An Efficient MFCC Extraction Method in Speech Recognition

Wei HAN, Cheong-Fat CHAN, Chiu-Sing CHOY and Kong-Pang PUN
Department of Electronic Engineering,
The Chinese University of Hong Kong
Hong Kong

***Abstract*—This paper introduces a new algorithm of extracting MFCC for speech recognition. The new algorithm reduces the computation power by 53% compared to the conventional algorithm. Simulation results indicate the new algorithm has a recognition accuracy of 92.93%. There is only a 1.5% reduction in recognition accuracy compared to the conventional MFCC extraction algorithm, which has an accuracy of 94.43%. However, the number of logic gates required to implement the new algorithm is about half of the MFCC algorithm, which makes the new algorithm very efficient for hardware implementation.**

I. Introduction

Automatic speech recognition by machine has been studied for decades. There are several kinds of parametric representations for the acoustic signals. Among them the Mel-Frequency Cepstrum Coefficients (MFCC) is the most widely used [1-3]. There are many reported works on MFCC, especially on the improvement of the recognition accuracy [4-6]. However, all these algorithms require large amount of calculations, which will increase the cost and reduce the performance of the hardware speech recognizer. The main objective of this work is to design a more hardware efficient algorithm.

In this paper we propose a novel and an efficient way to calculate MFCC. Section II introduces the conventional MFCC extraction algorithm. The new approach is introduced in Section III and the simulation results are presented in Section IV. The speech signals used in this paper are all from the Aurora 2 database [7]. The speech signal has a 10 dB signal-to-noise ratio and a spectrum between 0.3 kHz to 3.4 kHz at a sampling frequency of 8 kHz.

II. Conventional MFCC Extraction Method

Fig. 1 is the block diagram of the conventional MFCC extraction algorithm.

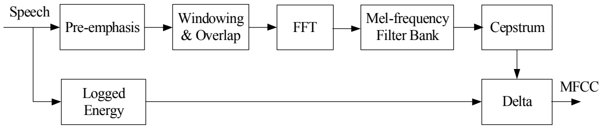

Figure 1. Block diagram of the conventional MFCC extraction algorithm

The speech is first pre-emphasized with a pre-emphasis filter 1-az^{-1} to spectrally flatten the signal, where "*a*" is between 0.9 and 1. In the time domain, the relationship between the output s'_n and the input s_n of the pre-emphasis block is shown in (1). The default value of *a* is 0.97 in HTK [8].

$$s'_n = s_n - as_{n-1} \tag{1}$$

Then the pre-emphasized speech is separated into short segments called frame. The frame length is set to 20ms (160 samples) to guarantee stationarity inside the frame. There is a 10ms (80 samples) overlap between two adjacent frames to ensure stationary between frames, as shown in Fig. 2.

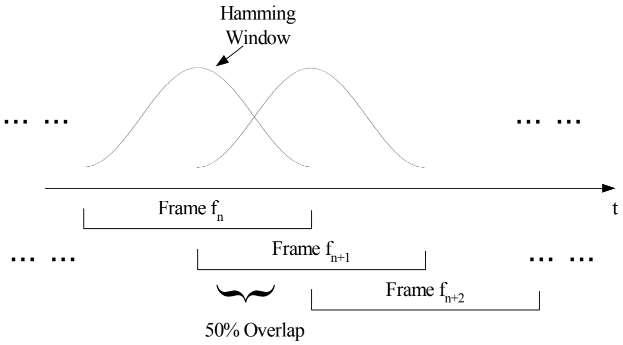

Figure 2. Frame blocking in the conventional MFCC extraction algorithm

A frame can be seen as the result of the speech waveform multiplies a rectangular pulse whose width is equal to the frame length. This will introduce significant high frequency noise at the beginning and end points of the frame because of the sudden changes from zero to signal and from signal to zero. To reduce this edge effect, a 160-points Hamming window is applied to each frame. The mathematical expression of the Hamming window is shown in equation (2),

$$Ham(N) = 0.54 - 0.46\cos(2\pi \frac{n-1}{N-1}) \tag{2}$$

where N is equal to 160, the number of points in one frame, and n is from 1 to N.

After the FFT block, the spectrum of each frame is filtered by a set of filters, and the power of each band is calculated. To obtain a good frequency resolution, a 256-point FFT is used [9]. Because of the symmetry property of FFT, we only need to calculate the first 128 coefficients. The filter bank consists of 33 triangular shaped band-pass filters, which ar**e** centered on equally spaced frequencies in the Mel domain between 0Hz and 4kHz, as shown in Fig. 3. The mapping from linear frequency to Mel-Frequency is shown in equation (3).

$$Mel(f) = 1127\ln(1 + \frac{f}{700}) \tag{3}$$

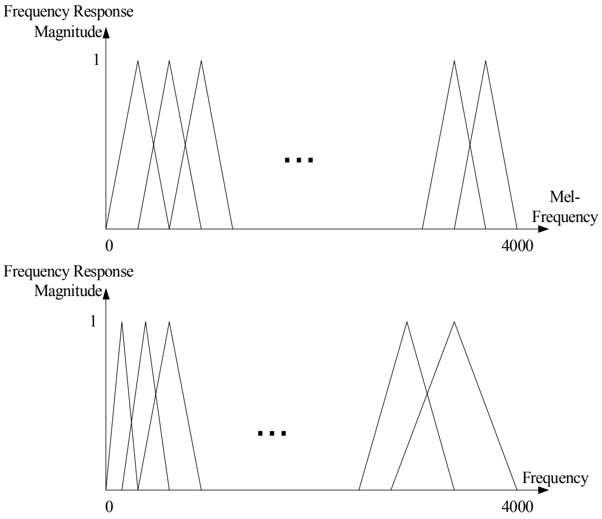

Figure 3. Mel-Frequency filter bank in Mel scale and normal scale.

We can calculate the Mel-Frequency cepstrum from the output power of the filter bank using equation (5),

$$c_n = \sum_{k=1}^{K} (\log S_k)\cos[n(k-0.5)\frac{\pi}{K}] \tag{4}$$

where S_k is the output power of the kth filter of the filter bank, and n is from 1 to 12. We can also calculate the logged energy of each frame as one of the coefficients,

$$E = \log \sum_{n=1}^{160} {s_n}^2 \tag{5}$$

which is calculated without any windowing and pre-emphasis. Up to now we have got 13 cepstrum coefficients. To enhance the performance of the speech recognition system, time derivatives are added to the basic static parameters. The delta coefficients are obtained from the following formula:

$$dc_t = \frac{2(c_{t+2} - c_{t-2}) + (c_{t+1} - c_{t-1})}{10} \tag{6}$$

After all the calculations, the total number of MFCC for one frame is 26.

III. THE PROPOSED METHOD FOR MFCC EXTRACTION

Each frame of the conventional algorithm described in Section II requires 160 multiplications for the window operation, 128xlog$_2$(256) multiplications for the FFT calculation, 128 multiplications for the filter power calculation and 33x12 multiplications for the DCT calculation. A total of 1708 multiplications are required for each frame, which requires a huge amount of computational power.

We propose a new MFCC algorithm that only requires half of the multiplication steps. The new algorithm is shown in Fig. 4. The dashed blocks highlight the main differences between the new and the old algorithms.

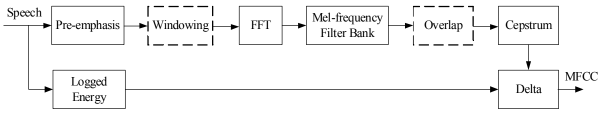

Figure 4. Block diagram of the proposed MFCC extraction algorithm.

There is no change to the algorithm of the pre-emphasis block. However, we make a modification to eliminate the multiplication step. Approximate the "*a*" value in equation (1) by 31/32 instead of 0.97, then (1) becomes:

$$\begin{aligned} s'_n &= s_n - as_{n-1} = s_n - \frac{31}{32}s_{n-1} \\ s'_n &= s_n - (s_{n-1} - \frac{s_{n-1}}{32}) \end{aligned} \tag{7}$$

We have replaced the multiplication as_{n-1} in (1) with simple addition and shift operations. The divide by 32 operation in (7) is simply shifting the binary number 5 bits to the right. The complex multiplication operation is replaced with simple shift operation without affecting the recognition accuracy, as shown in Table I of Section IV.

The original approach shown in Fig. 1 combines the window and overlap functions. In the new design, we move the overlap function after the filter bank as shown in Fig. 4. The speech is separated into segments called sub-frame here. One sub-frame is composed of 80 points and no overlap between them. Thus, one can picture a conventional frame f$_n$ consisting of two adjacent sub-frames sf$_n$ and sf$_{n+1}$, as shown in Fig. 5. As stated in Section II, the Hamming window is used mainly to reduce the edge effect, so the length of the Hamming window can be reduced from 160 points to 80 points consequently. As if the window size becomes smaller, the short-time spectrum will give a poorer frequency resolution but a better estimate of the overall spectral envelope, this modification will affect the recognition accuracy slightly as shown in Section IV.

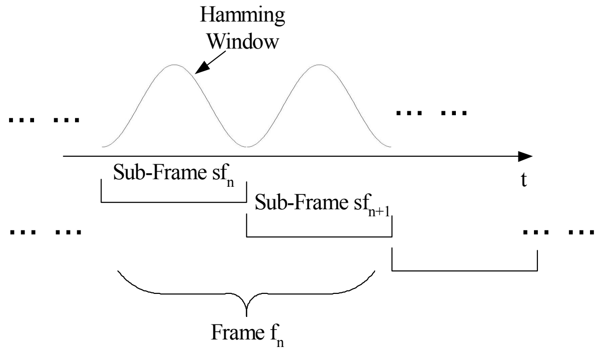

Figure 5. Block diagram of the proposed MFCC extraction algorithm.

The following block is the FFT, which is the same as the original design. However, the calculation is reduced by half to 128 points because of the new window size. We only need to calculate the first 64 coefficients because of the FFT symmetry.

We modify the filter bank from equally spaced triangular filters as shown in Fig. 3 to equally spaced rectangular filters as shown in Fig. 6. A filter bank is acceptable for speech recognition so far as its composite frequency response is flat over the entire frequency range of interest [10, 11]. Thus, a rectangle filter bank satisfies this requirement. In the conventional approach, the FFT outputs are multiplied by the characteristic of the triangular filter to generate the filter outputs and then these filter outputs are summed to generate the power of each filter. However, if we use a rectangular filter, the output characteristic of a rectangular filter is either a "1" or a "0", thus the operations are changed to "add" or "not add". For a 128-point FFT, the rectangular filter bank is reduced to 23 equally spaced rectangular filters based on the simulation results shown in Fig. 7, which indicates 23 filters produce the highest recognition accuracy. The original triangular filters require 128 multiplications per frame. However, the rectangular filters only require 120 additions per frame.

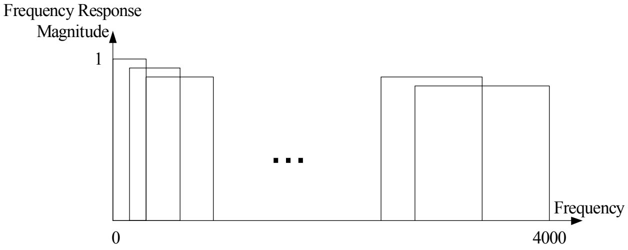

Figure 6. The new Mel-Frequency filter bank. (The Magnitude of all filters is 1, the difference in the figure is for readability.)

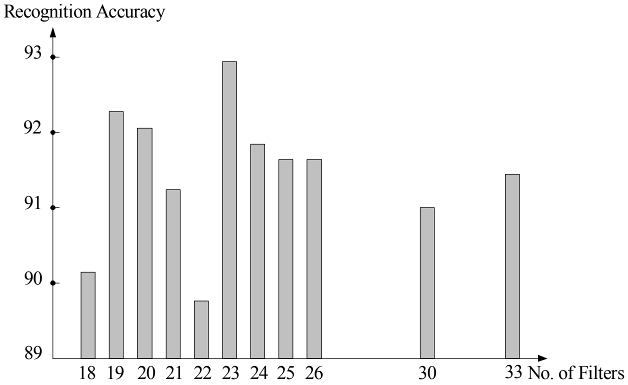

Figure 7. Recognition accuracy when using different number of filters in the rectangular filter bank.

The new overlap algorithm is illustrated in Fig. 8, where f_n and f_{n+1} represent the original 160-point frames with 50% overlap, and sf_n and sf_{n+1} represent the new 80-point non-overlapped sub-frames. We add the filter bank outputs $SF_{n,k}$ and $SF_{n+1,k}$ to generate the power coefficient $S_{n,k}$. The next power coefficient $S_{n+1,\ k}$ is equal to the sum of the filter outputs of sub-frame sf_{n+1} and sf_{n+2}. Thus, $S_{n,k}$ and $S_{n+1,k}$ are identical to the kth power coefficient of the original frame f_n and f_{n+1}. We have reduced almost half of the computation by moving the overlap operation to the end of the spectrum calculation.

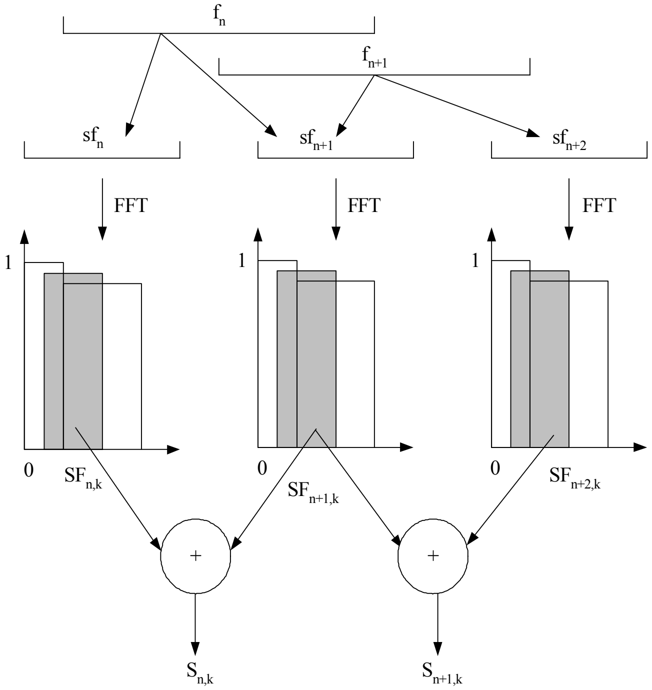

Figure 8. Overlapping in the new algorithm.

The following DCT and delta calculations are the same as the conventional algorithm. There are also 26 features in each frame as the original algorithm. The new feature extraction algorithm reduces the total number of multiplications from 1708 to 804 per frame. Each frame of the new algorithm requires 80 multiplications for the window operation, 64xlog$_2$(128) multiplications for the FFT calculation, and 23x12 multiplications for the DCT calculation.

IV. Simulation Results

We performed several simulations using different parameters to evaluate the performance of the new algorithm. The training and test speech data are from the ARURA 2 database. They are isolated English digits from "0" to "9", and for "0" there are two kinds of pronunciation: "zero" and "O". There are totally 1144 utterances for training and 468 test utterances. The speeches are 10dB SNR noise signal constructed by adding noises to the clean data. The training process and the back end of the speech recognition system were done by the HTK toolkit. The back end of the recognizer used the Hidden Markov Model (HMM) [1] approach. The models are from left to right with 8 states and there are 2 mixtures in each state, as illustrated in Fig. 9.

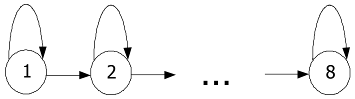

Figure 9. 8-state left-right HMM..

The recognition results of the new and old algorithms with different window lengths and FFT points are summarized in Table I. F1 is the original MFCC features extraction algorithm. F5 is the proposed new algorithm. As we have pointed out in Section III, if we change the "*a*" value of equation (1) from 0.97 to 31/32 then the multiplication step in the pre-emphasis block can be replaced with shift operation without affecting the recognition accuracy. This approximation is verified by the fact that there is no change of recognition accuracy of F1 and F2. F3 segments the speech into 80-point sub-frames, using the proposed architecture in Fig. 4 but keeping the other settings as same as F1. There is a slight drop in recognition accuracy. F3 and F4 compare the recognition accuracy between triangular and rectangular filters. F4 and F5 compare the new algorithm at different FFT points. From Table I, we can find that F5 produces relatively high recognition accuracy with the minimum requirement of calculation power.

V. Conclusion

We have demonstrated that the new extraction algorithm reduces the number of multiplications from 1708 to 804 with only 1.5% drop in recognition accuracy. The new algorithm is more efficient for hardware implementation than the original algorithm. We are in the process of building a FPGA recognizer with the new algorithm and we expect the new algorithm will have significant improvements on the hardware performance such as power consumption, speed, and cost.

TABLE I. Recognition Accuracy of Different Feature Set

Feature Set	a	Window Length	FFT Point	Filter Shape	No. of Filters	Recognition Accuracy
F1	0.97	160	256	Triangle	33	94.43%
F2	31/32	160	256	Triangle	33	94.43%
F3	31/32	80	256	Triangle	33	92.29%
F4	31/32	80	256	Rectangle	33	92.08%
F5	31/32	80	128	Rectangle	23	92.93%

References

[1] L. Rabiner and Biing-Hwang Juang, Fundamentals of Speech Recognition, Prentice Hall PTR, c1993

[2] Joseph W. Picone, "Signal Modeling Techniques in Speech Recognition", Proceedings of the IEEE, vol. 81, No. 9, pages 1215--1247, 1993.

[3] Steven B. Davis and Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-28, No. 4, August 1980.

[4] Qi Li, Frank K, Soong and Olivier Siohan, "A High-Performance Auditory Feature for Robust Speech Recognition", 6th International Conference on Spoken Language Processing, Beijing, October 2000.

[5] Jia Lei and Xu Bo, "Including detailed information feature in MFCC for large vocabulary continuous speech recognition", Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on, Volume 1, 2002 Page(s):I-805 - I-808 vol.1.

[6] Phadke, S.; Limaye, R.; Verma, S.; Subramanian, K., "On design and implementation of an embedded automatic speech recognition system", VLSI Design, 2004. Proceedings. 17th International Conference on, 2004 Page(s):127 – 132.

[7] Hans-Gunter Hirsch and David Pearce, "The AURORA Experimental Framework for the Performance Evaluation of Speech Recognition Systems Under Noisy Conditions", ISCA ITRW ASR2000, Paris, France, Sept. 2000.

[8] S. J. Young, P.C. Woodland and W. J. Byrne, The HTK BOOK (for HTK Version 2.2), Entropic Ltd., Jan. 1999

[9] Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 1997, page(s):169-174.

[10] B. A. Dautrich, L. R. Rabiner, and T. B. Martin. "On the effects of varying filter bank parameters on isolated word recognition." IEEE Trans. Acoust., Speech, Signal Processing, ASSP-31(4):793-807, 1983.

[11] Fang Zheng, Guoliang Zhang and Zhanjiang Song, "Comparison of Different Implementations of MFCC", J. Computer Science & Technology, 16(6): 582-589, Sept. 2001.