Noise-Robust Speech Features Based on Cepstral Time Coefficients

Ja-Zang Yeh
Department of Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan
ycc97m@cse.nsysu.edu.tw
Chia-Ping Chen
Department of Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan
cpchen@cse.nsysu.edu.tw

Abstract

In this paper, we investigate the noise-robustness of features based on the **cepstral time coefficients** (CTC). By cepstral time coefficients, we mean the coefficients obtained from applying the discrete cosine transform to the commonly used mel-frequency cepstral coefficients (MFCC). Furthermore, we apply temporal filters used for computing delta and acceleration dynamic features to the CTC, resulting in delta and acceleration features in the frequency domain. We experiment with five different variations of such CTC-based features. The evaluation is done on the Aurora 3 noisy digit recognition tasks with four different languages. The results show all but one such feature set performance gain, the other feature sets actually lead to performance gains. The best feature set achieves an improvement of 25% over the baseline feature set of MFCC.

Keywords: **MFCC, CTC, delta, robust feature**

1. Introduction

A front-end of a speech recognition system may consist of several stages for noise-robustness to achieve good performance. In the early stage of spectral domain, well-known methods such as spectral subtraction [1] and Wiener filter [2] may be applied. In the middle stage of cepstral domain, the mel-frequency cepstral coefficients (MFCC) are commonly used as the static feature set. In the post-processing stage, there may be normalization, temporal information integration, and transformation modules.

It has been observed that simple normalization approaches, such as the cepstral mean subtraction (CMS) [3], cepstral variance normalization (CVN) [4], and histogram normalization (HEQ) [5] can lead to significant performance improvement in recognition accuracy in noisy environment. Apparently such methods are capable of alleviating the *mismatch* between the clean and noisy data.

In this paper we investigate novel features based on simple transformation methods. Specifically, we insert a window of static cepstral vectors in a matrix and then apply the *discrete cosine transform* (DCT) along the temporal axis. The coefficents after the DCT is called the cepstral time coefficients,

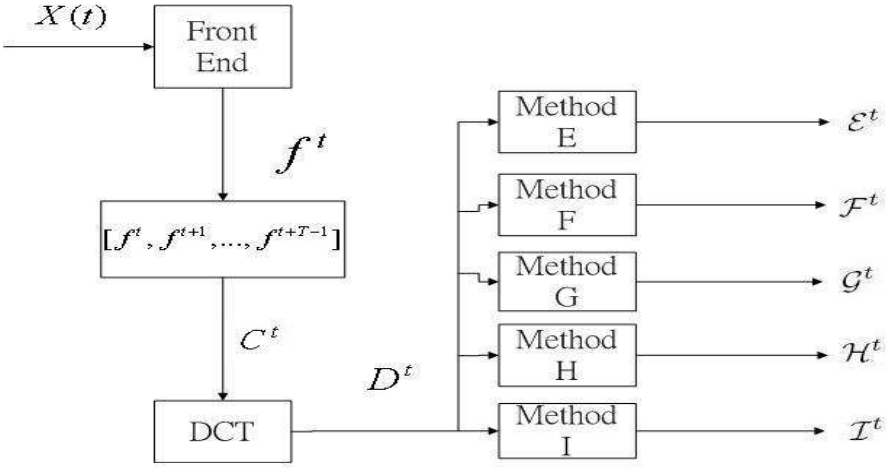


Figure 1: The block diagram of the proposed feature transformation methods.

and the resultant matrix is called the cepstral time matrix (CTM) [6, 7]. After CTM for each frame is extracted, we further apply normalization and routines for delta and acceleration feature extraction to the cepstral time coefficients. The transformed features are combined with the static MFCC features to form the final feature vector.

This paper is organized as follows. Section 2 defines the cepstral time matrix and introduces the investigated feature transformations. The experimental setup and recognition results are described in Section 3. In Section 4, we draw conclusions.

2. Feature Transformations

Our feature extraction and transformation process is illustrated in Figure 1. We begin with a review of the cepstral time matrix, which is followed by the mathematical definition of the proposed additive transformation methods.

2.1. Cepstral Time Coefficients

We first insert a fixed number of adjacent feature vectors in a matrix

$$C^t \triangleq \begin{bmatrix} C^t_{11} & C^t_{12} & \ldots & C^t_{1T} \\ \vdots & \ddots & & \vdots \\ C^t_{K1} & C^t_{K2} & \ldots & C^t_{KT} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{f}^t & \mathbf{f}^{t+1} & \ldots & \mathbf{f}^{t+T-1} \end{bmatrix}. \tag{1}$$

Here K is the feature vector dimension, and $\mathbf{f}^t$ is the feature vector of frame t, C^t is the matrix whose column vectors are the T consecutive feature vectors starting from frame t.

The cepstral time matrix at frame t, D^t, is related to C^t by the discrete-cosine transform. Each **row** of D^t is the discrete-cosine transform of the corresponding row of C^t. That is,

$$D^t_{i:} = DCT(C^t_{i:}). \tag{2}$$

Here $D^t_{i:}$ is the i-th row of matrix D.[1] We call D^t_{in} the nth cepstral time coefficient (CTC) of channel i at frame t. D is also called cepstral time matrix (CTM). It represents the spectral information of cepstral coefficient in an analysis window of frames. [1]Since our matrix index starts from 1 instead of 0, here the DCT needs to be

$$D_{in}^t = \sum_{\tau=1}^{T} C_{i\tau}^t \cos\left(\frac{(2\tau-1)(n-1)\pi}{2T}\right). \tag{3}$$

2.2. *CTC-Based Features*

In this paper, we have 5 different transforms applied to CTC, each leading to a different feature vector.

2.2.1. *Method E*

The first transform is dividing the first column of D^t by the number of frames (T), while leaving other columns unchanged. Let E^t be the new feature matrix, we have

$$\begin{cases} E_{:1}^t & = D_{:1}^t/T \\ E_{:n}^t & = D_{:n}^t, \;\; n \neq 1 \end{cases} \tag{4}$$

Note $E_{:1}^t$ has a physical meaning. According to (2), it is the mean of the cepstral coefficients within an analysis window (while $D_{:1}^t$ is the sum).

We then compute a novel feature set based on E^t. Specifically, we treat the columns in E^t as a temporal sequence and apply the delta and acceleration feature extraction steps. That is,

$$\begin{cases} \breve{E}_{:2}^t & = E_{:2}^t - E_{:1}^t \\ \breve{E}_{:3}^t & = E_{:3}^t - 2E_{:2}^t + E_{:1}^t. \end{cases} \tag{5}$$

We add the $\breve{E}_{:2}^{(t)}$ and $\breve{E}_{:3}^{(t)}$ to the static MFCCs, resulting in a feature vector of

$$\mathcal{E}^t = \begin{bmatrix} C_{:1}^t \\ \breve{E}_{:2}^t \\ \breve{E}_{:3}^t \end{bmatrix}. \tag{6}$$

2.2.2. *Method F*

An alternative transform is to normalize the feature values in the first column to the range of $[-1, 1]$. This is achieved by dividing $D_{:1}^t$ by the maximum magnitude of the first column. Let F^t be defined by

$$\begin{cases} F_{:1}^t & = D_{:1}^t/N^t \\ F_{:n}^t & = D_{:n}^t, \;\; n \neq 1 \end{cases} \tag{7}$$

where N^t is the maximum magnitude in the first column, i.e.,

$$N^t = \max_d |D_{d1}^t|.$$

The remaining operations are similar to Method E. That is,

$$\begin{cases} \breve{F}_{:2}^t & = F_{:2}^t - F_{:1}^t \\ \breve{F}_{:3}^t & = F_{:3}^t - 2F_{:2}^t + F_{:1}^t. \end{cases} \tag{8}$$

[1] In general, we will use notation $A_{i:}$ to denote the i-th row vector and $A_{:j}$ to denote the j-th column vector, of matrix A.

We add $\breve{F}_{:2}^{(t)}$ and $\breve{F}_{:3}^{(t)}$ to the static MFCCs, resulting in a feature vector of

$$\mathcal{F}^t = \begin{bmatrix} C_{:1}^t \\ \breve{F}_{:2}^t \\ \breve{F}_{:3}^t \end{bmatrix}. \tag{9}$$

2.2.3. Method G

In Method G, we add the first and second columns of CTM, which represents the zeroth and first cepstral time coefficients, to the static MFCC vector,

$$\mathcal{G}^t = \begin{bmatrix} C_{:1}^t \\ D_{:1}^t \\ D_{:2}^t \end{bmatrix}. \tag{10}$$

2.2.4. Method H

In Method H, we add the second and third columns of CTM, which represent the first and second cepstral time coefficients, to the static MFCC vector,

$$\mathcal{H}^t = \begin{bmatrix} C_{:1}^t \\ D_{:2}^t \\ D_{:3}^t \end{bmatrix}. \tag{11}$$

2.2.5. Method I

In Method I, we no longer use the MFCC. Instead, we simply use the zeroth, first, and second cepstral time coefficients,

$$\mathcal{I}^t = \begin{bmatrix} D_{:1}^t \\ D_{:2}^t \\ D_{:3}^t \end{bmatrix}. \tag{12}$$

2.2.6. Method B

For completeness, we describe our baseline features as Method B. Our baseline simply uses the 12 MFCCs ($c_1, \ldots, c_{12}$), the log energy, and the delta and delta-delta features. Therefore, the feature vector has a dimension of 39, which agrees with other methods. Furthermore, our baseline results agree with the Aurora 3 baseline results [8, 9].

3. Experiments

3.1. Experimental Database

We evaluate the proposed CTC-based speech features on the Aurora 3 noisy-digit recognition tasks [8, 9]. Aurora 3 is a multi-lingual speech database, consisting of digit-string utterances in Danish, German, Finnish and Spanish. It provides a platform for fair comparison between systems of different front-ends. All the results reported in this paper follow the Aurora 3 evaluation guidelines.

3.2. Results

We first evaluate the number of vectors to be included in C^t, and decide to use $T = 15$. For the static features we use 12 MFCC features and the log energy, making $K = 13$. Therefore, the initial matrix C^t is of size 13×15.

Table 1 lists the experimental results on the Aurora 3 database. The entries in the table are the averaged relative improvements of word error rates over the baseline.

Consistent performance across different methods have been observed in the experiments. Specifically, Method H achieves the best performance, while Method G yields the worst performance, in all languages. Given that Method G and Method H differ only in the cepstral time coefficients they include in the final feature vector, it is fair to say that *the zeroth cepstral time coefficient is detrimental to recognition accuracy*.

Methods E, Method F, and Method I yield mixed results. In Finnish, Method E outperforms Method F and Method I. In Spanish and Danish, Method F outperforms Method I and Method E. Method E and Method F are similar in the sense that the first column (zeroth cepstral time coefficients) are normalized, and then used in procedures similar to delta and acceleration feature extraction, in the frequency domain rather than in the time domain. It is not surprising that they have similar performance level.

Table 1: *The overall (averaged over conditions) relative improvements of the word error rates in the Aurora 3 tasks.*

	German	Spanish	Finnish	Danish
E	-12.4	16.2	16.5	16.3
F	-10.5	22.4	10.8	16.3
G	-58.1	-29.0	-42.9	-19.2
H	7.5	26.6	25.4	23.2
I	-10.8	19.8	8.5	13.1

The comparison of Method G and H concludes that the zeroth CTC is detrimental of recognition accuracy. The zeroth CTC corresponds to the first column of CTM. Therefore in Method E and F, we try schemes of normalizing the first column of CTM. In Method E we divide the first column of CTM by T, and in Mthod F we normalize the value of first column to the range -1 to 1. The performance of E and F given in Table 1 are better than the baseline. Lastly, we also try Method I, which uses only CTCs, and excludes MFCCs. Its recognition accuracy is also better than the baseline.

Figure 2 plots the temporal sequences of the fifth dimension of the third column (Dimension 31 out of 39) of the feature vectors of Method B, F, and H of a pair of Danish utterances. The pair consists of an utterance of Channel 0 (the cleaner instance) and an utterance of Channel 1 (the noisier instance). Specifically, using our previously defined notations, Figure 2(B) is the plot of $\triangle^2 f_5^t$, Figure 2(F) is the plot of $\breve{F}_{53}^t$, and Figure 2(H) is the plot of $\breve{H}_{53}^t$. It appears that the difference between Channel 0 and Channel 1 is smaller in the cases of (F) and (H) than in the case of (B). Therefore the mismatchedness is reduced.

Table 2 lists the experimental results of Method H on the Aurora 3 database, given as percent word error rate (WER) results. These results include the four Aurora 3.0 languages (Finnish, Spanish, German, and Danish) and the Well-Matched(WM), Medium-Matched(MM), and Highly-Mismatched(HM) training/testing cases.

Table 2: *Our most recent Aurora 3.0 results using the method H, given as percent word error rate (WER) results. These results include the four Aurora 3.0 languages (Finnish, Spanish, German, and Danish) and the Well-Matched(WM), Medium-Matched(MM), and Highly-Mismatched(HM) training/testing cases.*

Aurora3 Reference Word Error Rate				
	German	Spanish	Finnish	Danish
WM	9.4	13.1	9.5	20.4
MM	21.9	26.3	27.5	50.6
HM	25.7	57.8	69.6	66.8

Aurora3 Word Error Rate, Method H				
	German	Spanish	Finnish	Danish
Well	9.1	9.7	7.0	15.4
Mid	19.8	18.4	21.3	39.0
High	21.7	45.4	50.2	52.4

Aurora3 Relative Percentage Improvement					
	German	Spanish	Finnish	Danish	Avg.
Well	4.4	26.0	26.2	24.5	20.3
Mid	5.3	29.9	22.7	22.9	20.2
High	15.5	23.0	27.9	21.5	22.0
overall	7.5	26.6	25.4	23.2	20.7

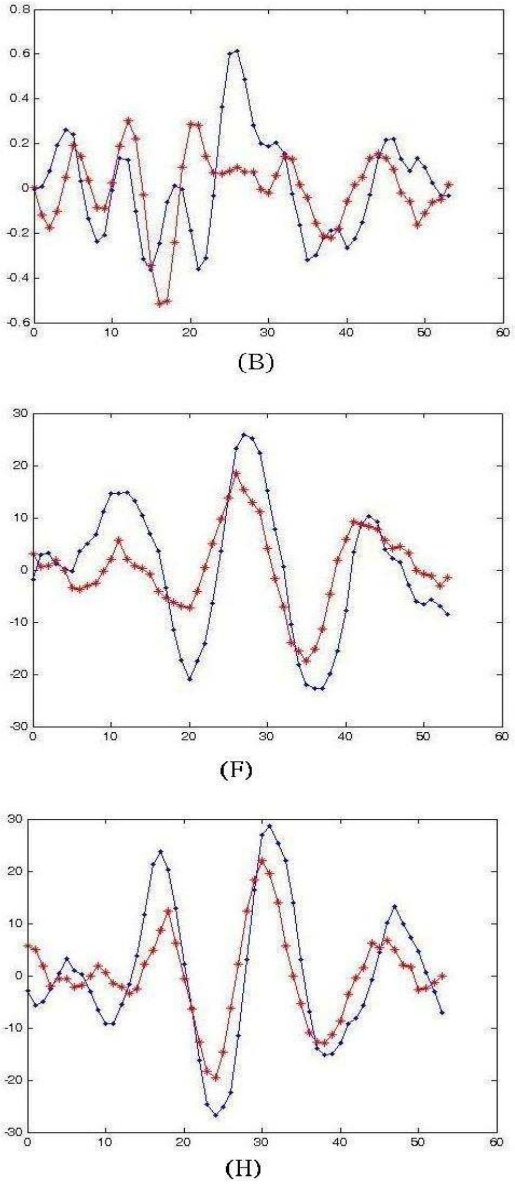
4. Conclusion and Future Work

In this paper, we use five difference feature sets based on the cepstral time coefficients. Method E and F, which first normalize the first column and then apply the delta and delta-delta operations on the first 3 columns of CTM, lead to performance gains over the baseline. Method G and H, which combine different sets of columns of CTM with the raw MFCC vector, lead to mixed results. Method I, which uses all cepstral time coefficients, leads to improvement. Overall, the combination of raw MFCC and the second and the third columns of CTM yields the best results among all experimented feature sets.

5. References

[1] S. Boll, “Supression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, April 1979.

[2] A. Berstein and I. Shallom, “An hypothesized Wiener filtering approach to noisy speechrecognition,” in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, 1991, pp. 913–916.



(B)

(F)

(H)

Figure 2: *Plot of Dimension* 31 *(out of* 39*) of a Danish utterance recorded in two mismatched channels. (B) is the* $\triangle^2 f_5^t$*, (F) is* $\breve{F}_{53}^t$*, and (H) is* $\breve{H}_{53}^t$*. The horizontal axis is the frmae index and the vertical axis is the feature value. The dotted line ('.') represents Channel 0 and the starred line ('*') represents Channel 1.*

[3] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[4] O. Viikki, D. Bye, and K. Laurila, "A recursive feature vector normalization approach for robust speechrecognition in noise," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2, 1998.

[5] A. de La Torre, A. Peinado, J. Segura, J. Perez-Cordoba, M. Benitez, and A. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[6] B. Milner, “Inclusion of temporal information into features for speechrecognition,” in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 1, 1996.

[7] ——, “A comparison of front-end configurations for robust speechrecognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. Proceedings.(ICASSP’02)*, vol. 1, 2002.

[8] Motorola Au/374/01, “Small vocabulary evaluation: Baseline mel-cepstrum performances with speech endpoints,” October 2001.

[9] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, “Speechdat-car: A large speech database for automotive environments,” in *Proceedings of the II LREC Conference*, vol. 1, no. 2, 2000.