

PCA-Based Speech Enhancement for Distorted Speech Recognition

Tetsuya Takiguchi, Yasuo Ariki

Department of Computer and System Engineering, Kobe University, Japan

Email: {takigu, ariki}@kobe-u.ac.jp

Abstract— We investigated a robust speech feature extraction method using kernel PCA (Principal Component Analysis) for distorted speech recognition. Kernel PCA has been suggested for various image processing tasks requiring an image model, such as denoising, where a noise-free image is constructed from a noisy input image [1].

Much research for robust speech feature extraction has been done, but it remains difficult to completely remove additive or convolution noise (distortion). The most commonly used noise-removal techniques are based on the spectral-domain operation, and then for speech recognition, the MFCC (Mel Frequency Cepstral Coefficient) is computed, where DCT (Discrete Cosine Transform) is applied to the mel-scale filter bank output. This paper describes a new PCA-based speech enhancement algorithm using kernel PCA instead of DCT, where the main speech element is projected onto low-order features, while the noise or distortion element is projected onto high-order features. Its effectiveness is confirmed by word recognition experiments on distorted speech.

Index Terms— kernel PCA, distorted speech, feature extraction, speech enhancement

I. INTRODUCTION

In hands-free speech recognition, one of the key issues for practical use is the development of technologies that allow accurate recognition of noisy and reverberant speech. Current speech recognition systems are capable of achieving impressive performance in clean acoustic environments. However, if the user speaks at a distance from the microphone, the recognition accuracy is seriously degraded by the influence of additive and convolution noise.

Convolution distortion (noise) is usually caused by telephone channels, microphone characteristics, reverberation, and so on. Its effect on the input speech appears as a convolution in the wave domain and is represented as a multiplication in the linear-spectral domain. Conventional normalization techniques, such as CMS (Cepstral Mean Subtraction) and RASTA, have been proposed, and their effectiveness has been confirmed for the telephone channel or microphone characteristics, which have a short impulse response [2]. When the length of the impulse response is shorter than the analysis window used for the spectral analysis of speech, those methods are effective.

This paper is based on "Robust Feature Extraction Using Kernel PCA," by T. Takiguchi and Y. Ariki, which appeared in the Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, May 2006. © 2006 IEEE.

However, as the length of the impulse response of the room reverberation (acoustic transfer function) becomes longer than the analysis window, the performance degrades.

To solve problems caused by additive and convolution noise, many methods have been presented in robust speech recognition (e.g. [3]–[8]), but it is difficult to completely remove non-stationary or unknown noise. The most commonly used noise-removal techniques are based on the spectral-domain operation, and then for speech recognition, the MFCC (Mel Frequency Cepstral Coefficient) is computed, where DCT is applied to the mel-scale filter bank output.

In current speech recognition technology, the MFCC (Mel Frequency Cepstral Coefficient) has been widely used. The feature is derived from the mel-scale filter bank output using DCT (Discrete Cosine Transform). The low-order MFCCs account for the slowly changing spectral envelope, while the high-order ones describe the fast variations of the spectrum. Therefore, a large number of MFCCs is not used for speech recognition because we are only interested in the spectral envelope, not in the fine structure.

Ref. [9] has investigated a suitable transformation based on PCA that can reflect the statistics of speech data better than DCT to compute the MFCC. In [10], a PCA-based approach for speech enhancement is proposed, where PCA is applied to the wave domain instead of the Fourier Transform. In [11], the filter-bank coefficients are estimated by applying PCA to the FFT spectrum. In [12], the effect of a PCA filter on room reflections is investigated for microphone-array systems. A feature extraction approach using kernel PCA has been also proposed in [13] and [14], where the kernel PCA was applied only to the low-order MFCCs that account for the spectral envelope.

In this paper, we investigate robust feature extraction using kernel PCA instead of DCT, where kernel PCA is applied to the mel-scale filter bank output (Fig. 1) because we expect that kernel PCA will project the main speech element onto low-order features, while noise (reverberant) elements will be projected onto high-order ones. Our recognition results show that the use of kernel PCA instead of DCT provides better performance for reverberant speech.

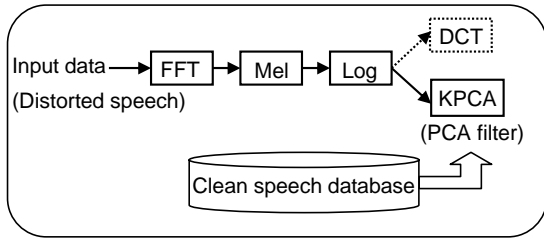


Figure 1. Feature extraction using kernel PCA. PCA filter represents the statistics of clean speech data.

II. FEATURE EXTRACTION USING KERNEL PCA

A. Speech Enhancement

The distorted speech, $X_n(\omega)$, is generally considered as the multiplication of the clean speech and the convolution noise:

$$X_n(\omega) = S_n(\omega) \cdot H_n(\omega) \quad (1)$$

where $S_n(\omega)$ and $H_n(\omega)$ are the short-term linear spectrum for the clean speech and the convolution noise (acoustic transfer function) of the frequency ω at the n -th frame (n -th analysis window), respectively.

The length of the acoustic transfer function is generally longer than that of the window. Therefore, the observed distorted spectrum is approximately represented by

$$X_n(\omega) \approx S_n(\omega) \cdot H_n(\omega). \quad (2)$$

The multiplication can be converted to addition in the log-spectral domain as follows:

$$X_{\log-n}(\omega) \approx S_{\log-n}(\omega) + H_{\log-n}(\omega), \quad (3)$$

where $X_{\log-n}(\omega)$, $H_{\log-n}(\omega)$, and $S_{\log-n}(\omega)$ are the log spectra for the observed signal, acoustic transfer function (convolution noise), and speech signal, respectively.

Next, we consider the following filtering based on PCA in order to extract the feature of clean speech only,

$$\hat{S} = \mathbf{V} X_{\log}. \quad (4)$$

The filter (eigenvector matrix), \mathbf{V} , is derived by the eigenvalue decomposition of the centered covariance matrix of a clean speech data set, in which the filter consists of the eigenvectors corresponding to the L dominant eigenvalues (L eigenvectors corresponding to the biggest L eigenvalues).

$$\mathbf{V} = [\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(L)}] \quad (5)$$

Due to the orthogonality, the component of the convolution noise belonging to the subspace $[\mathbf{v}^{(L+1)}, \dots, \mathbf{v}^{(M)}]$ is canceled by this filtering operation. However, as shown in (3), the observed signal is approximately represented under the assumption of non-correlation between the clean speech and the convolution noise. In this paper, we focus on non-linear PCA (kernel PCA) in order to deal with the influence of the approximation. Kernel PCA first maps the data into high-dimensional feature space by a non-linear function and then performs linear PCA on the mapped data. We can expect that noise will be canceled in the high-dimensional space.

B. Kernel PCA

PCA is a powerful technique for extracting structure from possibly high-dimensional data sets. But it is not effective for data with non-linear structure. In kernel PCA, the input data with nonlinear structure is transformed into a higher-dimensional feature space with linear structure, and then linear PCA is performed in the high-dimensional space [15].

Given the mel-scale filter bank output (log spectrum) \mathbf{x}_j at j -frame, the covariance matrix is defined as

$$C = \frac{1}{N} \sum_{j=1}^N \bar{\Phi}(\mathbf{x}_j) \bar{\Phi}(\mathbf{x}_j)^T, \quad (6)$$

$$\bar{\Phi}(\mathbf{x}_j) = \Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{j=1}^N \Phi(\mathbf{x}_j), \quad (7)$$

where the total number of frames is N , and Φ is a nonlinear map.

$$\Phi : \mathbf{R}^d \rightarrow \mathbf{R}^\infty \quad (8)$$

Note that the data in the high-dimensional space could have an arbitrarily large, possibly infinite, dimensionality, and d is the dimension of \mathbf{x} .

We now have to find eigenvalues λ and eigenvectors \mathbf{v} satisfying

$$\lambda \mathbf{v} = C \mathbf{v}, \quad (9)$$

$$\lambda(\bar{\Phi}(\mathbf{x}_k) \cdot \mathbf{v}) = (\bar{\Phi}(\mathbf{x}_k) \cdot C \mathbf{v}), \quad k = 1, \dots, N \quad (10)$$

Also, there exist coefficients α_i such that

$$\mathbf{v} = \sum_{i=1}^N \alpha_i \bar{\Phi}(\mathbf{x}_i). \quad (11)$$

Substituting (6) and (11) in (10), we get for the left side of the equation

$$\begin{aligned} \lambda(\bar{\Phi}(\mathbf{x}_k) \cdot \mathbf{v}) &= \lambda \sum_i \alpha_i \bar{\Phi}(\mathbf{x}_k) \cdot \bar{\Phi}(\mathbf{x}_i) \\ &= \lambda \sum_i \alpha_i \bar{K}_{ki}, \end{aligned} \quad (12)$$

where

$$\bar{K}_{ki} = \bar{\Phi}(\mathbf{x}_k) \cdot \bar{\Phi}(\mathbf{x}_i). \quad (13)$$

Also, for the right side of the equation

$$\begin{aligned}
 & \bar{\Phi}(\mathbf{x}_k) \cdot C\mathbf{v} \\
 &= \bar{\Phi}(\mathbf{x}_k) \cdot \frac{1}{N} \sum_j \bar{\Phi}(\mathbf{x}_j) \bar{\Phi}(\mathbf{x}_j)^T \sum_i \alpha_i \bar{\Phi}(\mathbf{x}_i) \\
 &= \bar{\Phi}(\mathbf{x}_k) \cdot \frac{1}{N} \sum_i \alpha_i \left\{ \sum_j \bar{\Phi}(\mathbf{x}_j) \bar{\Phi}(\mathbf{x}_j)^T \bar{\Phi}(\mathbf{x}_i) \right\} \\
 &= \frac{1}{N} \sum_i \alpha_i \left[\bar{\Phi}(\mathbf{x}_k) \cdot \left\{ \sum_j \bar{\Phi}(\mathbf{x}_j) \bar{\Phi}(\mathbf{x}_j)^T \bar{\Phi}(\mathbf{x}_i) \right\} \right] \\
 &= \frac{1}{N} \sum_i \alpha_i \sum_j \{ \bar{\Phi}(\mathbf{x}_k) \cdot \bar{\Phi}(\mathbf{x}_j) \} \{ \bar{\Phi}(\mathbf{x}_j) \cdot \bar{\Phi}(\mathbf{x}_i) \} \\
 &= \frac{1}{N} \sum_i \alpha_i \sum_j \bar{K}_{kj} \bar{K}_{ji}. \tag{14}
 \end{aligned}$$

Thus we get

$$\begin{aligned}
 N\lambda\alpha &= \bar{\mathbf{K}}\alpha \\
 \hat{\lambda}\alpha &= \bar{\mathbf{K}}\alpha. \tag{15}
 \end{aligned}$$

Consequently, we only need to diagonalize $\bar{\mathbf{K}}$ which is computed as follows.

$$\begin{aligned}
 \bar{K}_{ij} &= \bar{\Phi}(\mathbf{x}_i) \cdot \bar{\Phi}(\mathbf{x}_j) \\
 &= \left(\Phi(\mathbf{x}_i) - \frac{1}{N} \sum_{m=1}^N \Phi(\mathbf{x}_m) \right) \\
 &\quad \cdot \left(\Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{n=1}^N \Phi(\mathbf{x}_n) \right) \\
 &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{m=1}^N \Phi(\mathbf{x}_m) \cdot \Phi(\mathbf{x}_j) \\
 &\quad - \frac{1}{N} \sum_{n=1}^N \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_n) \\
 &\quad + \frac{1}{N^2} \sum_{m,n=1}^N \Phi(\mathbf{x}_m) \cdot \Phi(\mathbf{x}_n) \\
 &= K_{ij} - \frac{1}{N} \sum_{m=1}^N 1_{im} K_{mj} - \frac{1}{N} \sum_{n=1}^N K_{in} 1_{nj} \\
 &\quad + \frac{1}{N^2} \sum_{m,n=1}^N 1_{im} K_{mn} 1_{nj} \tag{16}
 \end{aligned}$$

$$K_{ij} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{17}$$

$$1_{ij} = 1 \quad \text{for all } i, j \tag{18}$$

Using the $N \times N$ matrix $(\mathbf{1}_N)_{ij} := 1/N$, we get the more compact expression

$$\bar{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N \mathbf{K} - \mathbf{K} \mathbf{1}_N + \mathbf{1}_N \mathbf{K} \mathbf{1}_N. \tag{19}$$

We thus can compute $\bar{\mathbf{K}}$ from \mathbf{K} , and then solve the eigenvalue problem (15).

Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ denote the eigenvalues, and $\alpha^{(1)}, \dots, \alpha^{(N)}$ the corresponding complete set of eigenvectors, with λ_p being the first nonzero eigenvalue. We normalize

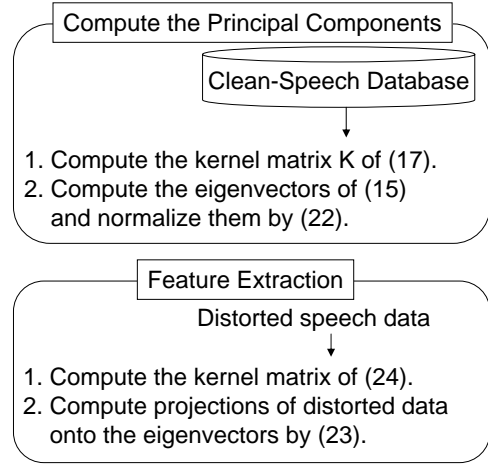


Figure 2. Procedure of feature extraction

$\alpha^{(p)}, \dots, \alpha^{(N)}$ by requiring that the corresponding vectors are normalized:

$$\mathbf{v}^{(l)} \cdot \mathbf{v}^{(l)} = 1, \quad \text{for all } l = p, \dots, N \tag{20}$$

From (11) and (15) we get

$$\begin{aligned}
 1 &= \sum_{i,j}^N \alpha_i^{(l)} \alpha_j^{(l)} (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \\
 &= \sum_{i,j}^N \alpha_i^{(l)} \alpha_j^{(l)} K_{ij} \\
 &= (\alpha^{(l)} \cdot \bar{\mathbf{K}} \alpha^{(l)}) \\
 &= \hat{\lambda}_l (\alpha^{(l)} \cdot \alpha^{(l)}). \tag{21}
 \end{aligned}$$

Therefore, we finally normalize α by

$$\hat{\alpha}^{(l)} = \frac{\alpha^{(l)}}{\sqrt{\hat{\lambda}_l}}. \tag{22}$$

Next, for feature extraction, we project test data \mathbf{y} onto eigenvectors $\mathbf{v}^{(l)}$ in the high-dimensional space.

$$\begin{aligned}
 (\mathbf{v}^{(l)} \cdot \bar{\Phi}(\mathbf{y})) &= \sum_{i=1}^N \hat{\alpha}_i^{(l)} (\bar{\Phi}(\mathbf{x}_i) \cdot \bar{\Phi}(\mathbf{y})) \\
 &= \sum_{i=1}^N \hat{\alpha}_i^{(l)} \bar{K}^{test}(\mathbf{x}_i, \mathbf{y}) \tag{23}
 \end{aligned}$$

Similar to (16), we can compute \bar{K}^{test} from K^{test} .

$$\begin{aligned}
 \bar{K}_{ij}^{test} &= \left(\Phi(\mathbf{y}_i) - \frac{1}{N} \sum_{m=1}^N \Phi(\mathbf{x}_m) \right) \\
 &\quad \cdot \left(\Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{n=1}^N \Phi(\mathbf{x}_n) \right) \tag{24}
 \end{aligned}$$

$$\bar{\mathbf{K}}^{test} = \mathbf{K}^{test} - \mathbf{1}'_N \mathbf{K} - \mathbf{K}^{test} \mathbf{1}_N + \mathbf{1}'_N \mathbf{K} \mathbf{1}_N \tag{25}$$

Here $\mathbf{1}'_N$ is the $L \times N$ matrix with all entries equal to $1/N$, and the total number of frames for the test data is L . The procedure of the feature extraction is summarized in Fig. 2.

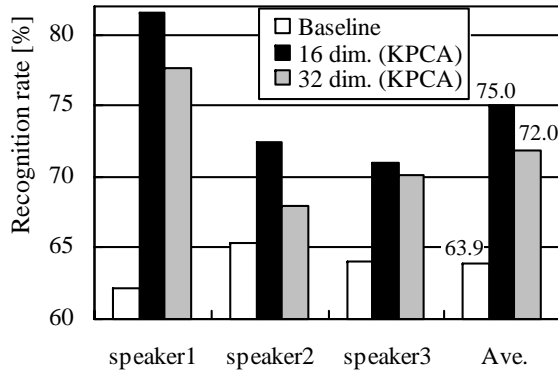


Figure 3. Recognition rates for the reverberant speech (reverberation time: 470 msec) by the proposed method ($p = 1$ in polynomial function)

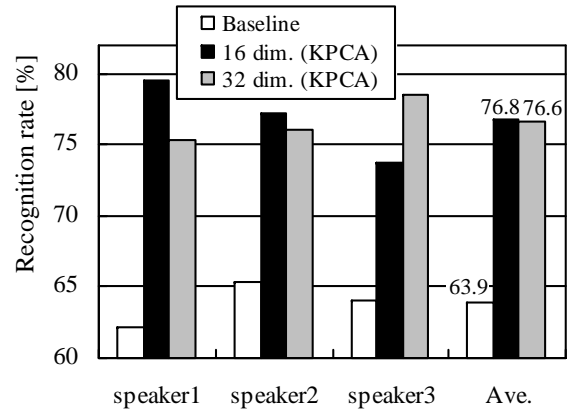


Figure 4. Recognition rates for the reverberant speech (reverberation time: 470 msec) by the proposed method ($p = 2$ in polynomial function)

III. RECOGNITION EXPERIMENT

A. Experimental Conditions

The new feature extraction method was evaluated on reverberant speech recognition tasks. Reverberant speech was simulated using a linear convolution of clean speech and impulse response. The impulse response was taken from the RWCP sound scene database [16]. The reverberation time was 470 msec. The distance to the microphone was about 2 meters, and the size of the recording room was about $6.7 \text{ m} \times 4.2 \text{ m}$ (width \times depth).

In order to compute the matrix, \mathbf{K} , it would be necessary to use all the training data, but it is not realistic in terms of the cost of the computation. Therefore, in this experiment, $N = 2,500$ frames were randomly picked from the training data, and we used the polynomial kernel function.

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (26)$$

The speech signal was sampled at 12 kHz and windowed with a 32-msec Hamming window every 8 msec. The models of 54 context-independent phonemes were trained by using 2,620 words in the ATR Japanese speech database for the speaker-dependent HMM. Each HMM has three states and three self-loops, and each state has four Gaussian mixture components. The tests were carried out on 1,000-word recognition tasks, and three males spoke the 1,000 words. The baseline recognition rate was 63.9%, where 16-order MFCCs and their delta coefficients were used as feature vectors.

B. Experimental Results

Figure 3 shows the recognition rates using kernel PCA ($p = 1$ in polynomial function). As can be seen from Fig. 3, the use of kernel PCA instead of DCT improves the recognition rates from 63.9% to 75.0%. Here, in the new feature extraction, kernel PCA was applied to 32-dimension mel-scale filter bank output, and then the delta coefficients were also computed. Figure 4 shows the recognition rates using kernel PCA ($p = 2$ in polynomial function). These results clearly show that the performance is better when using kernel PCA instead of DCT.

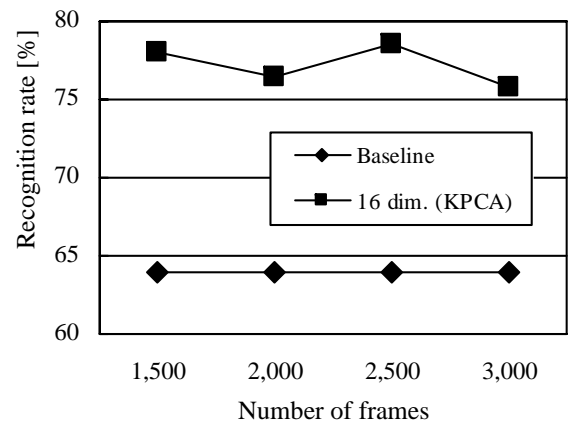


Figure 5. Recognition rates for test speaker3 when kernel PCA is applied using different amounts of training data

The kernel PCA for the polynomial function of $p = 1$ is almost same as the linear PCA. The recognition rate using the linear PCA described in Section II-A is actually 75% on average. Compared to Figure 3, the recognition rate is equal to that of the kernel PCA ($p = 1$).

Next, we applied kernel PCA to 16-order MFCCs [13] [14]. The recognition rate improved from 63.9% to 67.8%. As can be seen from Figure 4, a further improvement was obtained by the new method, where kernel PCA was applied to the mel-scale filter bank output. This is because we can expect that kernel PCA in the spectral domain will project the main speech element onto low-order features, while the reverberant elements will be projected onto high-order features.

Figure 5 shows the performance of test speaker3 when the kernel PCA is applied using different amounts of training data in (6). In this case, increasing the amount of training data does not significantly improve the performance of the kernel PCA. This result shows that the use of 2,500 frames of training data is suitable for this experiment.

Figure 6 shows the recognition rates for clean speech by the proposed method. The recognition rate with the new feature extraction was 97.6%, and the baseline performance using DCT was 97.3%. In clean environments, the

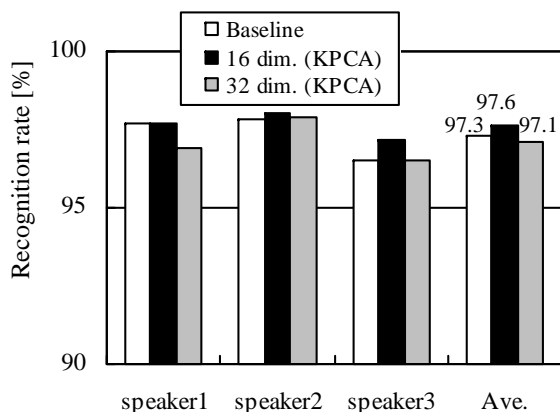


Figure 6. Recognition rates for the clean speech by the proposed method. ($p = 2$ in polynomial function)

experiment results indicate that the new method achieves almost the same performance as that of DCT.

Next, Table I shows the performance using the sigmoid kernel as shown in (27) instead of the polynomial kernel,

$$K(\mathbf{x}, \mathbf{y}) = \tanh(a\mathbf{x} \cdot \mathbf{y} - \sigma), \quad (27)$$

where $\sigma = 0.01$, and the recognition rates for test speaker3 are shown. The results in Table I show a decrease in recognition rate, compared to the polynomial kernel. Also, it is difficult to find two appropriate parameters, a and σ , in the sigmoid kernel.

Finally, we examined the performance for the kernel principal component based on the speaker-independent (SI) data instead of the speaker-dependent (SD) data. In this case, 2,500 frames from 25 males were used for calculation of $\bar{\mathbf{K}}$ in (15), and the acoustic model was trained using the SD data in order to examine only the accuracy of the PCA filter estimated by SI data. Table II shows the recognition rates for test speaker3 when the principal component is estimated by SI data. (*) shows the recognition rates for the speaker-dependent data. The recognition rate results in a 1.5% decrease on average because of increasing the speaker variability.

IV. SUMMARY

This paper has described a PCA-based speech enhancement technique for distorted speech recognition, where kernel PCA is applied to the mel-scale filter bank output. It can be expected that kernel PCA will project the main speech element onto low-order features, while the reverberant (noise) element will be projected onto high-order features, and the PCA-based filter will extract the feature of clean speech only. From our recognition results, it is shown that the use of kernel PCA instead of DCT provides better performance for reverberant speech (reverberation time: 470 msec).

REFERENCES

[1] S. Mika, B. Scholkopf, A.J. Smola, K.-R. Muller, M. Scholz, and G. Ratsch, "Kernel PCA and de-noising in feature spaces," In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pp. 536–542, MIT Press, 1999.

TABLE I.
RECOGNITION RATES [%] WITH THE SIGMOID FUNCTION

	16 dim.	24 dim.	32 dim.
a=0.0001	58.8	60.7	61.7
a=0.00005	71.6	69.7	68.3
a=0.00001	73.0	71.3	72.6
a=0.000005	71.6	72.7	73.4

TABLE II.
RECOGNITION RATES [%] WHEN THE KERNEL PRINCIPAL COMPONENT IS ESTIMATED BY SPEAKER-INDEPENDENT DATA

	16 dim.	24 dim.	32 dim.
$p = 1$	70.7 (71.0)	72.9 (74.0)	72.2 (70.1)
$p = 2$	72.0 (73.7)	73.7 (74.8)	74.4 (78.5)
$p = 3$	72.0 (75.6)	73.3 (74.1)	73.3 (76.1)

[2] H. Hermansky and N. Morgan, "RASTA Processing of Speech," *IEEE Trans. on Speech and Audio Processing*, Vol. 2, No. 4, pp. 578-589, 1994.

[3] C. Avendano, S. Tivrewala, and H. Hermansky, "Multiresolution channel normalization for ASR in reverberant environments," *Eurospeech*, pp. 1107-1110, 1997.

[4] U. H. Yapanel and J. H. L. Hansen, "A New Perspective on Feature Extraction for Robust In-Vehicle Speech Recognition," *Eurospeech*, pp. 1281-1284, 2003.

[5] B. J. Shannon and K. K. Paliwal, "Influence of Autocorrelation Lag Ranges on Robust Speech Recognition," *ICASSP*, pp. 545-548, 2005.

[6] W. Li, K. Itou, K. Takeda and F. Itakura, "Two-Stage Noise Spectra Estimation and Regression Based In-Car Speech Recognition Using Single Distant Microphone," *ICASSP*, pp. 533-536, 2005.

[7] M. Fujimoto, S. Nakamura, "Particle Filter Based Non-Stationary Noise Tracking for Robust Speech Recognition," *ICASSP*, pp. 257-260, 2005.

[8] K. Kinoshita, T. Nakatani and M. Miyoshi, "Efficient Blind Dereverberation Framework for Automatic Speech Recognition," *Interspeech*, pp. 3145-3148, 2005.

[9] M. Tokuhira and Y. Ariki, "Effectiveness of KL-Transformation in Spectral Delta Expansion," *Eurospeech99*, pp. 359-362, 1999.

[10] R. Vetter, N. Virag, P. Renevey and J.-M. Vesin, "Single Channel Speech Enhancement Using Principal Component Analysis and MDL Subspace Selection," *Eurospeech*, 1999.

[11] S.-M. Lee, S.-H. Fang, J.-W. Hung and L.-S. Lee, "Improved MFCC Feature Extraction by PCA-Optimized Filter Bank for Speech Recognition," *Automatic Speech Recognition and Understanding*, 2001, ASRU, pp. 49-52, 2001.

[12] F. Asano, Y. Motomura, H. Asoh and T. Matsui, "Effect of PCA Filter in Blind Source Separation," *Proc. ICA2000*, pp. 57-62, 2000.

[13] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "On the Use of Kernel PCA for Feature Extraction in Speech Recognition," *IEICE Trans. Inf. & Syst.*, Vol. E87-D, No. 12, pp. 2802-2811, 2004.

[14] A. Lima, H. Zen, Y. Nankaku, K. Tokuda, T. Kitamura and F. G. Resende, "Applying Sparse KPCA for Feature Extraction in Speech Recognition," *IEICE Trans. Inf. & Syst.*, Vol. E88-D, No. 3, pp. 401-409, 2005.

[15] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear

component analysis as a kernel eigenvalue problem," *Neural Computation*, Vol. 10, pp. 1299-1319, 1998.

- [16] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, T. Yamada, "Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition," *Proceedings of International Conference on Language Resources and Evaluation*, Vol. 2, pp. 965-968, 2000.

Tetsuya Takiguchi received the B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and the M.E. and Dr. Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively. From 1999 to 2004, he was a researcher at IBM Research, Tokyo Research Laboratory, Kanagawa, Japan. He is currently a Lecturer with Kobe University. His research interests include robust speech recognition, signal processing, and microphone arrays. He received the Awaya Award from the Acoustical Society of Japan in 2002. He is a member of the IEEE, the Information Processing Society of Japan, and the Acoustical Society of Japan.

Yasuo Ariki received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976 and 1979, respectively. He was an assistant professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an associate professor and from 1992 to 2003 a professor at Ryukoku University. Since 2003 he has been a professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database. He is a member of IEEE, IPSJ, JSAI, ITE and IIEEJ.