ICSP2008 Proceedings

Further Studies of a FFT-Based Auditory Spectrum with Application in Audio Classification

Wei Chu and Benoît Champagne
Department of Electrical and Computer Engineering
McGill University, Montréal, Québec, Canada
wei.chu@mail.mcgill.ca, benoit.champagne@mcgill.ca

Abstract

In this paper, the noise-robustness of a recently proposed fast Fourier transform (FFT)-based auditory spectrum (FFT-AS) is further evaluated through speech/music/noise classification experiments wherein mismatched test cases are considered. The features obtained from the FFT-AS show more robust performance as compared to the conventional mel-frequency cepstral coefficient (MFCC) features. To further explore the FFT-AS from a perspective of practical audio classification, an audio classification algorithm using features derived from the FFT-AS is implemented on the floating-point DSP platform TMS320C6713. Through various optimization approaches, a significant reduction in the computational complexity is achieved wherein the implemented system demonstrates the ability to classify among speech, music and noise under the constraint of real-time processing.

1. Introduction

In a previous work [1], we have proposed an algorithmic implementation to calculate a fast Fourier transform (FFT)-based auditory spectrum (FFT-AS) motivated from the study of the noise-suppression property of an early auditory (EA) model which outputs an auditory spectrum [2].

The noise-robustness of the proposed FFT-AS was evaluated and confirmed through audio classification experiments wherein clean speech and music clips were involved in the training whereas noisy speech and music clips were used for the testing. It is of interest to further evaluate this robustness, or the ability to handle the mismatch between the SNR values of the training and testing samples, from other perspectives, as specifically to train the algorithm with noisy samples while testing with clean samples.

978-1-4244-2179-4/08/$25.00 ©2008 IEEE

As discussed in [1], besides the robustness to noise, the potential of the proposed FFT-AS lies in its low computational complexity as compared to the original auditory spectrum in [2]. It is therefore of interest to conduct further research on the computational complexity of the FFT-AS when implemented for real-time operations on a DSP platform.

In this paper, the noise-robustness of the FFT-AS proposed in [1] is further evaluated through speech/music/noise classification experiments. Results from mismatched tests confirm the superior robustness of the features derived from the FFT-AS as compared to the conventional mel-frequency cepstral coefficient (MFCC) features. In addition, using the discrete cosine transform (DCT)-based features obtained from the FFT-AS, a classification algorithm is implemented on the floating-point DSP platform TMS320C6713 [3]. Through a set of optimization approaches, a significant reduction in the computational complexity is achieved.

The paper is organized as follows. Section 2 briefly reviews the classification algorithms and the associated feature sets. The classification performance is analyzed in Section 3. Section 4 discusses the DSP implementation and optimization of a classification algorithm.

2. Audio classification algorithm

2.1. FFT-based auditory spectrum

Fig. 1 shows the structure of the implementation proposed in [1] to calculate the FFT-AS wherein the main operations include the calculation of a short-time power spectrum, power spectrum selection using characteristic frequency values of the cochlear filters of the original EA model, and spectral self-normalization through a pair of fast and slow running averages.

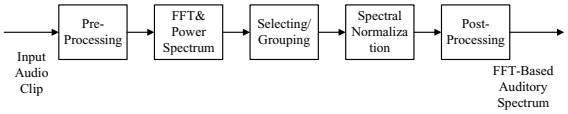

Fig. 1. The FFT-based implementation in [1].

2.2. Audio features

Both frame-level and clip-level features are calculated. Four sets of frame-level features are listed below:

- The 13-dimensional conventional MFCC features.
- The 13-dimentsional DCT-based feature sets calculated by applying DCT to both the original auditory spectrum and the FFT-based auditory spectrum.
- The 10-dimensional spectral features calculated using the proposed FFT-AS (see [1] for a description of the specific features).

The corresponding clip-level features are the statistical mean and variance values of these frame-level features calculated over a time window of one second. The clip-level features are used for the training and testing of the classification algorithm.

2.3. Classification

Recent years have seen an increasing interest in the use of support vector machine (SVM) algorithms in audio classification applications, and excellent performance with SVM has been reported. In this work, we use the SVM^{STRUCT} algorithm [4] for classification. For the purpose of performance comparison, the decision tree learning algorithm C4.5 [5] is also used.

3. Performance analysis

The total length of the audio samples is 12000 seconds, including clean speech, clean music and noise. The sampling rate is 16 kHz. These samples are divided equally into two parts for training and testing. Noisy samples are also digitally generated by adding noise segments to clean speech/music segments based on the long-term average energy measurement. Below, a training or testing data set with a specific SNR value, e.g. 15-dB set, refers to a data set consisting of noisy speech and noisy music (both with SNR = 15 dB), and noise. Audio classification experiments are conducted under both matched and mismatched situations, which refer to a match and mismatch between the SNR values of the training and testing sets, respectively. Results from matched tests may reveal the interclass discriminability while those from mismatched ones may indicate the noise-robustness which is the main focus of our research.

3.1. Training with clean data set

The error rates of three-class (i.e., speech, music and noise) classification are presented in Table. 1, where MFCC, DCT1, DCT2, and SPEC represent the conventional MFCC features, the DCT-based features obtained from the original auditory spectrum [2], the DCT-based features obtained from the FFT-AS [1], and the spectral features obtained from the FFT-AS, respectively. In Table 1, error rates presented under the category "Match (clean)" refers to the results of the matched test case, i.e., with clean set as the testing set, whereas those under "Mismatch" refer to the error rates of two mismatched tests, i.e., testing with 10-dB and 5-dB data sets.

Despite an excellent performance in the matched test case, the robustness of the MFCC features under mismatched tests is relatively poor. On the other hand, three sets of auditory-based features (i.e. DCT1, DCT2 and SPEC) are more robust in mismatched test cases as compared to the MFCC features. The DCT2 feature set yields the best average performance in the mismatched tests. As for the classification, in most cases, SVM outperforms C4.5. In addition, the DCT2 feature set slightly outperforms the DCT1 feature set. As discussed in [1], besides the robustness to noise, the potential of the FFT-AS lies in its low computational complexity compared to the original auditory spectrum.

Table 1. Error classification rates with clean set as the training data (%)

	SVM			C4.5		
	Match	Mismatch		Match	Mismatch	
	(clean)	10dB	5dB	(clean)	10dB	5dB
MFCC	1.9	41.8	50.3	2.8	40.7	47.4
DCT1	3.3	13.4	29.8	6.2	18.2	32.5
DCT2	2.2	11.1	27.3	4.8	10.6	25.4
SPEC	3.2	10.9	29.7	3.6	16.1	33.0

3.2. Distribution of features

Fig. 2 shows distributions of clip-level MFCC and DCT2 features. To simplify the analysis, we only use the first and the third components of each feature set to plot two-dimensional distributions. For the MFCC features, when the background noise level is increased to SNR = 15 dB, a relatively large change in the distribution can be seen (see Figs. 2(a) and 2(b)). Comparatively, under the same change of the SNR, the DCT2 features exhibit a relatively small change in the distribution (see Figs. 2(d) and 2(e)). The results in Fig. 2 may justify the robustness of the DCT2 features as compared to the MFCC features to some extent.

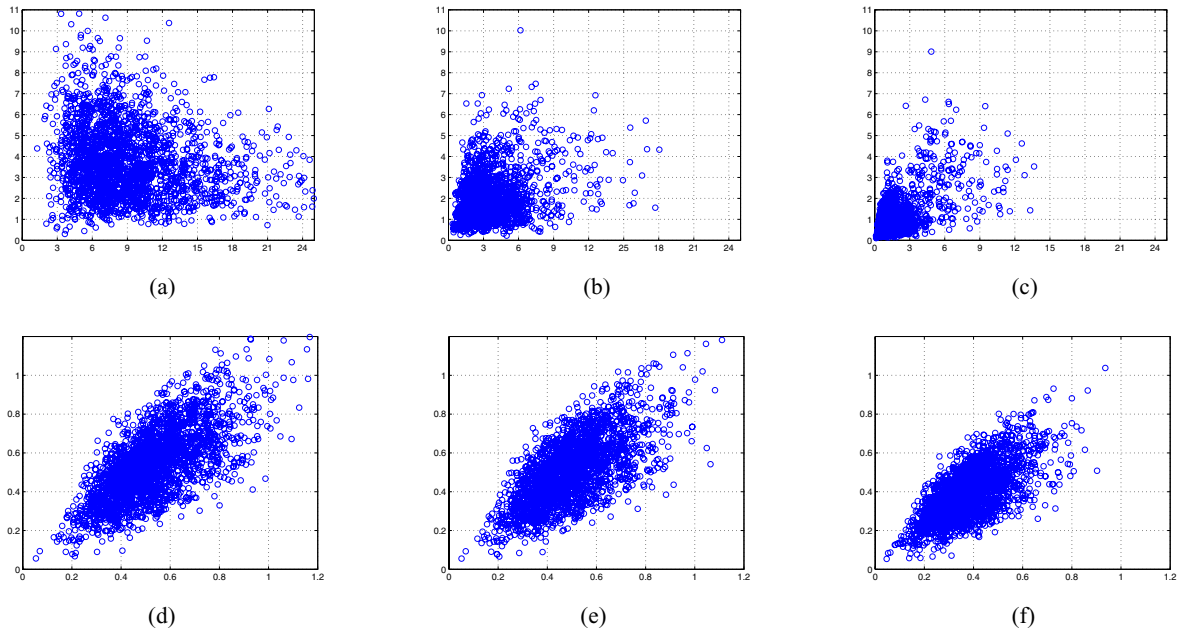

Fig. 2. Distributions of MFCC and DCT2 features of speech clips. For all figures, horizontal and vertical axes refer to the first and the third components of the feature vector respectively. (a) MFCC (clean). (b) MFCC (15 dB). (c) MFCC (10 dB). (d) DCT2 (clean). (e) DCT2 (15 dB). (f) DCT2 (10 dB).

For MFCC features, as the background noise level is increased, e.g., from SNR = 15 dB to 10 dB, the change in the distribution becomes smaller (see Figs. 2(b) and 2(c)). Therefore, if a noisy set (e.g., the 15-dB data set) is used for the training instead of the clean set, we may expect an improved performance in the mismatched test cases, e.g., using the 10-dB data set. Below, the robustness of the features is further investigated in mismatched tests wherein the 15-dB data set is used for the training.

3.3. Training with 15-dB data set

The error rates of the speech/music/noise classification are given in Table. 2, wherein "Match (15dB)" and "Mismatch" refer to the error rates from the 15-dB testing set, and the error rates from two mismatched testing sets (i.e., SNR = 10 and 5 dB), respectively.

In the matched test case (i.e. SNR = 15 dB), MFCC features with SVM again show the best discriminability with an error rate of 2.1%, whereas DCT2 and SPEC features with SVM perform best in the two mismatched test cases. Compared to the performance given in Table 1, the robustness of the MFCC feature set is improved, which is consistent with the discussion in Section 3.2. Overall, the robustness of the three auditory-based feature sets (i.e. DCT1, DCT2 and SPEC) in mismatched test cases is still better than that of the MFCC features, though the performance gap between the auditory-based and the conventional MFCC feature sets becomes smaller as compared to the results in Table 1.

Table 2. Error classification rates with 15-dB set as the training data (%)

	SVM			C4.5		
	Match	Mismatch		Match	Mismatch	
	(15dB)	10dB	5dB	(15dB)	10dB	5dB
MFCC	2.1	15.0	32.6	5.6	14.8	30.6
DCT1	3.8	6.9	21.7	6.4	10.5	26.7
DCT2	3.4	5.5	18.2	5.8	8.5	19.7
SPEC	3.4	6.4	17.6	4.0	8.1	22.1

3.4 Asymmetric performance

For conventional MFCC features, when training with the clean set, the error rates (with SVM) of the testing using the clean and 15-dB data sets are 1.9% and 32.0%, respectively. When training with the 15-dB data set, the corresponding error rates of the testing are 3.7% and 2.1%. For such a pair of symmetric mismatched tests (i.e. training with the clean set while testing with the 15-dB set, and vice versa), we have observed an asymmetric pattern in the error rates (i.e. 32% vs. 3.7%). However, for DCT2 features, the cor-

responding error rates of such a pair of symmetric mismatched tests are 4.2% and 3.3%. Thus, the performance gap with the DCT2 features is less significant than that with the MFCC features, pointing to the superior robustness of the DCT2 features.

4. Implementation on TMS320C6713 DSK

Besides the robustness to noise, the potential of the proposed FFT-AS lies in its low complexity as compared to the original auditory spectrum. To further explore this from a practical perspective, a speech/music/noise classification algorithm is implemented and optimized using the floating-point TMS320C6713 DSP Starter Kit (DSK) from Texas Instruments (TI) [3]. C6713 DSK is a low-cost full featured development platform which characterizes a 225 MHz 32-bit floating-point C6713 digital signal processor (DSP) and an AIC23 stereo codec. C6713 DSK is supported by TI's real-time eXpressDSP software and development tools [6].

4.1. Initial implementation

The DSP implementation is conducted in two steps: the initial implementation and the optimization. To reduce the requirements on the memory units of the DSP target, the algorithm is designed to work on a frame basis (i.e. 10 ms). Accordingly, the classification algorithm is organized into two parts, namely: the frame-based processing and the clip-based processing, which are shown in Fig. 3. For an input signal frame of 10 ms, the frame-based processing outputs a set of frame-level DCT2 features. The clip-level processing is triggered as soon as all signal frames in a clip (i.e. 1 s) have been processed. The C code contains a conventional *main* function and a function named *classify*. The processing modules shown in Fig. 3 are implemented in the function *classify* which is used as an interrupt service routine (ISR). Function *main* is effectively acting as the initial processing of the implementation.

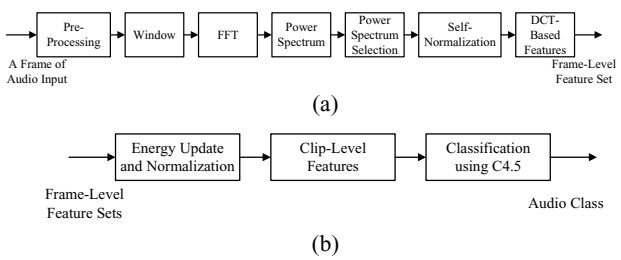

Fig. 3. Processing modules of the proposed C6713 DSK-based implementation. (a) Frame-based processing. (b) Clip-based processing.

Two host channels provided by C6713 DSK are used to transfer data [7]: the input channel is employed for the classification algorithm to access audio samples on the PC, whereas the output channel is used to send the classification decisions back to the host PC. Whenever a full signal frame is available in the input channel and a frame of free space is available in the output channel, a software interrupt is posted and the ISR *classify* is called to process a signal frame. This way, a block input/output-based classification system is implemented. A Matlab algorithm is also developed to monitor the execution of the code as it runs on the DSP target by displaying the output classification decisions.

The maximum computational complexity (MCC) and the average computational complexity (ACC) are defined below to measure the relative performance:

$$\text{MCC} = \text{MaxCount} / \text{Reference}$$
$$\text{ACC} = \text{AveCount} / \text{Reference}$$

where MaxCount and AveCount refer to, respectively: the maximum and the average runtime in ms (or instruction cycles) taken for the processing of a signal frame of 10 ms, and Reference is 10 ms (or 2.25×10^6 for instruction cycles). To meet the constraint of real-time processing, the MCC value of the classification algorithm (i.e. function *classify*) should be less than one. With the C6713 DSP/BIOS real-time analysis tool and APIs [7], we are able to instrument the target by capturing and uploading the real-time information that drives the code composer studio visual analysis tools [6]. Specifically, the information such as the instruction cycles or the runtime used for the execution of the code is reported in the Statistics View window.

At the stage of initial implementation, the main focus is placed on the functional correctness of the code. The correctness of the initial implementation is verified using various audio samples. The MCC and ACC values of the classification algorithm are 1.4830 and 1.4048 respectively. It is found that the calculation of the DCT-based features, the calculation of the FFT, and the spectral self-normalization are the three most computationally intensive modules (see Fig. 3). The initial implementation does not follow the real-time constraint. Optimizations are further applied.

4.2 Optimization

4.2.1. Compiler optimization. The TMS320C6x compiler *cl6x* performs various optimizations to improve the execution speed and to reduce the code size [8]. As the first step of optimization, file-level and program-level optimization options provided by *cl6x* are applied,

leading to reductions of 32% and 31%, respectively, in the MCC and ACC values for the function *classify*.

4.2.2. Coefficient tables. The heavy computational load of the DCT module comes in a large part from the calculation of the cosine coefficients. We can tabulate these coefficients instead of calculating them in real time. As such, the module is re-organized into two sub-modules, wherein the first one calculates the cosine coefficients while the second one calculates the DCT-based features using the cosine coefficients prepared by the first sub-module. The first sub-module is placed in the function *main* and acts as the initial processing of the algorithm. The computational complexity is determined only by the second sub-module which remains in the function *classify*. A reduction of 98% is thus achieved in MCC and ACC values for the DCT module.

4.2.3. C67x DSP library. The optimized FFT function *DSPF_sp_cfftr2_dit* from TI's C67x optimized DSP library [9] is used to replace the FFT algorithm in the initial implementation (a radix-2 decimation in time algorithm). Meanwhile, to bit-reverse the output from the function *DSPF_sp_cfftr2_dit* as required, the optimized function *DSPF_sp_bitrev_cplx* is also used. The computation of the relevant coefficients which are required by these two optimized functions is included in the initial processing part in the function *main*. The computational complexity is now determined by these two optimized functions. The reduction in the MCC and ACC values for the FFT module is about 96%.

4.2.4. C67x FastRTS library. To conduct optimization for the division and square-root operations in the spectral self-normalization module, we now change to use the C67x's assembly-optimized division and square-root subroutines, which are provided by the C67x fast run-time-support (FastRTS) library [10]. The MCC and ACC values for the self-normalization module are decreased by 78% and 79% respectively.

4.3 Experimental results

After applying the optimization approaches discussed in Section 4.2, a significant reduction in the computational complexity has been achieved for the proposed initial implementation. The reduction in the computational complexity is given in Table 4, where Classify, DCT, FFT and Self-Norm represent the function *classify*, the DCT module, the FFT module, and the self-normalization module, respectively. In Table 4, results are given in MCC/ACC pairs. From Table 4, the MCC of the optimized implementation is only 0.053, indicating that the algorithm only consumes 5.3% of the computational power of the C6713 DSK.

Table 4. Complexity information (MCC/ACC values) after applying optimizations

	Before	After	Reduction (%)
Classify	1.483/1.405	0.053/0.037	96.4/97.4
DCT	0.623/0.617	0.006/0.006	99.0/99.0
FFT	0.728/0.719	0.017/0.017	97.7/97.6
Self-Norm	0.038/0.038	0.008/0.007	78.9/81.6

5. Conclusions

In this paper, we have further evaluated the noise-robustness of the FFT-AS [1]. Audio classification results from mismatched tests confirmed the noise-robustness of the features calculated from the FFT-AS as compared to the conventional MFCCs. In addition, an audio classification algorithm was implemented on a floating-point DSP platform using the DCT-based features obtained from the FFT-AS. Through various optimization approaches, a significant reduction in the computational complexity has been achieved.

6. References

[1] W. Chu and B. Champagne, "A noise-robust FFT-based auditory spectrum with application in audio classification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 1, pp. 137-150, Jan. 2008.

[2] K. Wang and S. Shamma, "Self-normalization and noise-robustness in early auditory representations," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 3, pp. 421-435, Jul. 1994.

[3] *TMS320C6713 DSK Technical Reference*, Spectrum Digital, Stafford, TX, Jan. 2004.

[4] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.,* vol. 6, pp. 1453-1484, Sep. 2005. (http://svmlight.joachims.org/svm_struct.html.)

[5] J. R. Quinlan*, C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[6] *Code Composer Studio IDE Getting Started Guide*, Texas Instruments, Dallas, TX, May 2005.

[7] *TMS320 DSP/BIOS User's Guide*, Texas Instruments, Dallas, TX, Apr. 2004.

[8] *TMS320C6000 Optimizing Compiler User's Guide*, Texas Instruments, Dallas, TX, May 2004.

[9] *TMS320C67x DSP Library Programmer's Reference Guide*, Texas Instruments, Dallas, TX, Mar. 2006.

[10] *TMS320C67x FastRTS Library Programmer's Reference*, Texas Instruments, Dallas, TX, Oct. 2002.