A COMPLEXITY REDUCTION OF ETSI ADVANCED FRONT-END FOR DSR

Jin-Yu Li, Bo Liu, Ren-Hua Wang, Li-Rong Dai

iFlytek Speech Lab, University of Science and Technology of China, Hefei, Anhui, P.R.China
{jinyuli, liubo}@ustc.edu, {rhw, lrdai}@ustc.edu.cn

ABSTRACT

In Oct. 2002, Advanced Front-End (AFE) for Distributed Speech Recognition (DSR) was standardized by ETSI. In order to use AFE feature on low computational resource devices, we propose a novel approach to improve the computational efficiency. In our new algorithm, the structure of two-stage mel-warped Wiener filtering algorithm, which is the main part of AFE, is modified. Wiener filter is constructed and applied directly in mel-warped filter-bank domain. The measures we take make many time-consuming operations in original algorithm completely unnecessary, including the re-calculations of power spectrum and the time-domain convolution operations. Consequently, a large amount of computations are saved. Experiments show that the new approach can substantially reduce the computation load while preserving the excellent performance of the ETSI AFE.

1. INTRODUCTION

Owing to some factors, such as additive noises, channel mismatch and Lombard effects, speech recognition systems that work well in laboratories may suffer from severe performance degradation in realistic applications.

In spite of its long history, Wiener filtering is still an effective and widely used technique for robust speech recognition. An improved version of Wiener filtering, two-stage mel-warped Wiener filtering [1], has been approved as the main part of the ETSI AFE for DSR [2]. However, in original two-stage mel-warped Wiener filtering algorithm, the Wiener filter is constructed in linear-frequency domain using power spectrum, whereas the applying of the filter on the signal is in the time-domain using convolution operations. Such a time-frequency switch requires the power spectrum to be re-calculated at each stage of the algorithm. The operations above will introduce quite a number of additional computations.

For some low computational resource devices, such extra computation load may be unacceptable. So we propose a new computational efficient algorithm, in which both the construction and the applying of Wiener filter are put into the mel-warped filter-bank domain. Therefore the time-consuming convolution operations and the re-calculation of power spectrum can be avoided. Consequently, the computation load is reduced.

The rest of this paper is organized as follows. In section 2 we analyze the causes of large computation load of AFE, then propose a solution to the problem. The details of our proposal are described in section 3. In section 4, the results of comparative experiments are listed. Finally, we present the conclusions in section 5.

2. MODIFICATIONS TO ORIGINAL AFE SYSTEM

The AFE for DSR [2] can be roughly divided into two parts: the terminal side front-end and the server side feature processing. Only the terminal part is considered in this paper, since the contribution of server side part to the overall performance is comparatively trivial.

Three modules are implemented on the terminal side. They are noise reduction module, waveform processing module and blind equalization module. Two-stage mel-warped Wiener filtering algorithm [1] is the main body of the noise reduction module and is very time-consuming. Therefore our modifications are primarily concentrated on this algorithm. The other two modules are kept intact except that the waveform processing module is performed before the noise-reduction module.

Some operations in original two-stage mel-warped Wiener filtering algorithm will cause large computation load. First, the construction of Wiener filter in linear-frequency domain requires the power spectrum to be calculated at both stages of the algorithm, so it introduces the power spectrum re-calculations at both second stage and cepstrum calculation part. Second, the Wiener filter is applied in time-domain by time-consuming convolution operations. Both the spectrum re-calculation and convolution operation contribute to the large computation load of the algorithm.

In order to improve the efficiency, we propose a new structure for Wiener filtering algorithm, called two-stage Mel-warped filter-bank Wiener filtering. The block diagram of the proposed algorithm is shown in Figure 1.

The new algorithm is based on the mel-warped triangular filter-bank energies. We reduce the computation load from three aspects. First, the mel-warped Wiener filter coefficients are directly computed using the mel-warped triangular filter-bank energies. Since bands of triangular filter-bank are much fewer than bins of linear-frequency FFT power spectrum, the number of computations is effectively reduced. Second, mel-warped Wiener filter coefficients are smoothed and applied back on mel-warped filter-bank energies, because frequency-domain Wiener filter coefficients can also be viewed as the gains of the spectrum. This measure makes the time-domain convolution operations of applying the Wiener filter completely unnecessary. Third, since the de-noised mel-warpd filter-bank energies, instead of the de-noised time-domain signal, are fed into the next stage, the re-calculations of power spectrum are also

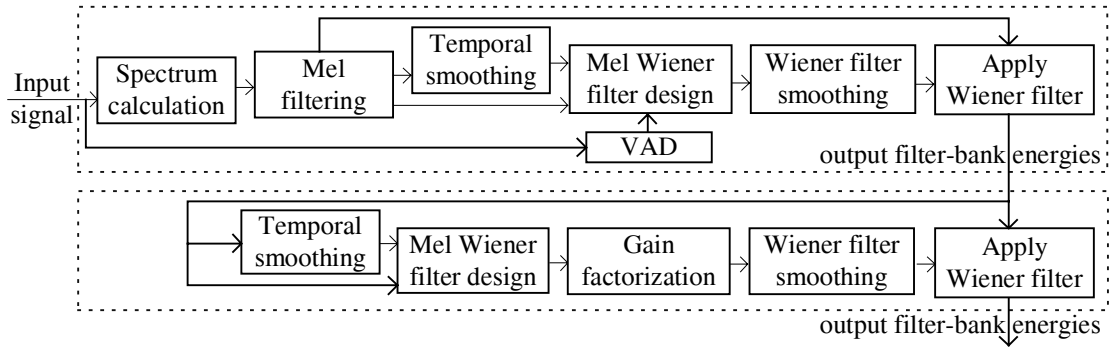

Figure 1: Block diagram of two-stage Mel-warped filter-bank Wiener filter algorithm

avoided. The power spectrum is calculated only once in the whole algorithm.

3. IMPLEMENTATION DETAILS

The implementation of the three main modules of our modified AFE is explained in detail as follows.

3.1. SNR-dependent Waveform Processing

SNR-dependent waveform processing [3] is a time-domain noise reduction method adopted in AFE. Since our new Wiener filtering process is not performed in time-domain, we have to move SNR-dependent waveform processing module to the front of the AFE, but without any modification in the module itself. Our experimental results show that only minor performance degradation is introduced by the location shift of this module.

3.2. Two-stage mel-warped filter-bank Wiener filtering

Two-stage mel-warped filter-bank Wiener filtering algorithm, the main body of noise reduction module, is proposed to simplify the original two-stage mel-warped Wiener filtering. The principle of the new algorithm has been introduced in section 2, and the implementation details are explained below.

First, the power spectrum of speech signal is calculated, all the configurations are just the same as those used in original AFE, including framing, windowing and FFT operations. The mel-warped triangular filter-bank is applied on the power spectrum to get the energy of each band. The mel-warped filter-bank we choose has 25 triangular filters without coefficients normalization.

Then we obtain the mel-warped Wiener filter coefficients from the mel-warped triangular filter-bank energies in Mel Wiener filter design part. We use the same computation equations as those used in the linear-frequency Wiener filter construction process of the original AFE, with the power spectrum of FFT bins replaced the output of the mel-warped triangular filter-bank bands.

The time-domain impulse response is computed from mel-warped Wiener filter coefficients using Mel-IDCT operation, which is not a time-consuming operation. The time-domain impulse response is then truncated, just as that in the original algorithm.

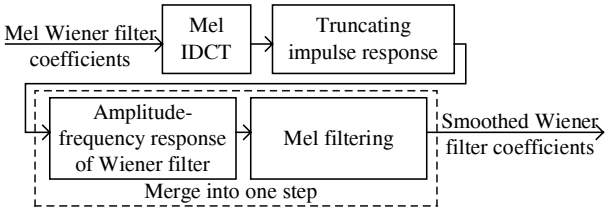

Figure 2: Smoothing of Mel-warped Wiener filter coefficients

Then we move to Wiener filter coefficients smoothing, as shown in Figure 2. It is well known that Wiener filter coefficients can also be viewed as the amplitude-frequency response, or equivalently, the gains that can be directly applied on the power spectrum or energies. If amplitude-frequency response is ready, the Wiener filter can be applied in frequency-domain using simple multiplication operations.

The smoothed Wiener filter coefficients H_{mel} are computed from truncated impulse response h_{WF} by two steps (Figure 2). First, amplitude-frequency response of Wiener filter H is obtained from truncated impulse response h_{WF} according to equation (1). Second, Mel filtering is applied on the amplitude-frequency response H to get H_{mel} as shown in equation (2). However, it is proved that the two steps can be merged into only one step. The merged computation is expressed by equation (3). The computation of equation (3) is very fast.

$$H(bin) = \sum_{n=0}^{N_{FFT}} h_{WF}(n) \times \exp(-j\frac{2\pi \cdot n \cdot bin}{N_{FFT}}) = h_{WF}(0) + \sum_{n=1}^{(K_{FL}-1)/2} h_{WF}(n) \times \left[2\cos(\frac{2\pi \cdot n \cdot bin}{N_{FFT}})\right] \quad (1)$$

where N_{FFT} is the FFT length, K_{FL} is the length of truncated time-domain impulse response of Wiener filter.

$$H_{mel}(k) = \sum_{i=0}^{N_{FFT}/2} W(k,i) \times H(i) \quad (2)$$

W denotes the coefficients of triangular filter-bank.

$$H_{mel}(k) = \sum_{n=0}^{(K_{FL}-1)/2} h_{WF}(n) \times B(n,k) \quad (3)$$

where B is the basis of the merged transformation, expressed as equation (4):

$$\begin{cases} B(n,k) = \sum_{i=0}^{N_{FFT}/2} W(k,i) \times \left[2\cos(\frac{2\pi \cdot n \cdot i}{N_{FFT}}) \right] \\ B(0,k) = \sum_{i=0}^{N_{FFT}/2} W(k,i) \end{cases} \quad (4)$$

Then, we obtain de-noised mel-warped filter-bank energies by applying Wiener filter in mel-warped filter-bank domain using simple multiplication operations. That is the output of the first stage of Wiener filtering.

The de-noised filter-bank energies are directly fed into the second stage of Wiener filtering. Then an almost identical Wiener filtering process is repeated. It is clear that the re-calculation of the power spectrum of speech signal and the Mel filtering at the second stage of Wiener filtering is completely unnecessary, because the input to the second stage is already the de-noised filter-bank energies.

Finally, a logarithm function is applied on the outputs of the second stage, and 13 cepstral coefficients are calculated from log filter-bank energies by applying a DCT.

3.3. Blind Equalization

A blind equalization algorithm [4] is applied on the cepstral coefficients to mitigate the channel effects in AFE. We use the same algorithm as that implemented in AFE.

4. EXPERIMENTS

4.1. Databases and Back-End Configurations

We evaluate the performance and computation load of the proposed method on Aurora2 database [5], which is a subset of TI digits database contaminated by additive noises and channel effects. And the same back-end configurations as those adopted in the evaluation of ETSI AFE standard [6] are used in our experiments.

4.2. Experimental Results

First, we compare the computation load of our two-stage mel-warped filter-bank Wiener filtering algorithm (Filter-Bank WF) with that of the original two-stage mel-warped Wiener filtering (WF). MFCC baseline Front-End distributed by ETSI on Apr. 2000 [7] is also used as a reference. Four types of operations are considered as shown in Figure 3. They are floating addition (subtraction), floating multiplication, floating division and non-linear operation (such as logarithm). It is obvious that the computation load of our Filter-Bank Wiener filtering algorithm is just a little larger than that of the MFCC base line front-end, but much smaller than that of the original algorithm, and about two thirds of the original computation load is saved.

Then the performances of the two algorithms are compared. In this paper, both the absolute performance and the performance relative to MFCC baseline (WI007 baseline [7]) are listed. It is interesting to find that the overall performance of

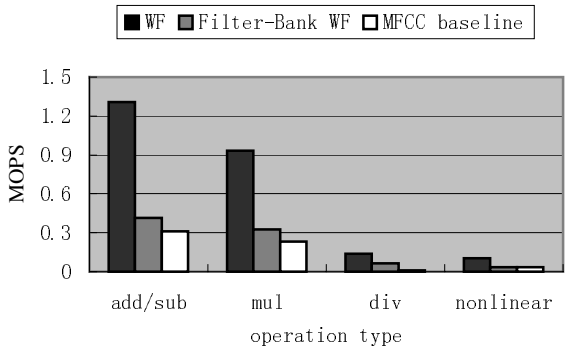

Figure 3: The computation load of Wiener filtering

Aurora2 Absolute Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	91.26	90.28	86.04	89.82
Clean	84.46	83.08	78.64	82.74
Average	87.86	86.68	82.34	**86.28**

Aurora2 Relative Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	28.27%	29.20%	13.97%	25.24%
Clean	59.79%	61.77%	36.91%	56.79%
Average	44.03%	45.49%	25.44%	**41.01%**

Table 1: Absolute and Relative Performance of original Wiener filtering

Aurora2 Absolute Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	90.80	89.86	87.60	89.78
Clean	84.35	82.30	80.98	82.86
Average	87.58	86.08	84.29	**86.32**

Aurora2 Relative Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	24.51%	26.12%	23.58%	24.94%
Clean	59.53%	60.01%	43.81%	57.08%
Average	42.02%	43.06%	33.70%	**41.01%**

Table 2: Absolute and Relative Performance of proposed Wiener filtering

the two Wiener filtering algorithms is almost the same, as listed in Table 1 and Table 2. This result confirms the correctness of our modifications on original Wiener filtering algorithm. Each of above two Wiener filtering algorithms is combined with both waveform processing module and blind equalization module to form the abridged AFE systems, without the server side feature processing and feature compression-decoding part. The

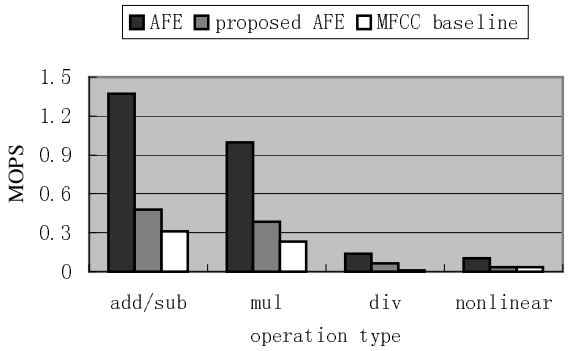

Figure 4: The computation load of three systems

Aurora2 Absolute Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	92.20	91.54	89.21	91.34
Clean	87.18	86.29	83.25	86.04
Average	89.69	88.92	86.23	**88.69**

Aurora2 Relative Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	36.01%	38.41%	33.50%	36.38%
Clean	66.83%	69.01%	50.52%	65.03%
Average	51.42%	53.71%	42.01%	**50.71%**

Table 3: Absolute and Relative Performance of AFE (abridged but unmodified)

Aurora2 Absolute Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	91.28	91.31	89.71	90.98
Clean	86.31	85.98	84.15	85.74
Average	88.79	88.64	86.93	**88.36**

Aurora2 Relative Performance				
Training Mode	Set A	Set B	Set C	Overall
Multi	28.41%	36.70%	36.58%	33.70%
Clean	64.58%	68.32%	53.18%	64.30%
Average	46.49%	52.51%	44.88%	**49.00%**

Table 4: Absolute and Relative Performance of proposed AFE (abridged and modified)

abridgement will introduce about 2% performance degradation, compared with the unabridged AFE system, which gets 53% performance.

The computation load comparison of AFE systems is shown in Figure 4, which is very similar to Figure 3, except a little more addition and multiplication operations.

As shown in Table 3, the performance of the abridged AFE system is 50.71%, while our proposed AFE gets 49.00% (Table 4). It is clear that there is slight performance degradation (less than 2%) due to our modifications. However, compared with the substantial computation load saved, such performance degradation is acceptable, especially for low computational resource devices.

5. CONCLUSIONS

We have proposed a novel, efficient algorithm to replace the two-stage Wiener filtering in AFE standard for DSR.

In our new algorithm, both the construction and the applying of Wiener filter are in mel-warped filter-bank domain, so the convolution operations in time-domain and the re-calculation of power spectrum are not necessary. Therefore, a large amount of computations are saved.

Both the computation load and the performance of the modified and original versions of two-stage mel-warped Wiener filtering are compared on Aurora2 database. No performance degradation is observed, and more than two thirds of computation load of original algorithm is saved.

Together with the SNR-dependent waveform processing module and the blind equalization module, the two versions of Wiener filtering are compared as a part of abridged AFE systems. The experiments show that our proposal can achieve a substantial decrease in computation load at the cost of very slight performance degradation.

The method proposed in this paper is especially suitable for the low-resource computing environments, such as embedded devices.

6. REFERENCES

[1] A. Agarwal, Y. M. Cheng, "Two-stage Mel-warped Wiener Filter for Robust Speech Recognition". The 1999 International Workshop on Automatic Speech Recognition and Understanding (ASRU'99), pp. 67-70, 1999.

[2] ETSI standard doc. “Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced feature extraction algorithm”, ETSI ES 202 050 Ver.1.1.1 (2002-10).

[3] D. Macho, Y.M. Cheng, “SNR-dependent Waveform Processing for Robust Speech Recognition”, Proc. ICASSP’01, pp. 305-308, 2001.

[4] L. Mauuary, “Blind Equalization in the Cepstral Domain for Robust Telephone based Speech Recognition”, Proc. EUSPICO’98, Vol. 1, pp. 359-363, 1998.

[5] H. G. Hirsch, D. Pearce, “The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions”, ISCA ITRW ASR 2000, Sept 2000.

[6] D. Macho, L. Mauuary, and B. Noe, “Evaluation of a Noise-Robust DSR Front-End on Aurora Databases”, Proc. ICSLP’02, pp. 17-20, 2002.

[7] ETSI standard doc. “Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms”, ETSI ES 201 108 v1.1.2 (2000-04).