFCC averaging explains why variance re-scaling is not necessary for CMU PDA speech database. In addition to using only two microphones in the experiments of Fig. 2, which causes less degradation as shown in Fig. 3, applying DCN reduces the necessity of variance re-scaling because DCN includes normalization of the variance.

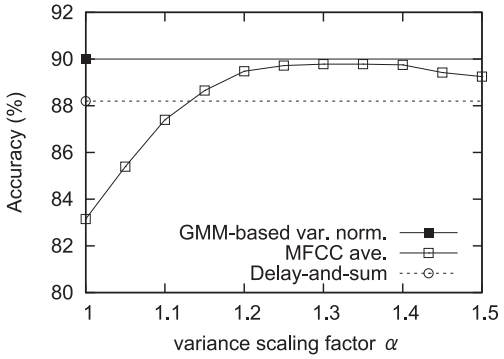


Fig. 4 Experimental results of HITNAVI05 using CMN, MFCC averaging, and variance re-scaling.

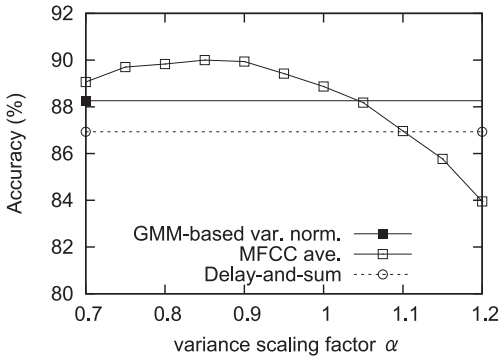


Fig. 5 Experimental results of HITNAVI05 using MVN, MFCC averaging, and variance re-scaling.

4.3 GMM-Based Variance Normalization

So far we confirmed that variance re-scaling can improve the recognition accuracy if we use appropriate value of α. The next problem is how to find the optimal or sub-optimal value of α. To solve this problem, we introduced a typical approach to prepare a Gaussian Mixture Model (GMM) as the reference of variance re-scaling. We created 1,024 mixture clean speech GMM using our training database, and defined the GMM score function

$$S(\mathbf{X}) = \sum_{t=1}^{T} \sum_{j=1}^{M} m_j \mathcal{N}(\mathbf{x}(t) : \mu_j, \Sigma_j) \tag{7}$$

where $\mathbf{x}(t)$ is the feature vector of the time index t, $\mathbf{X} = \{\mathbf{x}(t) | t = 1, 2, \ldots, T\}$ is the sequence of feature vectors, T is the total number of frames, M (= 1024) is the number of

[†]The only exception was that w_{zero} was uttered, all microphones gave incorrect results, and the averaged feature gave w_{zero} correctly.

[††]It must be noted that CMN made the MFCC average over the utterance as zero, so α is proportional to the MFCC variance

[†††]In these experiments, the delay-and-sum beamformer was realized using fixed delay values calculated from the geometric arrangement of the microphones and the passenger's head position.

Gaussian mixtures, m_j is the mixture weight, μ_j and Σ_j are the mean vector and covariance matrix (diagonal covariance matrix was used for simplicity) of the j-th Gaussian mixture, and

$$\mathcal{N}(\mathbf{x} : \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T |\Sigma|^{-1}(\mathbf{x}-\mu)\right) \quad (8)$$

is the Gaussian probability distribution function.

One may think that variance re-scaling must keep the GMM score as high as possible. In fact, the experimental results revealed that it is not true. Instead, variance re-scaling must keep the GMM score as close to the score of single microphone as possible. It could be interpreted that the problem occurs not because the input features are generally too far from the correct model, but because some specific input features are too close to inappropriate models. Based on this empirical assumption, we introduced so-called GMM-based Variance Normalization [21], defined by the following equation.

$$\alpha_{\text{opt}} = \underset{\alpha_i \in A}{\text{argmin}}(|S(\alpha_i \mathbf{X}_{\text{ave}}) - S(\mathbf{X}_0)|) \quad (9)$$

where A is a set of finite candidates of α and $\mathbf{X}_0$ is a feature vector sequence obtained by an arbitrarily selected single microphone.

In Figs. 4 and 5, we plotted the results obtained by GMM-based Variance Normalization by the solid lines. In these experiments, the central microphone (#4) was used as the reference ($\mathbf{X}_0$ of Eq. (8)). In the case of CMN (Fig. 4), the recognition rate was greatly improved to 90.08%. It is even better than the highest recognition rate of variance re-scaling with fixed α, because α was optimized for each utterance in GMM-based Variance Normalization. In the case of MVN (Fig. 5), the contribution of GMM-based Variance Normalization is not as large as in the previous case. However, it is still useful because we can avoid unwanted degradation of the recognition rate due to inappropriate setting of α.

5. Hypothesis-Based Feature Combination

5.1 Basic Concept and Implementation

One of the advantages of working in the cepstral domain it that the human speech can be modeled precisely using hidden Markov models (HMMs) in this domain. It suggests that the feature combination in the cepstral domain can take advantage of the acoustic model as the prior knowledge. This idea led us to the new algorithm, referred to as Hypothesis-Based Feature Combination (HBFC) [22], in which two input features are combined through the bridging roles of the speech recognition hypothesis and acoustic model.

The basic idea of HBFC is based on the finding that the speech recognition hypothesis can be converted to a sequence of the feature vectors using the Viterbi alignment. When we obtain a speech recognition hypothesis, it can be force-aligned to the acoustic model to make an HMM state sequence $\{s_t | t = 1, 2, \ldots, T\}$. As shown in the middle row of Fig. 6, such a state sequence can be converted to a sequence of feature vectors simply by concatenating the mean vectors of the optimal (most likely) Gaussian mixture of each state as follows.

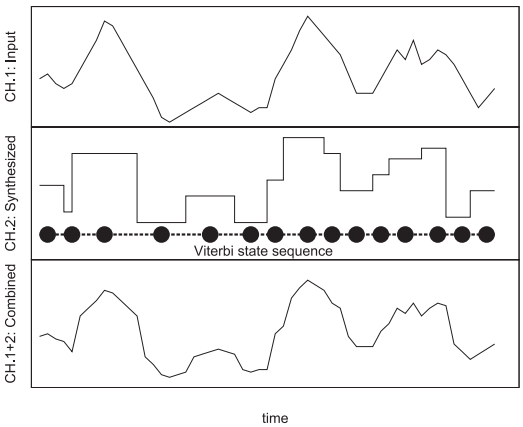


Fig. 6 Conceptual diagram of Hypothesis-Based Feature Combination.

$$\mathbf{y}(t) = \mu_{k(t)}(s_t) \quad (10)$$

$$k(t) = \underset{j=1,2,\ldots,M}{\text{argmax}}\, m_j(s_t)\mathcal{N}(\mathbf{x}(t) : \mu_j(s_t), \Sigma_j(s_t)) \quad (11)$$

where $m_j(s_t)$, $\mu_j(s_t)$, and $\Sigma_j(s_t)$ are the mixture weight, mean, and variance of the j-th mixture of s_t. The new feature vector $\mathbf{y}$ is called *synthesized feature*.

If we obtain such synthesized features from the signal of one channel, they can be combined with the input features of the other channel (top row of Fig. 6) as the weighted average (bottom row of Fig. 6).

$$\mathbf{x}_{\text{HBFC}-1} = (1 - W)\mathbf{y}_1 + W\mathbf{x}_2 \quad (12)$$

$$\mathbf{x}_{\text{HBFC}-2} = (1 - W)\mathbf{x}_1 + W\mathbf{y}_2 \quad (13)$$

It must be noted that there can be two outputs of the feature combination because the algorithm is symmetric for two inputs. We can expect that the resulted feature inherited some part from the hypothesis where the decoder is confident, and some other part from the raw input. This procedure is illustrated in Fig. 7.

As for the optimization of W, we can compare the results obtained with various Ws as we discussed in the previous section. Here, it is useless to compare the likelihoods, because the synthesized feature always provides higher likelihood as they match completely with one of the Gaussians in the force-aligned state sequence.

An interesting extension of HBFC is its recursive execution. As shown in Fig. 8, the outputs of the standard HBFC can be decoded again, and the hypotheses can be used to generate another set of the synthesized features, and they can be combined with the other channel again. Such a procedure can be repeated unlimitedly in theory, although the combined feature is supposed to converge to a certain value.

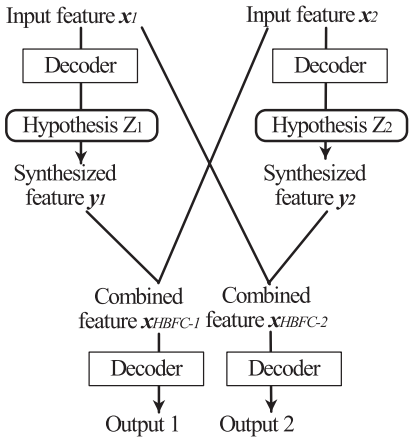


Fig. 7 Flowchart of simple version HBFC.

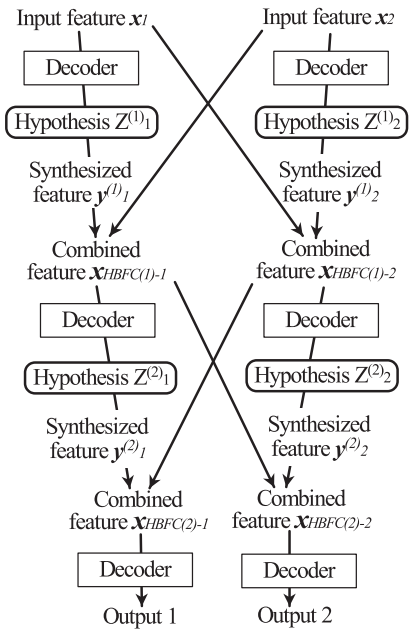


Fig. 8 Flowchart of recursive implementation of HBFC.

When we have more than two inputs, implementation of HBFC is not straightforward [23]. If we have N input channels and we can use the input or synthesized feature for each channel arbitrarily, there can be $2^N - 1$ ways to combine them. However, we used only the simplest implementation, in which only one input was used to generate hypotheses, and the other channels are used as raw features. The procedural flow is shown in Fig. 9, where the input features of $N-1$ channels $(x_1, x_2, \ldots, x_{N-1})$ are first averaged, and the resulted feature is again combined with the synthesized feature y_0 as

$$\mathbf{x}_{\mathrm{HBFC}} = (1 - W)\mathbf{x}_{\mathrm{ave}} + W\mathbf{y}_0 \tag{14}$$

where $\mathbf{x}_{ave}$ is the average of $N-1$ input features. In this case, the procedure is not symmetric, and we obtain only one feature vector sequence if we fix the value of W[†]. However, we must pay attention to variance re-scaling because we are dealing with many inputs.

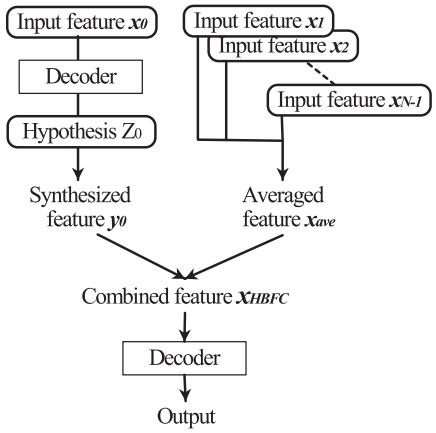


Fig. 9 Flowchart of HBFC applied to more than two input channels.

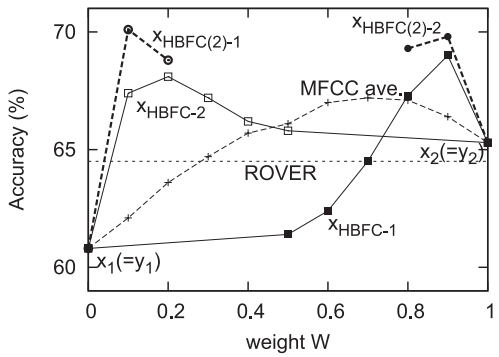


Fig. 10 Experimental results of HBFC applied to CMU PDA speech database. All feature vectors were normalized by DCN before combined.

5.2 Experimental Results

We carried out a set of experiments to evaluate the effectiveness of HBFC using both CMU PDA speech database and HITNAVI05.

Figure 10 show the results of CMU PDA speech database. These experiments correspond to Figs. 7 and 8. Since the synthesized feature has a very strong effect to raise the likelihood of the corresponding hypothesis, the recognition results of the combined feature are almost the same as those of the synthesized feature if the weight of the synthesized feature is larger than 0.5. However, if its weight is around 0.1, the accuracy increased rapidly. The best accuracy of 69.0% was obtained by $\mathbf{x}_{\mathrm{HBFC}-1}$ with $W = 0.9$, which corresponds to 5.5% relative error reduction from the best case ($W = 0.7$) of MFCC average. When we applied HBFC recursively (in this case, W of the first combination was fixed at 0.1/0.9), the result was even better. The best accuracy of 70.1% was obtained by $\mathbf{x}_{\mathrm{HBFC}(2)-1}$ with $W = 0.1$,

[†]We have an option in choosing the channel for the synthesized feature. Hence there can be seven different outputs, just like there are two different outputs in Figs. 7 and 8, although other outputs were not used in these experiments.

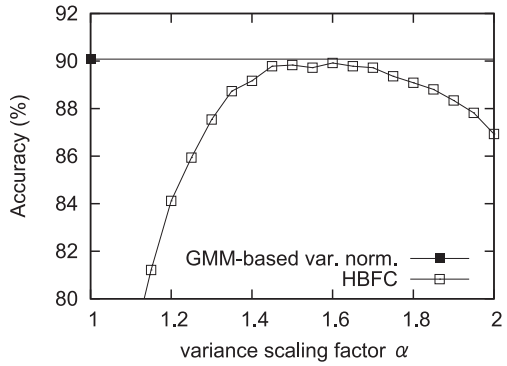


Fig. 11 Experimental results of HBFC applied to HITNAVI05. All feature vectors were normalized by CMN before combined.

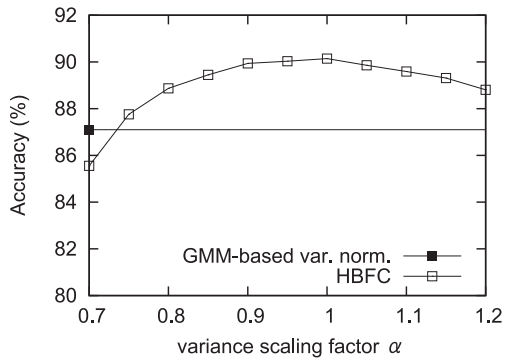


Fig. 12 Experimental results of HBFC applied to HITNAVI05. All feature vectors were normalized by MVN before combined.

which corresponds to 8.8% relative error reduction.

Since the algorithm of HBFC include multiple decoding paths, ROVER [8] could be another option to compare with. In the case of two input streams, ROVER is realized by the DP-based alignment of two hypotheses and likelihood comparison. As shown in the figure, we obtained by ROVER the recognition accuracy of 64.5%, which is slightly better than the delay-and-sum beamformer, but worse than Channel 2 only. (The score for a null arc is an adjustable parameter of ROVER, and we used the optimized value for the test set)

Figures 11 and 12 show the results of HITNAVI05. These experiments correspond to Fig. 9. In these experiments, W was fixed at 0.1, and the effect of variance re-scaling was investigated. If the original feature vector was normalized by CMN (Fig. 11), the recognition accuracy without variance re-scaling was miserable (68.15%). However, if α was set appropriately as 1.6, the accuracy raised up to 89.92%. It was still improved by introducing GMM-based Variance Normalization to 90.08%, which is slightly better than the that of MFCC averaging and GMM-based Variance Normalization (90.00%). In contrast, if the original feature vector was normalized by MVN (Fig. 12), the accuracy was acceptable even without variance re-scaling (90.14%) and this is in fact the optimal setting of α. It is even better than the case of MFCC averaging (Fig. 5), in which the best result was not obtained with $\alpha = 1.0$. GMM-based Variance Normalization caused slight degradation in this case, and the accuracy was 87.10%, which is slightly worse than that of MFCC averaging and GMM-based Variance Normalization (88.26%). Since more adjustable parameters are involved in these experiments, additional experiments would be needed to achieve the best performance of feature combination.

6. Conclusions

In this paper, we have introduced a new feature combination framework in the cepstral domain. In addition to the recognition accuracy improvement, this framework has a lot of merits (*ilities*) for use in real applications.

Evaluation experiments showed that the speech recognition accuracy was improved even by simple MFCC averaging. However, the accuracy was degraded when the number of inputs was increased. We solved this problem by introducing the concept of variance re-scaling, and proposed an algorithm called GMM-based Variance Normalization to optimize the scaling factor automatically.

Another advantage of feature combination in the cepstral domain is that the acoustic model can be absorbed seamlessly in the framework of feature combination. This means that we can use the prior knowledge of the speech model, and the recognition accuracy can be improved even more. In particular, the experimental results using a PDA-like mock-up with two microphones showed great improvement by this approach, Hypothesis-Based Feature Combination (HBFC). HBFC was also effective for the database collected in a moving car using a seven-microphone linear array, but the advantage was reduced when it was combined with GMM-based Variance Normalization. Although the effectiveness of the proposed algorithms vary depending on the task and environment, we expect that more investigation would present a clear guideline to implement these algorithms more effectively.

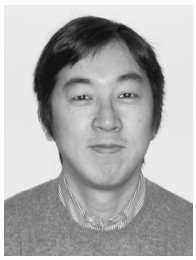
Acknowledgments

The authors are thankful to Prof. Sadaoki Furui of Tokyo Institute of Technology and Prof. Tetsunori Kobayashi of Waseda University for their valuable comments. A part of this work was supported by New Energy and Industrial Technology Development Organization (NEDO), Japan.

References

[1] J. Voas, "Software's secret sauce: The *-ilities*," IEEE Softw., vol.21, no.6, pp.14–15, Nov. 2004.
[2] Wikipedia, http://en.wikipedia.org/wiki/Ilities
[3] S.L. Leese, "Microphone arrays," in Noise Reduction in Speech Applications, ed. G.M. Davis, pp.179–197, CRC Press, 2002.
[4] W. Kellermann, "A self steering digital microphone array," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, 1991.
[5] L.J. Griffiths and C.W. Jim, "An alternative approach to linearly constrained adaptive beamforming," IEEE Trans. Antennas Propag.,

vol.30, no.1, pp.27–34, 1982.
[6] M. Mizumachi and M. Akagi, "Noise reduction by paired-microphones using spectral subtraction," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998.
[7] P. Comon, "Independent component analysis, a new concept?," Signal Process., vol.36, no.3, pp.287–314, 1994.
[8] J.G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, USA, 1997.
[9] T. Yamada, S. Nakamura, and K. Shikano, "Distant-talking speech recognition based on a 3-D Viterbi search using a microphone array," IEEE Trans. Speech Audio Process., vol.10, no.2, pp.48–56, 2002.
[10] Y. Obuchi, "Multiple-microphone robust speech recognition using decoder-based channel selection," Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea, 2004.
[11] Y. Obuchi, "Noise robust speech recognition using delta-cepstrum normalization and channel selection," Electron. Commun. Jpn. 2, Electron., vol.89, no.7, pp.9–20, 2006.
[12] T. Utsuro, Y. Kodama, T. Watanabe, H. Nishizaki, and S. Nakagawa, "An empirical study on multiple LVCSR model combination by machine learning," Proc. HLT/NAACL 2004, Boston, MA, USA, 2004.
[13] S. Matsuda, T. Jitsuhiro, K. Markov, and S. Nakamura, "ATR parallel decoding based speech recognition system robust to noise and speaking styles," IEICE Trans. Inf. & Syst., vol.E89-D, no.3, pp.989–997, March 2006.
[14] Y. Shimizu, S. Kajita, K. Takeda, and F. Itakura, "Speech recognition based on space diversity using distributed multi-microphones," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, 2000.
[15] W. Li, T. Shinde, H. Fujimura, C. Miyajima, T. Nishino, K. Itou, K. Takeda, and F. Itakura, "Multiple regression of log spectra for in-car speech recognition using multiple distributed microphones," IEICE Trans. Inf. & Syst., vol.E88-D, no.3, pp.384–390, March 2005.
[16] CMU PDA Database, http://www.speech.cs.cmu.edu/databases/pda/
[17] T.M. Sullivan, Multi-microphone correlation-based processing for robust automatic speech recognition, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.
[18] Y. Obuchi and N. Hataoka, "Development and evaluation of speech database in automotive environments for practical speech recognition systems," Proc. INTERSPEECH2006 - ICSLP, Pittsburgh, PA, USA, 2006.
[19] Y. Obuchi, N. Hataoka, and R.M. Stern, "Normalization of time-derivative parameters for robust speech recognition in small devices," IEICE Trans. Inf. & Syst., vol.E87-D, no.4, pp.1004–1011, April 2004.
[20] Y. Obuchi, "Mixture weight optimization for dual-microphone MFCC combination," Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, San Juan, Puerto Rico, 2005.
[21] Y. Obuchi and N. Hataoka, "Robust feature combination for speech recognition using linear microphone array in a car," Proc. Biennial on DSP for In-Vehicle and Mobile Systems, Istanbul, Turkey, 2007.
[22] Y. Obuchi, "Hypothesis-based feature combination for dual-microphone speech recognition," Proc. Joint Workshop on Hands-Free Speech Communication and Microphone Arrays, Piscataway, NJ, USA, 2005.
[23] Y. Obuchi and N. Hataoka, "Hypothesis-based feature combination of multiple speech inputs for robust speech recognition in automotive environments," Proc. INTERSPEECH2006 - ICSLP, Pittsburgh, PA, USA, 2006.



Yasunari Obuchi received the B.S. degree and the M.S. degree in physics in 1988 and 1990 respectively, and the Ph.D. in information science and technology in 2006, all from the University of Tokyo. Since 1992, he has been working for Central Research Laboratory and Advanced Research Laboratory, Hitachi Ltd. He worked for Carnegie Mellon University as a Visiting Researcher from 2002 to 2003. He has also been a Visiting Researcher at Waseda University from 2005 to the present. Currently he is a Senior Researcher of Central Research Laboratory, Hitachi Ltd. His research interests include robust speech recognition, spoken dialog systems, speech recognition in small devices, speech-to-speech translation, language identification, emotion recognition, and microphone arrays. Dr. Obuchi was a co-recipient of the Technology Development Award of the Acoustical Society of Japan in 2000. He is a member of IEEE, ISCA, IPSJ, and ASJ.

Nobuo Hataoka received the B.S.E.E. degree and the M. Sc. degree in Electrical and electronics Engineering from Tohoku University, in 1976 and 1978, respectively, and the Ph.D in Engineering in 1992 from Tohoku University. He joined Central Research Laboratory, Hitachi Ltd. in 1978, and he spent one year from 1988 to 1989 as Visiting Researcher at Carnegie Mellon University in U.S.A. From 1989 to 1993, he was Laboratory Manager of Hitachi Dublin Laboratory in Ireland. From 1993, he worked for Central Research Laboratories, Hitachi Ltd. as Chief Research Scientist. He is currently Professor of Tohoku Institute of Technology in Sendai, Japan. His current research interests include media implementation on microprocessors, and algorithm development on speech recognition, speech synthesis, speech translation, and artificial intelligence. Dr. Hataoka is a member of the IEEE and the Acoustical Society of Japan.