

This document is downloaded from DR-NTU, Nanyang Technological University Library, Singapore.

Title	Feature Extraction Techniques for Low-Power Ambulatory Wheeze Detection Wearables
Author(s)	Acharya, Jyotibdha; Basu, Arindam; Ser, Wee
Citation	Acharya, J., Basu, A., & Ser, W. Feature Extraction Techniques for Low-Power Ambulatory Wheeze Detection Wearables. 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2017).
Date	2017
URL	<a href="http://hdl.handle.net/10220/43612">http://hdl.handle.net/10220/43612</a>
Rights	© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Feature Extraction Techniques for Low-Power Ambulatory Wheeze Detection Wearables

Jyotibdha Acharya<sup>1</sup>, Arindam Basu<sup>2</sup> and Wee Ser<sup>2</sup>

**Abstract**—Presence of wheezes in breathing sounds has been associated with several respiratory and pulmonary diseases. In this paper we present a novel low-complexity wheeze detection method based on frequency contour tracking for automatic wheeze detection. Two hardware friendly variants of the algorithm have also been proposed. Applying the proposed feature extraction algorithm we achieved very high classification accuracy (> 99%) at considerably low computational complexity (3×-6×) compared to earlier methods and the power consumption of the proposed method is shown to be significantly less (70×-100×) compared to ‘record and transmit’ strategy in wearable devices.

## I. INTRODUCTION

Wheeze is a high-pitched whistling sound produced by partially obstructed respiratory airways during breathing. Presence of wheezes has been used extensively as a diagnostic tool by medical professionals to detect lung and chronic pulmonary diseases such as asthma, COPD, bronchiolitis etc[1]. Traditionally auscultation using stethoscopes has been used to detect and monitor wheezes. But this method suffers from two major drawbacks, namely availability of trained medical professionals and subjectivity in diagnosis due to disparate interpretation of wheeze sounds by diagnosticians[2]. Moreover, for a better identification of the underlying medical condition, continuous monitoring and analysis of wheeze sounds is often preferred.

To circumvent these issues, several automatic wheeze detection methods have been proposed in past decades. Some commonly used feature extraction methods include statistical analysis of the amplitude and power spectrum[3], Shannon entropy measurement[4], time-frequency(T-F) continuity measurement[5] etc. Though some these methods obtained high classification accuracies, often they are too complex and power hungry to be implemented in low power wearable hardware.

This paper aims to describe a low complexity T-F continuity based algorithm for feature extraction and wheeze detection with high accuracy. Two hardware friendly variants of the algorithm with reasonably high detection accuracy have also been proposed. The following sections are organized as follows: Section (II-A) describes the frequency contour tracking (FCT) algorithm and design space exploration for optimizing algorithm parameters. Section (II-B) introduces the low complexity hardware variant of the algorithm. In

Section (III) we report the noise performance of the algorithm and compare performance and power requirement of different methods. In section (IV) we discuss the conclusion and future work.

## II. METHODS & THEORY

As evident from Fig.1, the spectrograms of wheeze signals are characterized by continuous frequency contours which distinguishes them from normal breathing sounds. These frequency contours are 1) continuous in time 2) varying in shape for different patients and 3) present in different frequency bands for different patients. In the proposed method we amplify the amplitude of the frequency channels which have a similar high amplitude frequency in its neighboring channels in previous time instants while diminish the amplitude of the frequency channels that does not have any such prior high amplitude frequencies. As a result, temporally continuous lines are gradually amplified while isolated noise points are suppressed.

### A. Software Method

In the first step of our proposed method, we use short-time Fourier transform (STFT) with a fixed size overlapping window to obtain the spectrogram of the wheeze signal. The input to the feature extraction algorithm is the spectrogram of the signal. The algorithm requires three predefined parameters,  $Ch\_num$ : Total number of frequency channels to consider from each window output,  $L$ : No. of high amplitude channels selected from each window output,  $F_d$ : Neighborhood size. The algorithm works as follows:

```
1: for each window( $i$ ) output do
2: Find  $L$  channels with largest amplitude :  $f_L(i)$ 
3:   for each  $f_n \in f_L$  do
4:     if  $\exists f_k \in f_L(i-1)$  s.t.
5:      $f_n - F_d/2 \leq f_k \leq f_n + F_d/2$  then
6:        $Amp(f_n(i)) \leftarrow Amp(f_k(i-1)) + \Delta$ 
7:     else
8:        $Amp(f_n(i)) \leftarrow 0$ 
9:     end if
10:  end for
11: end for
12: for  $j = 1:Ch\_num$  do
13:  $FeatureVector(j) \leftarrow \sum_i^N Amp(f(i,j))$  ( $N$ =No. of win-
    dows per frame)
14: end for
```

In this algorithm, first the  $L$  frequencies with largest amplitudes are selected from each window. Then the amplitude

<sup>1</sup>Jyotibdha Acharya is a PhD student in NITHM, Interdisciplinary Graduate School, Nanyang Technological University, Singapore JYOTIBDH001@e.ntu.edu.sg

<sup>2</sup>Arindam Basu and Wee Ser are faculties with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore {ARINDAM.BASU, ewser}@ntu.edu.sg

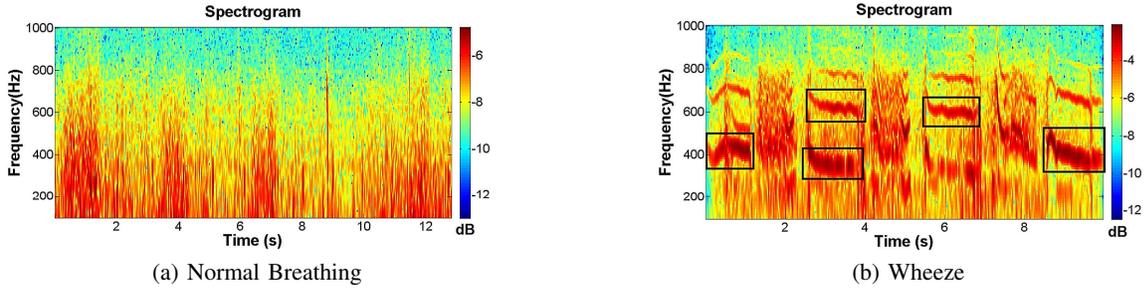


Fig. 1: Spectrogram of Breathing Sounds of Wheeze Patient and Normal Subject: frequency contours corresponding to wheeze are marked

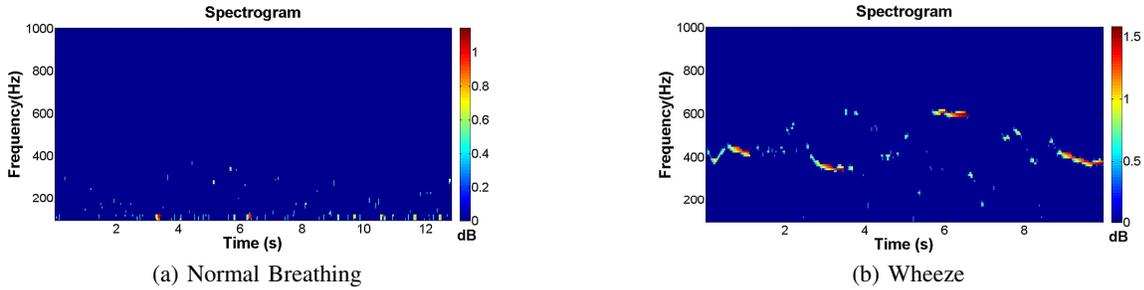


Fig. 2: Spectrogram of Breathing Sounds of Wheeze Patient and Normal Subject After applying FCT : Spectral lines corresponding to wheeze are amplified while noise is suppressed

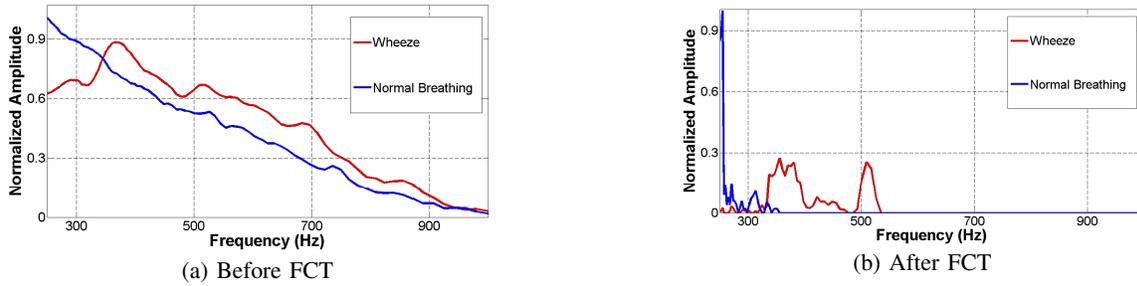


Fig. 3: Feature Vectors Computed Before and After Applying Feature Extraction Algorithm: Sharp peaks at frequencies corresponding to spectral contours are visible after FCT is applied

corresponding to a certain frequency channel (out of  $L$  frequencies) increases if there is any large amplitude frequency channel within its neighborhood in the previous time window and the amplitude becomes 0 otherwise. The feature vector is the sum of amplitudes corresponding to each frequency over the duration of a frame. Thus the length of the feature vector is equal to the number of frequency channels ( $ch\_num$ ). As the amplitude increases linearly ( $O(n)$ ) with the duration of continuous contours, the amplitude of the feature vector grows by  $O(n^2)$ . In Fig. 2 we can see the output spectrogram after applying the algorithm. It can be clearly seen that most of the noise points are removed while only frequency contours are present. Fig. 3 shows the sum of channel outputs (extracted feature) before and after applying the algorithm. Wheeze and normal signals show indistinguishable channel response before the algorithm is applied. But the feature vector corresponding to wheeze signal shows a clear peak corresponding to the location of spectral contours after the algorithm is applied. The trade-off between noise suppression and signal amplification is achieved by tuning algorithm

parameters  $F_d$  and  $L$ .

We checked the validity of the feature extraction on the dataset which contains breathing sounds of 6 normal subjects and 18 wheeze patients collected in practical environment (either in the clinic or a local hospital). IRB approval and informed consent of patients were obtained prior to the data collection. Breathing sounds were recorded over the right side of the chest using an acoustic sensor. The details of the database and data collection method are described in [6]. Each sample (duration : 9-12sec, re-sampled at  $4k Hz$ ) was divided into 3 sec frames and each frame was used to obtain one feature vector. We used a majority voting layer to obtain the sample accuracy from frame accuracies. The spectrograms were obtained using a 60 ms hanning window with 50% overlap.

For the binary classification task based on the extracted features, random forest (RF) algorithm has been used throughout this paper. For consistency, the size of the random forest is kept constant (50 trees) for all the experiments. 3-fold cross-validation was used to obtain the classification

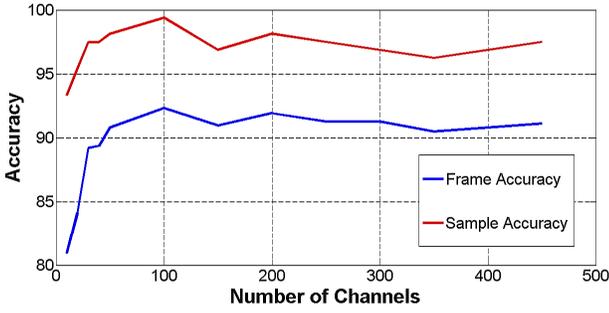


Fig. 4: Accuracy vs. No. of Frequency Channels: At least 50 Channels are required for reasonable accuracy

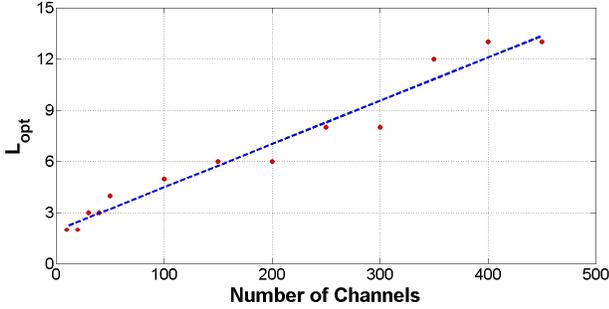


Fig. 5:  $L_{opt}$  vs. No. of Frequency Channels:  $L_{opt}$  is almost linearly dependent on No. of Channels

accuracies.

A design space exploration was done to determine optimum parameter settings of the algorithm. From Fig.4, we can infer that the accuracy reaches its peak value at  $Ch\_num = 100$  and then slightly decreases for larger  $Ch\_num$  values (larger  $Ch\_num$  results in larger feature vector size and therefore, higher classification complexity). Fig. 5 shows the optimum values of  $L$  as a function of  $Ch\_num$ . While the value of  $L$  increases almost linearly with number of channels, the maximum value of  $F_d$  is limited to 5. Higher value of  $F_d$  results in a significant increase in false contour detection.

### B. Hardware Method

We developed two low complexity variants of this algorithm for hardware implementation. Here instead of sorting the frequencies based on amplitudes and finding  $L$  largest components, we used thresholding to find the frequencies with amplitudes larger than a threshold, since it is computationally less expensive. We used two variants, one with fixed threshold (FT) and one with adaptive threshold (AT). The algorithm is as follows:

- 1: **for** each window( $i$ ) output **do**
- 2: Update Threshold. (Only for adaptive threshold)
- 3: Apply threshold.
- 4: **for** each channel( $j$ ) **do**
- 5:  $out(j) = out(j - F_d/2) + \dots + out(j) + \dots + out(j + F_d/2)$  (where '+' represents logical OR)
- 6: **if**  $out(j) = 1$  **then**
- 7:      $CounterOutput(j) \leftarrow CounterOutput(j) + 1$
- 8: **else**

	EBWD	2-D Threshold	Proposed SW Method	Proposed HW Method (FT)	Proposed HW Method (AT)
Frame Acc.	71.47	90.65	92.33	87.23	89.22
Sample Acc.	84.38	95.63	99.38	93.13	96.25

TABLE I: Accuracy Comparison (%)

- 9:      $CounterOutput(j) \leftarrow 0$
- 10: **end if**
- 11: **end for**
- 12: **end for**
- 13: Add Counter outputs till end of frame.

Each channel output is considered 1 if its amplitude is greater than the threshold and 0 if its threshold is less than the threshold. In adaptive thresholding, an additional adder is used to check the number of channel outputs above threshold and the threshold value is adjusted (multiplied or divided by a constant learning parameter) based on the value. We apply an array of OR gates to  $F_d$  consecutive channel outputs to determine if there is any high(1) amplitude frequency in the neighborhood. In the next stage the counter increments for OR gate output 1 and resets for OR gate output 0. Finally in the last stage an array of adders (one for each channel) sums up the outputs of each channel over a frame to produce the feature vectors. For  $Ch\_num = 100$ ,  $L = 10$  and  $F_d = 3$ , fixed threshold requires: 100 comparisons/window, 200 ORs/window, 15 counter increment and decrements/window (avg.) and 100 additions/window. Adaptive threshold method requires an additional 100 additions/window. As frame size is 3sec and one window output is generated each 30ms, the total no. of operations per frame is approximately 42kops for fixed thresholding and 52kops for adaptive thresholding.

### III. RESULTS

Table I compares the highest accuracies obtained by software and hardware methods. We also compared the performance of two earlier feature extraction methods 'EBWD' [4] and '2-D Threshold' [7] with our results since the datasets used in these papers are similar to ours. All the results were obtained using 3-fold cross-validation. The performance of our method is superior to that of EBWD while it is similar to '2-D Threshold' method. However, in our hardware methods, after the threshold is applied, the data is purely binary numbers while in 2-D threshold method, they are floating point numbers and complex functions like logarithm needs to be calculated. Thus, in a microcontroller based implementation our method requires  $3\times$  less computations and an ASIC based implementation requires  $6\times$  less computations. We checked the performance of the algorithm in presence of noise by adding white Gaussian noise to the signal as shown in Fig. 6. Here the algorithm parameters are set to  $Ch\_num = 100$ ,  $L = 5$ ,  $F_d = 3$ .

Methods	Proc.	Ops/frame	Proc. Power	Tx. bits/sec	Tx. Power	Remarks
I	-	-	-	$4k \times 16 = 64k$	2560 $\mu W$	-
II	FFT + FCT	$670k + 52k = 723k$	24 $\mu W$	$8 \times 100/3 \approx 267$	10.7 $\mu W$	Feature vec.=100 numbers (8-bit), Tx.once/frame
III	FFT +FCT +RF	$670k + 52k + 1k = 723k$	24 $\mu W$	1/3	13 $nW$	Binary Label Tx.once/frame.
IV	FFT + FCT + RF	$670k + 52k + 1k = 723k$	24 $\mu W$	$\alpha \times 4k \times 16 = \alpha \times 64k$	$\alpha \times 2560 \mu W$	$\alpha$ =frac. of frames with wheeze, e.g. $\alpha=0.3$ , Tx.power =768 $\mu W$

TABLE II: Power Requirement Comparison

Method I: Total Transmission, Method II: Processing + Transmit Feature vector, Method III: Processing+ transmit label, Method IV: Processing + transmit frames with wheeze.

Finally, compared to traditional record and transmit strategy (Method I), several on-node processing strategies can provide significant reduction in power consumption. The audio signals can be processed on wearable and feature vectors acquired using FCT can be transmitted (method II) or only the class labels obtained after classification can be transmitted (Method III). Alternatively, only the frames labeled as wheeze can be transmitted for further analysis (Method IV). Obviously, these methods present us with a trade-off between processing and transmission power and choice of optimum method depends on hardware and power constraints. Assuming the audio recording is sampled at  $4k Hz$  at a bit depth of 16 bit and frame size is 3sec, we compared the power requirement for arithmetic operations and transmission for these strategies (Table. II.). We used the specifications of *Apollo2 Ultra-Low Power Microcontroller* for calculating processing power ( $10\mu A/MHz$  at 3.3V [8]) and *Bluetooth low energy (BLE)* (power  $\approx$  bitrate  $\times$  40nJ/bit [9]) for calculating transmission power. We assume a trained random forest (with 50 trees and 20 avg. nodes/tree) is used for binary classification. We also assume each arithmetic operation requires 3 instruction cycles. The signal is analyzed using 256 point FFT (approx. 6.7K operations per FFT) with window size 60 ms and 50% overlap. With traditional methods (record and transmit), the transmission power is the main bottleneck. The use of feature extraction algorithm and on-device classification considerably reduces transmission power overhead and power consumption is primarily comprised of processing power which can be reduced sig-

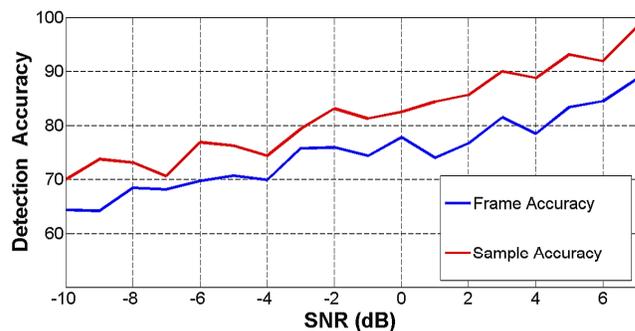


Fig. 6: Accuracy vs. SNR: Frame and Sample accuracies are 65% and 70% respectively for SNR=-10dB and increase to 89% and 99% respectively at SNR=7dB.

nificantly using power efficient microcontrollers. Moreover, as we can see from the Table. II the bulk of processing power is spent on computing the FFTs. This power can be reduced even further using low power digital filter banks.

#### IV. CONCLUSION

In this paper we have proposed a low complexity T-F continuity based feature extraction algorithm and its power efficient hardware implementation for wheeze detection. The algorithm produces highly distinguishable features for wheeze signals using nominal computational overhead. High classification accuracies were obtained for both software and hardware simulations. The method (coupled with a suitable classifier) can be used for low power implementation of both on-chip wheeze detection and selective transmission strategies. In future we plan to develop a low power wearable wheeze detection platform using microcontroller based implementation of the algorithm with commercially available audio recording devices .

#### REFERENCES

- [1] T. R. Fenton, H. Pasterkamp, A. Tal, and V. Chernick, "Automated spectral characterization of wheezing in asthmatic children," *IEEE Transactions on biomedical engineering*, no. 1, pp. 50–55, 1985.
- [2] M. Bahoura and C. Pelletier, "Respiratory sounds classification using gaussian mixture models," in *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 3. IEEE, 2004, pp. 1309–1312.
- [3] M. Wisniewski and T. P. Zielinski, "Application of tonal index to pulmonary wheezes detection in asthma monitoring," in *Signal Processing Conference, 2011 19th European*. IEEE, 2011, pp. 1544–1548.
- [4] J. Zhang, W. Ser, J. Yu, and T. Zhang, "A novel wheeze detection method for wearable monitoring systems," in *Intelligent Ubiquitous Computing and Education, 2009 International Symposium on*. IEEE, 2009, pp. 331–334.
- [5] S. A. Taplidou and L. J. Hadjileontiadis, "Wheeze detection based on time-frequency analysis of breath sounds," *Computers in biology and medicine*, vol. 37, no. 8, pp. 1073–1083, 2007.
- [6] J. Zhang, W. Ser, and D. Y. T. Goh, "A novel respiratory rate estimation method for sound-based wearable monitoring systems," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3213–3216.
- [7] X. Liu, W. Ser, J. Zhang, and D. Y. T. Goh, "Detection of adventitious lung sounds using entropy features and a 2-d threshold setting," in *Information, Communications and Signal Processing (ICIS), 2015 10th International Conference on*. IEEE, 2015, pp. 1–5.
- [8] "Apollo2 mcu ambiq micro," <http://ambiqmicro.com/apollo-ultra-low-power-mcu/apollo2-mcu/>, (Accessed on 01/11/2017).
- [9] N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Guttag, and A. P. Chandrakasan, "A micro-power eeg acquisition soc with integrated feature extraction processor for a chronic seizure detection system," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, 2010.