DEVELOPMENT OF ROBUST SPEECH RECOGNITION MIDDLEWARE ON MICROPROCESSOR

N. Hataoka, H. Kokubo, Y. Obuchi, and A. Amano

Central Research Laboratory, Hitachi Ltd.
Kokubunji, Tokyo 185, JAPAN

ABSTRACT

We have developed speech recognition middleware on a RISC microprocessor which has robust processing functions against environmental noise and speaker differences. The speech recognition middleware enables developers and users to use a speech recognition process for many possible speech applications, such as car navigation systems and handheld PCs. In this paper, we report implementation issues of speech recognition process in middleware of microprocessors and propose robust noise handling functions using ANC (Adaptive Noise Cancellation) and noise adaptive models. We also propose a new speaker adaptation algorithm, in which the relationships among HMMs (Hidden Markov Models) transfer vectors are provided as a set of pre-trained interpolation coefficients. Experimental evaluations on 1000-word vocabulary speech recognition showed promising results for both robust processing functions of the proposed noise handling methods and the proposed speaker adaptation method.

1. INTRODUCTION

Recently, quite a few efforts have been made to realize sophisticated user interfaces which have speech processing techniques. Especially, speech recognition technology has made a great progress, and many commercially available products have been announced in these days. However, many technical problems are still existing to use speech recognition systems in real applications. Robustness of speech recognition in noisy environment and robustness for different speakers' variations are main and key issues. Also, how to implement speech recognition process is another important issue to make speech recognition easy to use and to realize speech recognition applications successfully.

First, regarding implementation issues, we have developed speech recognition middleware on RISC microprocessors as one of SuperH Speech Middleware functions. Second, to realize robust speech recognition under environmental noise, many approaches such as spectral subtraction (SS) methods[1], Adaptive Noise Cancellation (ANC)[2], and speech model adaptation techniques based on HMMs decomposition have been proposed[3][4]. In this paper, we have modified these techniques to realize robust speech recognition middleware. Robust speech detection using ANC method have been implemented and the speech model adaptation by adding environmental noise have been used in the middleware developed. Finally, speaker adaptation mechanism has been implemented in the middleware by optimizing process time and data/process memory sizes.

In this paper, we report detailed specifications of SuperH Speech Middleware and implementation results of the proposed robust processing functions for environmental noise and speaker difference. The calculation speed and the memory size are limited in the middleware, but we have achieved real-time recognition of 1000 words with high recognition rate.

2. SuperH SPEECH MIDDLEWARE

2.1 Middleware Specifications

Middleware is a kind of library set which connects hardware and user applications. We have developed speech recognition middleware on a RISC microprocessor. The middleware helps to make an application which has the speech recognition function. Table 1 shows the specification of our RISC microprocessor SuperH¤ Risc Engine (SH-3) and the speech recognition middleware. The operation speed and the memory size are limited. We are using phonemic speech segments as HMM units. To reduce calculation burden, semi-continuous HMMs and tied mixtured 3-dimensional models have been used. Moreover, we introduced several approximation search techniques to save the calculation time. Thus, the middleware achieved the performance of 93% recognition rate for 1000 word vocabulary with only 0.6 second response time.

Table 1: Specification of SuperH SR Middleware

item	specification
Speech Model	Phonemic Speech Units / Semi-continuous HMM
Operation Speed	60 MHz
External Bus	60 MHz / 32 bit
Sampling	11.025 kHz / 16 bit
Frame Lengthl / Period	20 ms / 10ms
Processing Time	14 ms / frame
Response Time	~ 0.6 sec
Vocabulary Size	1000
Memory Size	256 kByte (phonetic model etc.) 500kByte (work)

2.2 Middleware Architecture

Figure 1 shows an example of the middleware architecture implemented on a SuperH board. The SH-3 has 60MHz cycle process power. The fundamental middleware which has 1000-word speech recognition ability needs 256kByte ROM (Read Only Memory) as data/program memory and 560kByte RAM (Random Access Memory) as work memory. The input speech is digitized by an 11.025kHz-sampling A/D converter, and processed by the middleware via. interface bus. Finally, the recognized results are shown to display terminals through a RS232C Interface.

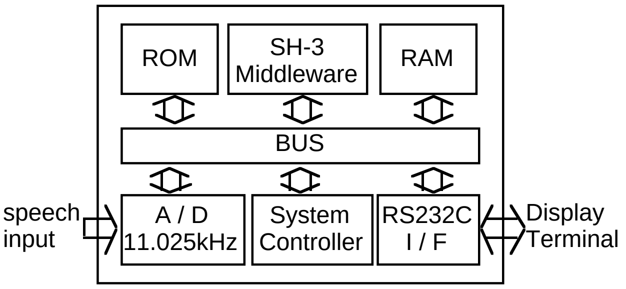


Figure 1: Example of System Architecture

3. NOISE HANDLING METHOD

3.1 ANC (Adaptive Noise Cancellation)

The ANC is used for a noise reduction technique which makes speech interval detection easy and precise. Figure 2 shows a block-diagram of ANC for the speech interval detection. The ANC needs normally two microphones, one for speech data and the other for noise data. In the middleware developed, a 300-tap adaptive digital filter has been used to reduce speech input data which includes noise data.

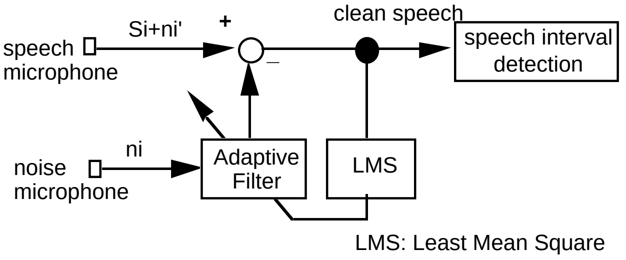


Figure 2: Block-diagram of ANC for Speech Interval Detection

3.2 Noise Adapted Speech HMMs

Many noise adaptation techniques based on decomposition and composition of speech and noise HMMs have been proposed and showed promising results[3][4]. We have modified these techniques and combined with the ANC speech detection technique to realize robust and concise speech recognition middleware.

The noise adapted HMMs are extracted by the processing flow shown in Figure 3[4]. Noise HMMs are calculated by the environmental noise and added to the stored HMMs which has been created using clean speech data. To add noise HMMs to the clean HMMs, two transform processes of cosine transform and exponential transform are used. In the convolution process, the following calculation is done to extract noise adapted HMMs in the linear spectrum domain.

$$R = S + k(SNR)*N \qquad (1)$$

where, S, N, and R show clean HMMs, noise HMMs and adapted HMMs in linear spectrum domain, respectively. k is multiple parameter determined by Signal-Noise-Ratio (SNR) of environments. The combined HMMs are extracted by Log Transform and Inverse Cosine Transform from R of the equation(1).

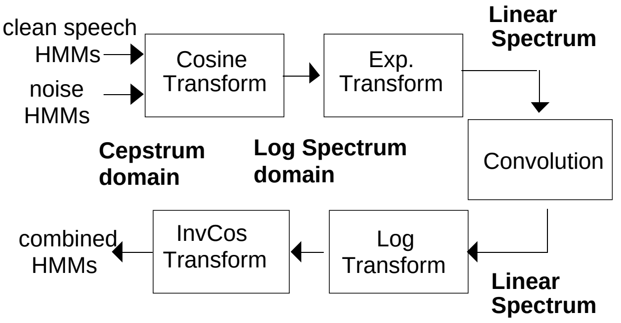


Figure 3: Noise Adaptation Flow

Figure 4 shows experimental evaluation results of the noise adaptation HMMs. We used two types noise, namely car running noise and car air conditioner noise as noise environments.

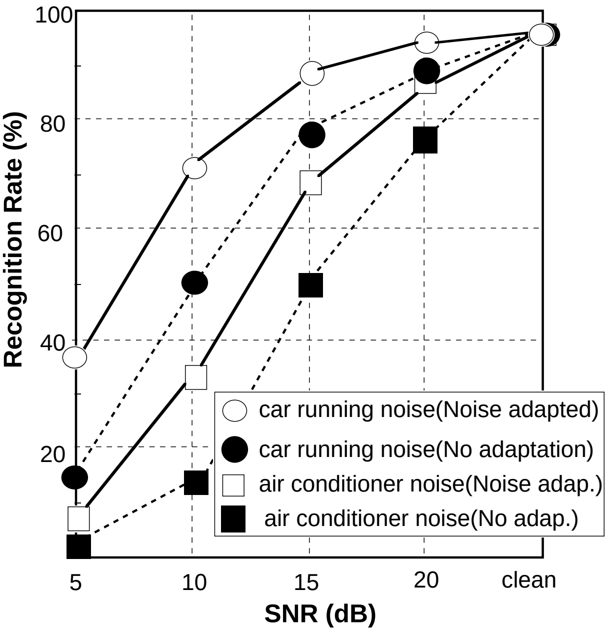


Figure 4: Experimental Evaluation Results

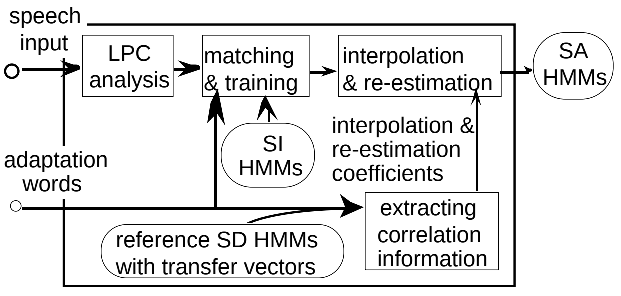
A speech recognition task is Japanese 1000 railway station names. Four types of SNR conditions were evaluated using 1000 station names uttered by 28 speakers. The air conditioner noise cases showed lower recognition results than car running noise cases. In car running noise cases, almost half of errors have been improved using the proposed combination method of ANC speech interval detection and noise adapted HMMs.

4. SPEAKER ADAPTATION

Speaker adaptation is one of the adaptation schemes, where the speaker independent(SI) HMMs are modified to the speaker adapted(SA) HMMs using small amount of adaptation speech data. The SI HMMs include many models corresponding to many types of the phonetic units. However, the adaptation data include only few models, so the problem is how to adapt those models that do not appear in the adaptation data. Moreover, even the models that appear in the adaptation data are not adapted correctly, because there are not enough data for each model. For these problems, many interpolation and smoothing techniques have been proposed[5][6]. All of these techniques are based on the assumption that the transfer vector field should be smooth. This assumption helps the good estimation of transfer vectors, but it brings the limit of the precise estimation. We propose a new speaker adaptation algorithm, which does not assume the smoothness of the transfer vector field. We prepare many reference speaker Dependent(SD) HMMs, and calculate correlation between each other. We use this information, instead of the smoothness assumption, to estimate unknown and uncertain transfer vectors. Similar approach was applied to the word recognition system based on dynamic time warping(DTW) by Furui[7].

4.1 Interpolation with Pre-trained Coefficients

Figure 5 shows a block-diagram of the proposed speaker adaptation algorithm named Interpolation with Pre-trained Coefficients(IPTC)[8]. The adaptation speech input comes with the corresponding adaptation word. The speech input is transformed to the feature vectors by LPC analysis, and then matched with the adaptation word. In the matching process, the time series of feature vectors are segmented into phonetic units using SI HMMs and Viterbi algorithm.



SI: Speaker Independent
SD: Speaker Dependent
SA: Speaker Adapted

Figure 5: Block-diagram of Speaker Adaptation Algorithm.

After matching, each HMM model is trained using MAP estimation. To reduce the calculation time and to avoid the over-learning, only the mean vectors of the Gaussian probability density are trained, and the covariance of SI HMMs are used in SA HMMs. In the semi-continuous HMM of our system, each HMM has two states and each state has three Gaussian mixtures, and all of those mixtures are tied to make HMM codebooks. Therefore, the mean vectors of those HMM codebooks are adapted. A transfer vector is defined as a difference between mean vectors before and after training.

$$V_{pi} = \hat{\mu}_{pi} - \mu_{pi} \qquad (2)$$

where Vpi is the i-th element of the transfer vector of the p-th HMM codebook. $\hat{\mu}_{pi}$ and μ_{pi} correspond to the mean vectors before and after training. Parallel to this process, correlation information among transfer vectors is extracted from reference SD HMMs. Correlation information is represented as linear combination coefficients for interpolation and re-estimation.

After matching and training procedures, HMM codebooks are divided into two groups; HMM codebooks that appear in the adaptation data (trained codebooks) and HMM codebooks that do not appear in the adaptation data (untrained codebooks). For untrained codebooks, interpolation is carried out using the following equation.

$$V_{pi} = \sum_{q \in N^{(I)}(p)} C^{(I)}_{pq} V_{qi} \qquad (3)$$

where $N^{(I)}(p)$ is the set of trained neighbors of the p-th HMM codebook, and $C^{(I)}pq$ is the interpolation coefficient of q-th HMM codebook. To avoid errors originated in data sparsity, a re-estimation procedure for all HMM codebooks follows the interpolation procedure.

$$V'_{pi} = \sum_{q \in N^{(R)}(p)} (C^{(R)}_{pq} V_{qi} + V_{pi}) / 2 \qquad (4)$$

where Vpi is the i-th element of the trained or interpolated transfer vector, and V'pi is the i-th element of the re-estimated transfer vector. The neighbor set $N^{(R)}(p)$ includes both trained and interpolated HMM codebooks, and $C^{(R)}pq$ is the re-estimation coefficient of q-th HMM codebook. Equation (4) becomes the same as that of the VFS in [5] if the values $C^{(R)}pq$ are calculated only from the distances between HMM codebooks. For simplicity, we add the original transfer vector and the estimated transfer vector with the same ratio, but the ratio can be changed if necessary.

In our algorithm, the coefficients $C^{(I)}pq$ and $C^{(R)}pq$ are calculated beforehand using transfer vectors of reference SD HMMs. That is the reason why we named the proposed algorithm "Interpolation with Pre-trained Coefficients (IPTC)." We prepare 36 reference SD HMMs from 36 speakers, which are made using 216 phonetically balanced words.

4.2 Experimental Evaluation Results

To evaluate the performance of the adaptation, the proposed algorithm IPTC was compared with MAP-VFS on the recognition task of 1000 words. The vocabulary consists of Japanese railway station names. Each word is transformed to the series of HMM states. The number of HMM states varies from 18 (3 phonemes) to 86 (20 phonemes), and the average is 39.54 (8.38 phonemes). Figure 6 shows the average recognition rate (circles) and the recognition rate for the speaker whose recognition rate for the SI HMMs is the worst in the six speakers (squares). In the experiment, 300 of 1000 utterances were picked up for each of six speakers. 50 of them are used as the adaptation utterances, and 250 are used for test. The average duration length of those 50 words is 42.96 states (9.24 phonemes). In one, two, and five word adaptation, 50 adaptation utterances are divided into 50, 25, and 10 subsets respectively. In 10, 20, and 30 word adaptation, 50 adaptation utterances are divided into 10 overlapping subsets, such as #1~#10, #6~#15, #11~#20, etc. (# denotes the word No.) The recognition rate for a speaker is calculated by averaging over all of those subsets, where the same testing data are used. The recognition rate for all speakers is calculated by averaging over six speakers. As shown in Figure 6, the recognition rate of IPTC is lower than MAP-VFS when adapted by one or two words. However, IPTC brings higher recognition rate when adapted by more than five words. Since interpolation and re-estimation procedures tend to depend on fewer neighboring HMMs in IPTC than in MAP-VFS, wrong adaptation by the small number of adaptation words may propagate in the interpolation and re-estimation processes. The adaptation by IPTC reduces 28.5% of recognition errors using 10 adaptation words, and 52.7% using 50 adaptation words, while MAP-VFS reduces only 22.9% and 38.4% respectively.

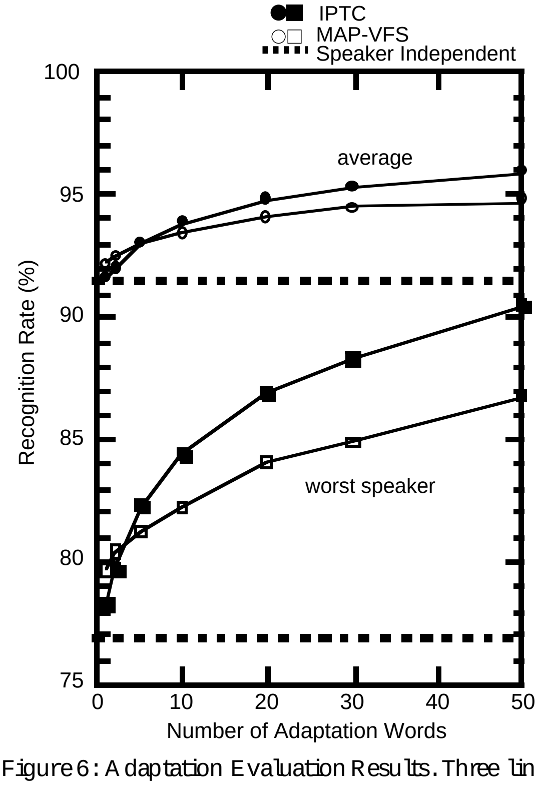


Figure 6: Adaptation Evaluation Results. Three lines on top represent average of six speakers, and three lines on bottom represent the speaker whose recognition rate is the worst in the six. IPTC results are plotted by black and MAP-VFS results are plotted by white.

The adaptation performance is remarkable for the worst speaker, where 32.6% are reduced using 10 words and 58.6% are reduced using 50 words (23.1% and 43.1% by MAP-VFS)

5. SUMMARY

This paper described a new implementation of speech recognition as middleware on RISC microprocessors. To realize robust processing functions against environmental noise and speaker differences, we have developed robust noise handling techniques using ANC (Adaptive Noise Cancellation) and noise adaptive models. We also have proposed a new speaker adaptation algorithm named Interpolation with Pre-trained Coefficients (IPTC). The algorithm uses interpolation and re-estimation coefficients which are calculated from the transfer vectors of the reference SD HMMs. The proposed robust processing techniques have been implemented as a part of the speech recognition middleware on RISC microprocessors. Experimental results have shown that the developed middleware compares favorably with other speech recognition systems.

6. ACKNOWLEDGMENTS

The authors would like to thank Dr. Shinya Ohba, Mr. Nakagawa, and Mr. Tetsuji Tohge of Hitachi Semi-conductor Division for implementing and evaluating the proposed algorithm on Hitachi's SuperH RISC microprocessors.

7. REFERENCES

[1] Boll, S.F. Suppression of Acoustic Noise in Speech using Spectral Subtraction, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol ASSP-27, No 2, pp.113-120, Ap 1993.

[2] Sakamoto, M., Oshima, Y. Word Recognition in a Noisy Environment using Adaptive Noise Cancelling, Proc. of Japan Acoustical Society, 2-Q-1, pp.131-132, 1996.

[3] Varga, A.P., and Moore, R.K. Hidden Markov Model Decomposition of Speech and Noise, Proc. of ICASSP90, pp.845-848, 1990.

[4] Martin, F., Shikano, K., Minami, Y., and Okabe, Y. Recognition of Noisy Speech by using Composition of Hidden Markov Models, Proc. of Japan Acoustical Society, 1-7-10, pp.65-66, Oct., 1992.

[5] Shinoda, K., Iso, K., and Watanabe, T. Speaker Adaptation for Demi-syllable Based Continuous Density HMM, Proc of ICASSP91, S13.7, pp.857-860, 1991.

[6] Tonomura, M., Kosaka, T., and Matsunaga, S. Speaker Adaptation Based on Transfer Vector Field Smoothing Using Maximum a Posteriori Probability Estimation, Proc of ICASSP95, pp.688-691, (1995)

[7] Furui, S. A Training Procedure for Isolated Word Recognition Systems, IEEE Trans. Acoustics, Speech, and Signal Processing, Vol ASSP-28, No 2, pp.129-136, 1980.

[8] Obuchi, Y., Amano, A., and Hataoka, N. A Novel Speaker Adaptation Algorithm and Its Implementation on a Risc Microprocessor, IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), December, 1997 (to be appeared)