Single Channel Dereverberation Method in Log-Melspectral Domain Using Limited Stereo Data for Distant Speaker Identification

Aditya Arie Nugraha*, Kazumasa Yamamoto*†, Seiichi Nakagawa*
*Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Japan
†Department of Information and Computer Engineering, Toyota National College of Technology, Toyota, Japan
E-mail: {arie, kyama, nakagawa}@slp.cs.tut.ac.jp

***Abstract*—In this paper, we present a feature enhancement method that uses neural networks (NNs) to map the reverberant feature in a log-melspectral domain to its corresponding anechoic feature. The mapping is done by cascade NNs trained using Cascade2 algorithm with an implementation of segment-based normalization. We assumed that the dimensions of feature were independent from each other and experimented on several assumptions of the room transfer function for each dimension. Speaker identification system was used to evaluate the method. Using limited stereo data, we could improve the identification rate for simulated and real datasets. On the simulated dataset, we could show that the proposed method is effective for both noiseless and noisy reverberant environments, with various noise and reverberation characteristics. On the real dataset, we could show that by using 6 independent NNs configuration for 24-dimensional feature and only 1 pair of utterances we could get 35% average error reduction relative to the baseline, which employed cepstral mean normalization (CMN).**

I. Introduction

The use of a distant-talking microphone for automatic speech/speaker recognition (ASR) system can improve user convenience. However, the use of reverberant signal captured by the microphone may degrade the system performance.

Several feature enhancement approaches have been proposed to deal with the reverberation problem; vector Taylor series (VTS) [1], particle filter [2], Kalman filter [3], and so on. Several methods assume that stereo *training* data can be acquired. In the context of distant speaker identification, stereo data are simultaneously recorded pairs of close-talking and distant-talking utterances. In [4], 13 multilayer perceptron (MLP) NNs were trained using stereo data to map the 13-dimensional reverberant cepstral feature, where one NN was used for one dimension of feature, to its corresponding anechoic feature. The input of each NN was a sequence of cepstral feature coefficients from 9 consecutive frames and the output was a cepstral feature coefficient. For the noise problem, SPLICE is a feature enhancement approach which also needs stereo data [5]. It estimates the clean cepstral feature from the noisy feature using a Gaussian Mixture Model (GMM) of noisy feature.

Several algorithms for distant text-independent speaker identification have been proposed, e.g. GMM, GMM-Universal Background Model (GMM-UBM), Support Vector Machine (SVM) [6]. Several more robust features also have been proposed, e.g. modulation spectral features [7] and short segment cepstral coefficient (SSCC) [8].

In [9], we introduced a single channel non-linear regression based dereverberation method using a single NN for distant speaker identification. The NN was trained on stereo data to compensate the reverberation effect by mapping the reverberant feature in a log-melspectral domain to its corresponding anechoic feature. The log-melspectral domain was used because it gave us a compressed representation of mel-filterbank output, which was beneficial for the NN. According to [10], several feature enhancement approaches work better in the log-spectral domain than in the power spectral domain. The log-spectral domain has also a linear relation to the cepstral domain, which is the final feature in many ASR system.

We use cascade NNs trained using Cascade2 algorithm, which is a variation of Cascade-Correlation (CasCor) algorithm [11]. Comparing to MLP, CasCor family does not have the issue of deciding the number of layers and hidden neurons to use in NN before the training. Cascade2 is used because it uses error minimization instead of covariance maximization, so it is suitable for our regression task.

In this paper, we extend the method to the use of multiple NNs by modifying our assumptions about the room transfer function for each dimension of log-melspectral feature. We also show how the difference of assumptions affects the performance of distant speaker identification system, which used MFCC-based speaker-specific GMMs as the speaker models [12], by using limited stereo data. We believe that the use of CasCor and the possibility of using limited stereo data increase the feasibility of our method.

II. Reverberation Model

The relation between anechoic and reverberant signal in log-melspectral domain should be represented as a non-linear model [3]. However, for simplicity, we defined it as

$$y_j(t) = \alpha_{j,0} s_j(t) + \sum_{n=1}^{N} \alpha_{j,i} s_j(t-n), \tag{1}$$

where $s_j(t)$ and $y_j(t)$ represent the log-melspectral coefficients of anechoic and reverberant signal, respectively, for feature dimension j and frame index t [13]. While, $\alpha_{j,0}, \alpha_{j,1}, \ldots, \alpha_{j,N}$ represent the room transfer function (RTF) for feature dimension j.

Then, the estimated anechoic coefficient $\hat{s}_j(t)$ could be expressed as

$$\hat{s}_j(t) = \beta_{j,0} y_j(t) + \sum_{k=1}^{L} \beta_{j,k} y_j(t-k) + \varepsilon \approx \sum_{k=0}^{L} \beta_{j,k} y_j(t-k), \tag{2}$$

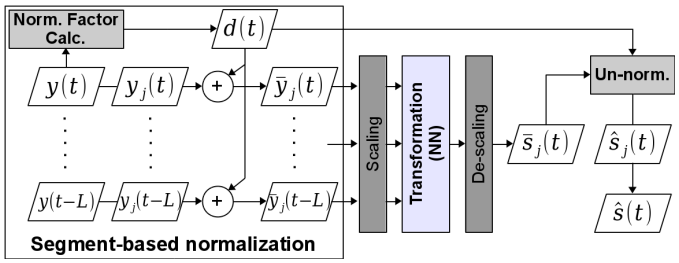

Fig. 1. *The proposed method.*

where $\beta_{j,0}, \beta_{j,1}, \ldots, \beta_{j,L}$ denotes the weights which are used to compensate the RTF and L denotes the number of past frames in the segment.

As can be interpreted from Eq. (2), we assumed that the dimensions of feature are independent from each other and certain dimension of reverberant coefficient is only affected by the same dimension of past and current coefficients. We also assumed that the RTFs for every dimension are different. It is an extension of the model we used in [9], where we assumed that the RTFs for all dimensions are the same.

III. Dereverberation Method

In matrix form, Eq. (2) could be written as $\hat{S} = BY$, where $\hat{S}$ denotes the estimated anechoic feature vector, Y denotes the supervector which consists of reverberant feature vectors, and B denotes the transformation matrix which represents the RTF compensation. Then, a regression using NN is done to determine the function B such that $\mathrm{argmin}_B \|\hat{S} - (B \otimes Y)\|^2$, where $\otimes$ denotes a non-linear transformation.

Fig. 1 shows the block diagram of our proposed method. The inputs are $y(t), y(t-1), \ldots, y(t-L)$, which are current and past reverberant log-melspectral coefficient vectors. The output is $\hat{s}(t)$, which is the estimated current anechoic log-melspectral coefficient vector.

Segment-based normalization (Eq. (4)) is employed to deal with the power difference between the anechoic speech signal and the reverberant signal captured by a distant-talking microphone. In the NN training stage, it is done by normalizing the current reverberant feature vector and the current anechoic feature vector (which is the target of training) to the normalization target. Besides, it is employed to preserve the power envelope of NN input segment, by normalizing the past frames relative to the current frame.

$$d(t) = \gamma - \frac{1}{D}\sum_{j=1}^{D} y_j(t), \tag{3}$$

$$\bar{y}_j(t-k) = y_j(t-k) + d(t), \quad \begin{array}{l}\text{for } j = 1, 2, \ldots, D, \\ \text{for } k = 0, 1, \ldots, L,\end{array} \tag{4}$$

where $d(t)$ is the normalization factor, γ is the normalization target, D is the number of feature dimensions, and $\bar{y}_j(t)$ is the normalized log-melspectral coefficient for feature dimension j and frame index t.

The mean of NN output $\bar{s}(t)$ should be equal to the normalization target, because we also normalize the target of NN training. Therefore, we use un-normalization (Eq. (5)) to return it to its original mean of power. We showed how the un-normalization improved both non-speech and speech parts of utterance in [9].

$$\hat{s}_j(t) = \bar{s}_j(t) - d(t) \tag{5}$$

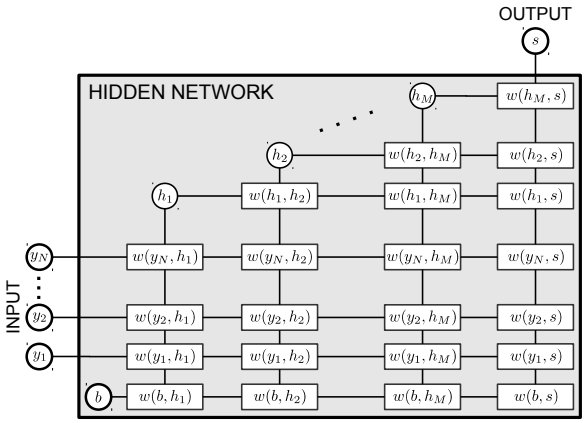

Fig. 2. *The cascade NN architecture used in this work.*

In general, the scaling and de-scaling can be regarded as the pre-processing and post-processing, respectively, of NN. The scaling is done so that the NN input and output have value range of (-1, 1). In contrast, the de-scaling is used to recover the log-melspectral coefficient value from its scaled value.

Fig. 2 shows an illustration of NN we used in our works, with N input neurons, M hidden neurons/layers, and 1 output neuron. We use the implementation of Cascade2 algorithm with RPROP weight update algorithm in FANN library[1] [14]. A linear activation function is used for the output neuron, while the hidden neurons use a symmetric sigmoid (tanh) function. We set the maximum number of hidden neurons to be equal to twice of the input neurons ($M \leq 2N$). Therefore, the NN is quite compact, e.g. for 8-frame frame selection, we use 8 input, 16 hidden, and 1 output neurons.

Beside 'linear' frame selection represented by Eq. (2), we also defined 'skip1' selection by using $y_j(t-2k)$ instead of $y_j(t-k)$. The use of 'skip1' frame selection can be regarded as dimensionality reduction by minimizing the redundant parts caused by the windowing. Therefore, we could get a representation of longer context using smaller number of frames, which is beneficial for the NN training.

IV. Experiments and Discussion

For evaluation, we implemented the proposed method into a speaker identification system. The dereverberation was done on a 24-dimensional log-melspectral feature domain. The dereverberation result was transformed to 12-dimensional melcepstral feature, normalized using CMN, and then used as the input of identification system. The system used speaker-specific GMMs as the speaker models, in which each speaker was modelled using a 32-mixture GMM. Voice activity detection (VAD) was also employed to remove the silence parts in the beginning and ending of recordings. All duration information written below excluded these silence parts. For the feature extraction, 25 ms Hamming window with 10 ms shift is used.

Using the simulated data experiment results, we show the effect of frame selection type and number of training data to the system performance for reverberant and noisy reverberant data. Using the real data experiment results, we show the effect of RTF assumption, which is related to the number of NN.

[1] http://leenissen.dk/fann

A. Simulated Data

The speech data of 100 male speakers from JNAS database [15] were used in the experiments. In average, each speaker has 105 utterances. The room impulse response (RIR) and noise data were taken from Aurora-5[2], while the simulation program was SImulation of REal Acoustics (SIREAC)[3] [16].

The GMMs for the speaker identification system was trained using randomly selected 500 clean utterances (100 speakers; 5 utterances for each speaker). The average duration of the utterances was about 3 seconds. CMN and VAD was employed as pre-processing of GMM training data.

A pool of training data was created for each type of simulated (noisy) reverberant data. This pool of data consisted of randomly selected 25 pairs of clean utterance and simulated (noisy) reverberant utterance. The average duration of the utterances was about 7 seconds. From this pool of training data, 1-utterance (”1u”), 5-utterance (”5u”), 10-utterance (”10u”), and 15-utterance (”15u”) NN training datasets were created.

A testing dataset was created also for each type of simulated (noisy) reverberant data. Each dataset consisted of randomly selected 1000 simulated (noisy) reverberant utterances (100 speakers; 10 utterances for each speaker). The average duration of the utterances was about 5 seconds.

The *baseline* for each type of simulated (noisy) reverberant data is shown in Table I. The identification rate for the clean version of testing dataset was *98.0%*. Note that we can regard this baseline as the result of enhancement using CMN.

Table II and Table III are the result of experiments using the model we proposed in this paper, where we used 1 NN for each dimension of feature, in total 24 NNs. The italic text in the table means that it is better than the baseline. The bold text represent the best rate for certain RIR characteristic (type, T60, noise) and amount of training data. Overall, by using our proposed method, we could improve the speaker identification rate for the reverberant and noisy reverberant data. For small T60, we could get improvement using short context (4-frame linear) and only 1 pair of utterances. For bigger T60, it is necessary to use longer context and also use more training data (pairs of utterancess). Therefore, we prefer using 8 frames as the NN input so that the method is still feasible when there are only limited stereo data available. The results show that skip1 frame selection tends to give better improvement than linear frame selection, especially for the reverberant data. While for the noisy reverberant, linear frame selection showed better performance. The average error reduction rate (ERR) for 8-frame skip1 frame selection trained using 5 pairs of utterances was 20.4%, while for 8-frame linear was 19.2%.

B. Real Data

All speech data used in the experiments were recorded in a recording room whose dimensions were about 5 x 6.4 x 2.65 m. There was no material that was intentionally installed to reduce reverberation or noise, except the materials of microphone arrays that were used to place the microphones. The reverberation time and background noise were approximately 330 ms and 35 dBA, respectively, measured from the middle of the room. The recording process was done using a 32-channel recording system with 16 kHz sampling rate. A more detailed explanation about the room can be found in [13].

In the experiments, we used two different datasets which were also used in [9]. Both datasets were recorded in the recording room as described above. These datasets consisted of close-talking utterances, which were recorded from the distance of 25 cm, and distant-talking utterances, which uttered from five different positions (P01-P05) and captured by eight microphone arrays. However, in the experiments, we only considered utterances from P01, P02, and P05, that was captured by the first microphone of microphone array A (hereafter, referred as A1). P01, P02, P05, and A1 were in-line and the speakers' utterances were directed to A1. The distance between P01-A1, P02-A1 and P05-A1 were about 4.1 m, 2.6 m and 1.6 m, respectively. Theoretically, the Direct-to-Reverberant Ratios (DRRs) of these distant-talking recordings were small because the distances were greater than the critical distance for the room, which is around 0.9 m.

The first dataset was a stereo dataset, which contained one session of three speakers' recordings where each speaker uttered 10 utterances from each position. Although they were recorded from the distance of 25 cm, we regarded the close-talking utterances as our clean speech signals and tried to map the distant-talking utterances to close-talking utterances using NN. The NN training datasets were created for each position and consisted of the first 1 or 5 pairs of utterances from each speaker. The average duration of the training utterances was about 6.3 seconds. We defined three kinds of dataset, i.e. 1-utterance (”1u”; 1 pair of utterances), 1-speaker (”1s”; 5 pairs), and 3-speaker (”3s”; 15 pairs).

The second dataset was a non-stereo dataset, which contained two sessions of 20 speakers' recordings where each speaker uttered 10 utterances from each position in each session. The GMMs for the speaker identification system was trained using 10 close-talking utterances for every speaker (2 sessions; the first 5 utterances from each session). The testing dataset consisted of 200 distant-talking utterances (20 speakers; the second 5 utterances from each session for each speaker) for each position.

The *baseline* for the position P01, P02, and P05 are *84.0%*, *93.0%*, and *91.5%*, respectively. In addition, the identification rate for the close-talking version of testing dataset was *99.5%*.

Similar to the experiment results using the simulated data, the multiple NNs showed better performance for the big datasets (”3s”), but worse for the small datasets (”1s”, ”1u”). Therefore, we tried to look for a good performance trade-off for all datasets by using modified multiple NNs configuration, where 1 NN is used for more than 1 dimension of feature. In our experiments, the number of dimension in each NN was divided evenly. For example, in 6 NNs configuration, the 1st NN is for the dimension 1–4 of 24-dimensional log-melspectral feature, 2nd NN is for 5–8, and so on. It means that we assumed that the RTF for the dimension 1–4 is the same, the RTF for the dimension 5–8 is the same, and so on.

Table IV shows our best experiment results among the frame selection types. ”1 NN” represents the single NN configuration [9] and ”24 NNs” represents the original multiple NNs configuration. We found that the 6 NNs configuration gave us the best trade-off. Comparing to 1 NN, it improved the performance of ”1s” and ”1u” datasets. Surprisingly,

[2]http://aurora.hsnr.de/aurora-5.html

[3]http://dnt.kr.hs-niederrhein.de/wwwsim

TABLE I

The baseline for experiments using simulated data.

RIR		Speaker Identification Rate (%)		
Type	T60 (ms)	Reverb.	Noisy Reverberant	
			SNR=20dB	SNR=10dB
office	300	81.0	60.3	23.4
	400	74.3	55.3	20.6
livingroom	400	70.9	N/A	N/A
	500	62.9	N/A	N/A

TABLE II

The results of experiments using simulated reverberant data on multiple neural networks configuration.

RIR		Frame Selection		Spk. Identification Rate (%)			
		Type	Num.	1u	5u	10u	15u
office	300 ms	linear	4	*83.1*	*85.2*	*84.9*	*86.2*
			8	74.9	*86.4*	*88.2*	*88.4*
			16	44.0	77.1	*84.0*	*86.5*
		skip1	4	***86.0***	*86.8*	*86.5*	*87.0*
			8	70.1	***87.1***	***88.8***	***88.8***
	400 ms	linear	4	73.9	*77.4*	*78.0*	*77.5*
			8	63.6	*80.3*	*81.9*	*82.3*
			16	35.7	68.7	*79.7*	*81.3*
		skip1	4	***78.0***	*80.0*	*79.7*	*80.2*
			8	63.6	***81.6***	***82.8***	***84.3***
livingroom	400 ms	linear	4	64.4	*71.1*	70.8	*71.5*
			8	56.1	***77.0***	***79.4***	*79.8*
			16	27.6	61.3	*73.2*	*76.7*
		skip1	4	**68.2**	*74.6*	*74.4*	*74.8*
			8	44.2	*75.7*	***79.4***	***80.3***
	500 ms	linear	4	54.9	60.9	61.8	61.2
			8	45.4	*68.2*	*70.8*	*71.0*
			16	20.1	54.1	*67.4*	*72.8*
		skip1	4	**60.2**	*68.1*	*65.9*	*68.8*
			8	35.5	***70.3***	***73.4***	***74.9***

comparing to 24 NNs, it also improved the performance of "3s" dataset. The average ERR for "3s", "1s", and "1u" were 50.9%, 45.6%, and 35.0%, respectively. In addition, we could get better performance for "3s" dataset by using 12 NNs, where its average ERR was 53.9%.

For limited number of stereo data, the 6 NNs and 12 NNs configurations could perform better than 24 NNs because they had more training data for each NN, e.g. the NNs in 24 NNs were only trained using the data from one dimension, but the NNs in 6 NNs were trained using the data from four dimensions. It will not result the best RTF compensation for a certain dimension, but it might prevent the overfitting problem caused by the lack of training data. Therefore, if there are enough stereo data available, 24 NNs is a very reasonable choice because the RTFs, which are represented by the NNs, should be frequency-dependent. That is also why the use of 1 NN is not good enough.

V. Conclusions

We presented a feature enhancement method that used cascade NNs to map the reverberant feature in the log-melspectral domain to its corresponding anechoic feature. Although the modelling aimed to suppress the reverberation effect, the method was also effective to be used in a noisy reverberant environment. By using only limited stereo training data, the method could improve the speaker identification rate. Using the real reverberant dataset, we could get 35% average ERR, relative to the baseline (CMN), for 20 speakers by using 6 NNs configuration and only 1 pair of utterances. Our future work will focus on omitting the need of stereo data and developing an unsupervised approach.

TABLE III

The results of experiments using simulated "office" noisy reverberant data on multiple neural networks.

T60	SNR	Frame Selection		Spk. Identification Rate (%)			
		Type	Num.	1u	5u	10u	15u
300 ms	20dB	linear	8	**48.3**	*66.8*	***69.4***	***70.0***
		skip1	8	41.6	***67.3***	*67.7*	*69.4*
	10dB	linear	8	**17.4**	*36.2*	***40.7***	***41.5***
		skip1	8	15.1	***37.7***	*38.9*	***41.5***
400 ms	20dB	linear	8	**46.8**	***61.9***	***64.2***	***65.7***
		skip1	8	33.6	*61.6*	*62.0*	*63.9*
	10dB	linear	8	**20.3**	***35.7***	***39.0***	***40.5***
		skip1	8	16.1	*33.1*	*37.8*	*39.0*

TABLE IV

The speaker identification rate of experiments using real data.

NN Conf.	Dataset	Spk. Identification Rate (%)			
		P01	P02	P05	Avg.
1 NN	3s	90.0	95.3	93.5	92.9
	1s	89.6	95.0	93.5	92.7
	1u	89.1	94.4	93.3	92.3
6 NNs	3s	92.3	**98.0**	94.0	94.8
	1s	**91.7**	**97.4**	93.7	94.3
	1u	**90.5**	**96.1**	93.2	93.3
12 NNs	3s	**92.8**	**98.0**	**94.5**	95.1
	1s	90.7	96.9	**94.4**	94.0
	1u	89.3	95.4	**93.4**	92.7
24 NNs	3s	92.2	96.7	94.0	94.3
	1s	88.8	95.2	93.4	92.5
	1u	86.9	93.6	92.4	91.0

References

[1] P. Moreno, B. Raj, and R. Stern, "A vector Taylor series approach for environment-independent speech recognition", in *Proc. of the IEEE ICASSP*, vol. 2, pp. 733–736, 1996.

[2] M. Wölfel, "Enhanced Speech Features by Single-Channel Joint Compensation of Noise and Reverberation", *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 17(2), pp. 312–323, 2009.

[3] A. Krueger and R. Haeb-Umbach, "Model-Based Feature Enhancement for Reverberant Speech Recognition", *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 18(7), pp. 1692–1707, 2010.

[4] Y. Pan and A. Waibel, "The effects of room acoustics on MFCC speech parameter", *Proc. of the ISCA INTERSPEECH*, pp. 129–132, 2000.

[5] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data", in *Proc. of the IEEE ICASSP*, pp. 301–304, 2001.

[6] H. Tang, Z. Chen, and T. S. Huang, "Comparison of Algorithms for Speaker Identification under Adverse Far-Field Recording Conditions with Extremely Short Utterances", in *Proc. of the IEEE ICNSC*, pp. 796–801, 2008.

[7] T. H. Falk and W.-Y. Chan, "Modulation spectral features for robust far-field speaker identification", *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 18(1), pp. 90–100, 2010.

[8] B. Avinash, S. Guruprasad, and B. Yegnanarayana, "Exploring subsegmental and suprasegmental features for a text-dependent speaker verification in distant speech signals", *Proc. of the ISCA INTERSPEECH*, pp. 1073–1076, 2010.

[9] A. A. Nugraha and S. Nakagawa, "Improving distant speaker identification robustness using a non-linear regression based dereverberation method in feature domain", in *Proc. of the Autumn Meeting of Acoust. Soc. of Japan*, pp. 163–166, 2012.

[10] M. Wölfel and J. McDonough, *Distant speech recognition*, John Wiley & Sons, 2009.

[11] S. Nissen, "Large Scale Reinforcement Learning using Q-SARSA(λ) and Cascading Neural Networks", M.Sc. Thesis, University of Copenhagen, Denmark, 2007.

[12] K. P. Markov and S. Nakagawa, "Text-independent speaker recognition using non-linear frame likelihood transformation", *Speech Communication*, vol. 24(3), pp. 193–209, 1998.

[13] K. Shimada, K. Yamamoto, and S. Nakagawa, "Consideration of robust speaker recognition for reverberation in distant speech", in *Proc. of the Spring Meeting of Acoust. Soc. of Japan*, pp. 199–202, 2012 (in Japanese).

[14] S. Nissen, "Implementation of a Fast Artificial Neural Network library (FANN)", Tech. Rep., University of Copenhagen, Denmark, 2007.

[15] K. Itou, M. Yamamoto, and K. Takeda, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research", *J. of the Acoust. Soc. of Japan*, vol. 20(3), pp. 199–206, 1999.

[16] H. G. Hirsch and H. Finster, "The Simulation of Realistic Acoustic Input Scenarios for Speech Recognition Systems", *Proc. of the ISCA INTERSPEECH*, pp. 2697–2700, 2005.