University of California

Los Angeles

Noise Robust Signal Processing for Human Pitch Tracking and Bird Song Classification and Detection

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Wei Chu

2012

The dissertation of Wei Chu is approved.

Daniel T. Blumstein

Mihaela van der Schaar

Kung Yao

Abeer Alwan, Committee Chair

University of California, Los Angeles

2012

To my mother Na Lin and my father Qingguo Chu

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost I would like to thank my advisor Professor Abeer Alwan. It has been an honor to be her Ph.D. student. She has taught me, both consciously and unconsciously, how to become a researcher with abilities and a human being with responsibilities. I appreciate all her contributions of time, ideas, and funding in making my Ph.D. experience productive and stimulating.

My sincere gratitude also goes to Professors Daniel Blumstein, Mihaela van der Schaar, and Kung Yao for being on my doctoral committee and for their interest in my research.

I would also extend my gratitude to Professors Charles Taylor and Daniel Blumstein for their valuable comments and funding support during the collaboration of the bird song project.

I am very thankful to Professor Bhiksha Raj, Dr. Bryan Pellom, Dr. Kadri Hacioglu, Dr. John McDonough, Dr. Ivan Tashev, and Dr. Mike Seltzer for being my mentors during my internships. The insightful and enlightening guidance I received from them have greatly benefited my research at UCLA.

I sincerely thank my Master thesis advisor Professor Jia Liu at Tsinghua University. He had shown me how perseverance and preciseness can act as important factors in successful speech research.

Thanks go to my labmates and friends, Professor Seiji Hayashi, Professor Kohichi Ogata, Dr. Qifeng Zhu, Dr. Jintao Jiang, Dr. Xiaodong Cui, Dr. Markus Iseli, Dr. Sankaran Panchapagesan, Dr. Ziad Al Bawab, Dr. Hong You, Dr. Bengt Jonas Borgstrom, Dr. Yen-Liang Shue, Dr. Shizhen Wang, Dr. Chanwoo Kim, Dr. Marek Vondrak, Dr. Xin Chen, Dr. Xiaodan Zhuang, Xin

Yan, Tao Hu, Dong Han, Ke Tan, Gang Chen, Lee Ngee Tan, Harish Arsikere, and many others for their help and friendship. I want to especially thank my senior labmate Shizhen for sparing his time discussing and verifying my ideas selflessly.

I am deeply in debt to my mother Na Lin and my father Qingguo Chu, who overcame uncountable difficulties in their lives to bring me to this world and up to a man. Without their unconditional love, support and encouragement, this dissertation work would not have been possible. I would like to dedicate this dissertation to them.

Vita

1981	Born, Dalian, Liaoning, China.
2000 - 2004	B.E. (Electronic Engineering), North China University of Technologies, China.
2004 - 2007	M.S. (Electrionic Engineering), Tsinghua University, China.
2007 - 2012	Ph.D. (Electrical Engineering), University of California, Los Angeles, USA.

Publications

W. Chu and A. Alwan, "SAFE: a statistical approach to F0 estimation under clean and noisy conditions," IEEE Trans. on Audio, Speech, and Language Processing, accepted.

W. Chu and D.T. Blumstein, "Noise robust bird song detection using syllable pattern-based hidden Markov models," ICASSP 2011, pp. 345-348.

W. Chu and A. Alwan, "SAFE: a statistical algorithm for F0 estimation for both clean and noisy speech," Interspeech 2010, pp. 2590-2593.

W. Chu and A. Alwan, "A correlation-maximization denoising filter used as

an enhancement frontend for noise robust bird call classification," Interspeech 2009, pp. 2831-2834.

W. Chu and A. Alwan, "Reducing F0 frame error of F0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," ICASSP 2009, pp. 3969-3972.

Abstract of the Dissertation

Noise Robust Signal Processing for Human Pitch Tracking and Bird Song Classification and Detection

by

Wei Chu

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Abeer Alwan, Chair

This dissertation investigates the extraction of discriminative information from noisy signals for human fundamental frequency (F0) tracking, and for bird song classification and detection.

For F0 tracking, the investigation is carried out in the direction of reducing F0 estimation and voicing decision errors.

To reduce F0 estimation errors, a novel Statistical Algorithm for F0 Estimation, SAFE, is proposed to improve the accuracy of F0 estimation under both clean and noisy conditions. Prominent Signal-to-Noise Ratio (SNR) peaks in speech spectra constitute a robust information source from which F0 can be inferred. A probabilistic framework is proposed to model the effect of noise on voiced speech spectra. Prominent SNR peaks in the low frequency band (0 - 1000 Hz) are important to F0 estimation, and prominent SNR peaks in the middle and high frequency bands (1000 - 3000 Hz) are also useful supplemental information to F0 estimation under noisy conditions, especially in the babble noise condition.

To reduce voicing decision errors, we introduce a model-based unvoiced/voiced

(U/V) classification frontend which can be used by any F0 tracking algorithm. We propose an F0 Frame Error (FFE) metric which combines Gross Pitch Error (GPE) and Voicing Decision Error (VDE) to objectively evaluate the performance of F0 tracking methods. A GPE-VDE curve is then developed to show the tradeoff between GPE and VDE.

For bird call classification, the investigation is carried out in the direction of signal denoising and discriminative feature extraction.

To enhance noisy signals, we propose a Correlation-Maximization denoising filter which utilizes periodicity information to remove additive noise in Antbird calls. We also develop a statistically-based noise-robust bird-call classification system which uses the denoising filter as a frontend. Enhanced bird calls which are the output of the denoising filter are used for feature extraction.

To obtain discriminative features, we extend the expectation-maximization (EM) algorithm to estimate not only optimal acoustic model parameters, but also optimal center frequencies and bandwidths of the filter bank used in cepstral feature extraction for bird call classification. The search is done using the gradient ascent method. Filter bank and model parameters are optimized iteratively.

For bird song detection, temporal, spectral, and structural characteristics of Robin songs and syllables are studied. Syllables in Robin songs are clustered by comparing a distance measure defined as the average of aligned Linear Predictive Coding (LPC)-based frame level differences. The syllable patterns inferred from the clustering results are used to improve acoustic modelling of a hidden Markov model (HMM)-based song detector.

CHAPTER 1

Introduction and Background

This dissertation studies aspects of pitch estimation and tracking in clean and in noisy signals, and of bird song classification and detection in noisy signals.

1.1 F0 Estimation and Tracking

The source-filter model of speech production [1] assumes that speech signals can be modeled as an excitation signal filtered by a linear vocal-tract transfer function. The fundamental frequency (F0) is defined as the inverse of the period of the excitation signal during the voicing state [2] [3]. Accurate F0 tracking in quiet and in noise is important for several speech applications, such as speech coding, analysis and recognition. In this dissertation, F0 estimation refers to estimating F0 values over voiced frames. F0 tracking refers to classifying voiced regions and estimating F0 values over those voiced regions.

1.1.1 F0 Estimation Methods

Some F0 estimation algorithms are based on the source-filter theory of speech production and estimate F0 for voiced speech segments. They assume that F0 is constant and the vocal tract transfer function is time invariant within a short period of time, e.g, a frame of 10-20 milliseconds. These algorithms usually have

two stages. The first stage consists of obtaining F0 candidates and the likelihood of voicing on a frame-by-frame basis. The second stage consists of using dynamic programming to decide the optimal F0 and voicing state for each frame.

The first stage can be classified into two categories: single-band and multi-band. In the single-band method, F0 candidates are extracted from one frequency band [2]. There are several methods to generate F0 candidates. SIFT [4] applies inverse filtering to voiced speech to obtain the excitation signal from which it estimates F0 by using autocorrelation. Cepstral-based methods (e.g., [5]) separate the excitation from the vocal tract information in the cepstral domain by using a homomorphic transformation; the interval to the first dominant peak in the cepstrum is related to the fundamental period. RAPT [6] and YAPPT [7] generate F0 candidates by extracting local maxima of the normalized cross correlation function which is calculated over voiced speech. Praat [8] calculates cross correlation or autocorrelation functions on the speech signal and regards local maxima as F0 hypotheses. TEMPO [9] obtains F0 candidates by evaluating the 'fundamentalness' of speech which achieves a maximum value when the AM and FM modulation magnitudes are minimized. YIN [10] uses the autocorrelation-based squared difference function and the cumulative mean normalized difference function calculated over voiced speech, with little post-processing, to acquire F0 candidates. Yegnanarayana et al. [11] obtain F0 candidates from exploiting the impulse-like characteristics of excitation in glottal vibrations. Finally, Le Roux et al. [12] simultaneously perform frame-wise F0 candidate generation and time-direction smoothing.

In the following, the details of the autocorrelation function in Praat [8], the cross-correlation function in RAPT [6], and the average magnitude difference function in YIN [10] are described.

1.1.1.1 Autocorrelation and Cross-Correlation Functions

Suppose there is a preprocessed acoustic signal $s[n]$. Given a frame shift of L and a frame length of N, the n_{th} sample in frame m is denoted by $s_m[n]$, i.e., $s[mL+n]$. P_m is the ground truth pitch period, in samples, of the frame m.

For the frame m, the autocorrelation function (ACF) is defined as

$$R_m^{\texttt{ACF}}[k] = \sum_{n=0}^{N-k-1} s_m[n]s_m[n+k], \quad k = 0, \cdots, N-1, \tag{1.1}$$

where k denotes the lag.

For a voiced frame, the ACF will have maxima at multiples of pitch periods. The pitch period in samples of the frame m is estimated as:

$$\hat{P}_m^{\texttt{ACF}} = \arg\max_k R_m^{\texttt{ACF}}[k], \quad k = k_{\min}, \cdots, k_{\max}-1, \tag{1.2}$$

where $k_{\min}$ and $k_{\max}$ are the minimum and maximum pitch periods in samples, respectively.

As the lag k increases, the number of the samples involved in calculating the ACF, i.e., $N-k-1$, decreases. To keep the number of samples in each calculation stable, the Cross Correlation Function (CCF) of the frame m is defined as:

$$R_m^{\texttt{CCF}}[k] = \sum_{n=0}^{N-1} s_m[n]s_m[n+k], \quad k = 0, \cdots, N-1, \tag{1.3}$$

The ACF can be normalized. The Normalized Autocorrelation Function (NACF) is defined as:

$$R_m^{\texttt{NACF}}[k] = \frac{R_m^{\texttt{ACF}}[k]}{\sqrt{\sum_{n=0}^{N-k-1} s_m^2[n] \sum_{n=0}^{N-k-1} s_m^2[n+k]}}, \quad k = 0, \cdots, N-1, \tag{1.4}$$

The pitch period estimation of a voiced frame using the CCF or NACF is similar to the ACF.

1.1.1.2 Average Magnitude Difference Function

For the frame m, the average magnitude difference function (AMDF) is defined as follows:

$$R_m^{\texttt{AMDF}}[k] = \sum_{n=0}^{N-1} [s_m[n+k] - s_m[n]]^2, \quad k = 0, \cdots, N-1. \tag{1.5}$$

The AMDF will have minima at multiples of the pitch period. The pitch period, in samples, of the frame m is estimated as:

$$\hat{P}_m^{\texttt{AMDF}} = \arg\min_k R_m^{\texttt{AMDF}}[k], \quad k = k_{\min}, \cdots, k_{\max} - 1. \tag{1.6}$$

The AMDF is less sensitive to amplitude changes compared to CCF [9].

1.1.1.3 Multiband Techniques

In the multi-band method, a decision module is usually used to reconcile F0 candidates generated from different bands. Gold and Rabiner [13] use measurements of peaks and valleys of voiced speech as input to six separate functions whose values are then processed by an F0 estimator to obtain F0 candidates. Lahat et al. [14] calculate autocorrelation functions of the spectral magnitudes in different bands and then obtain F0 candidates by evaluating the local maxima of the functions. Sha et al. [15] detect F0 candidates by minimizing the values of sinusoid-based error functions calculated on 4 frequency bands: 25-100, 50-200, 100-400, and 200-800 Hz. These multi-band methods focus mainly on the low frequency bands.

The multi-band approach has also been used to apply Licklider's pitch perception theory [16] to F0 estimation. The irregular excitation signal may cause voiced speech to be aperiodic in some frequency bands [17]. It is hypothesized

that the higher levels of auditory processing isolate groups of contiguous harmonics to infer the fundamental frequency from a selection of these groups. In this view, it is hypothesized that auditory nerves and the auditory brainstem are capable of using an autocorrelation mechanism to infer F0 over different frequency channels. de Cheveigne shows that integrating the values of AMDFs across different channels in the time domain can improve F0 estimation accuracy [18]. Wu et al. [19] used correlograms to select reliable frequency bands, modeled F0 dynamics using a statistical approach, and then searched for the optimal F0 contour in an HMM framework.

1.1.1.4 Noise Robust F0 Estimation

The above-mentioned F0 candidate generation methods can also be applied to noisy conditions. Krusback et al. [20] use an autocorrelation function with confidence measures. Shimamura et al. [21] proposed a weighted autocorrelation function. Abe et al. [22] use the instantaneous frequency spectrum to enhance harmonics and suppress aperiodic components, which improves F0 estimation accuracy. Liu et al. [23] use joint time-frequency analysis to obtain robust adaptive representation of the speech spectrum from which important harmonic structures can be extracted. Nakatani et al. [24] use dominance spectra based on instantaneous frequencies to evaluate the magnitudes of the harmonics relative to background noise, and estimate F0 using only the reliable harmonics. Deshmukh et al. [25] use an aperiodicity, periodicity, and pitch detector to generate F0 candidates by calculating the AMDFs over different frequency channels in the spectral domain.

1.1.2 Unvoiced/Voiced Decision Methods

Most F0-tracking algorithms, such as RAPT [6], TEMPO [9], Praat [8] make U/V decisions based on the values of energy-based or harmonic-based features exceeding certain thresholds or not. Under different noisy conditions, one has to adjust these thresholds carefully in order to avoid performance degradation.

For example, the calculations of voicing likelihoods in RAPT [6] and Praat [8] are described in the following.

Let $s_m[n]$ denote the n_{th} sample in the m_{th} frame of an acoustic signal. The frame length is N.

In RAPT, the normalized cross correlation function of the frame m denoted by $\phi_m[k]$ is calculated as:

$$\phi_m[k] = \frac{\sum_{n=0}^{N-1} s_m[n]s_m[n+k]}{\sqrt{\epsilon + \sum_{n=0}^{N-1} s_m^2[n] \sum_{n=0}^{N-1} s_m^2[n+k]}}, \quad k = 0, \cdots, N-1, \tag{1.7}$$

where ϵ is an additive constant.

The voicing likelihood of the frame m denoted by $d_m^{\texttt{RAPT}}$ is calculated as:

$$d_m^{\texttt{RAPT}} = \max_k \phi_m[k] + b, \quad k = k_{\min}, \cdots, k_{\max} - 1, \tag{1.8}$$

where b denotes the bias term in the voicing decision, $k_{\min}$ and $k_{\max}$ are the minimum and maximum pitch periods in samples, respectively.

In Praat, the autocorrelation-based function of a frame is calculated as:

$$R_m^{\texttt{Praat}}[k] = \frac{R_m^{\texttt{NACF}}[k]}{r^w[k]}, \quad k = k_{\min}, \cdots, k_{\max} - 1, \tag{1.9}$$

$r^w[k]$ is a window defined as:

$$r^w[k] = (1 - \frac{k}{N})(\frac{2}{3} + \frac{1}{3}\cos\frac{2\pi k}{N}) + \frac{1}{2\pi}\sin\frac{2\pi k}{N}, \tag{1.10}$$

Let M denote the number of frames in the utterance, if

$$\max_k R_m^{\texttt{Praat}}[k] < 0.4, \tag{1.11}$$

and

$$\max_k R_m^{\texttt{Praat}}[k] < 0.05 \max_{m,k} R_m^{\texttt{Praat}}[k], \quad , m = 0, \cdots, M-1, \tag{1.12}$$

then, the frame m is likely to be an unvoiced frame.

1.1.3 Postprocessing

Postprocessing is usually performed using dynamic processing. The objective of dynamic programming is to search for an F0 contour that minimizes an objective function. In RAPT [6], the objective function is defined as a summation of the frame-level local cost and transition cost functions given an F0 contour. The local cost function for a certain frequency at one frame is inversely related to the F0 likelihood value. The inter-frame F0 transition cost function is defined under 4 conditions: voiced-to-voiced (V→V), unvoiced-to-unvoiced (U→U), voiced-to-unvoiced (V→U), and unvoiced-to-voiced (U→V). In the V→V condition, the cost function is defined as an increasing function of inter-frame proportional frequency change, but allows for octave jumps at some specifiable cost. In the U→U condition, the cost function is defined as 0. In the V→U or U→V conditions, the cost function is defined as a combination of a spectral stationarity function and the inverse function of the Itakura distortion [26].

1.1.4 Error Metrics

Consider F0 tracking on an utterance of N frames shown in Fig. 1.1 where the F0 values of unvoiced frames are set to 0 Hz.

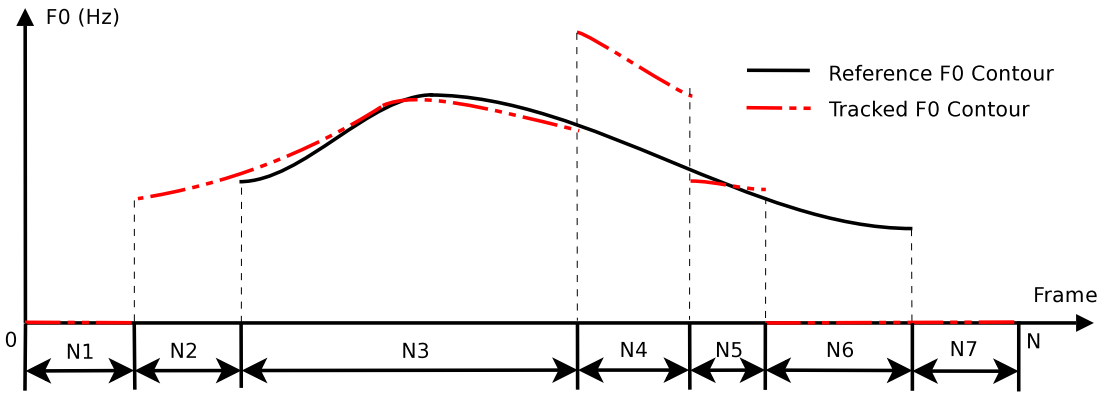

Figure 1.1: F0 Tracking Contour over Time for an utterance of N frames.

In F0 tracking, two types of error metrics are commonly used [2]. The first is Voicing Decision Error (VDE) [27]:

$$VDE = \frac{N_{V\to U} + N_{U\to V}}{N} \times 100\% \tag{1.13}$$

where N is the number of the frames in the utterance. The second is F0 value estimation error which is called the Gross Pitch Error (GPE):

$$GPE = \frac{N_{GE}}{N_{VV}} \times 100\% \tag{1.14}$$

where N_{VV} is the number of frames which both the F0 tracker and the ground truth consider to be voiced, N_{GE} is the number of frames for which

$$|\frac{F0_{i,estimated}}{F0_{i,reference}} - 1| > \delta\%, \quad i = 1, \cdots, N_{VV}, \tag{1.15}$$

where i is the frame index, and δ is a threshold which is typically 20.

In the example shown in Fig. 1.1,

$$\begin{aligned} VDE &= \frac{N_2 + N_6}{N} \times 100\%, \\ GPE &= \frac{N_4}{N_3 + N_4 + N_5} \times 100\%. \end{aligned} \tag{1.16}$$

Two other error metrics: Mean of the Fine Pitch Errors (MFPE) and Standard Deviation of the Fine Pitch Errors (SDFPE) [2], are used to measure the bias

and precision of the F0 estimation when no gross error occurs. The number of frames in which fine errors are made, denoted by N_{FE}, is equal to $N_{VV} - N_{GE}$. FE means 'fine error'. MFPE denoted by μ_{FPE} is defined as:

$$\mu_{FPE} = \frac{1}{N_{FE}} \sum_{i \in \mathcal{S}_{FE}} (f_{0i,estimated} - f_{0i,reference}), \tag{1.17}$$

where $\mathcal{S}_{FE}$ denotes the set of all the frames in which no gross error occurs.

SDFPE denoted by σ_{FPE} is defined as:

$$\sigma_{FPE} = \sqrt{\frac{1}{N_{FE}} \sum_{i \in \mathcal{S}_{FE}} f_{0i,estimated}^2 - \mu_{FPE}^2}, \tag{1.18}$$

In the example shown in Fig. 1.1, $N_{FE} = N_3 + N_5$.

1.2 Bird Song Classification and Detection

Bird songs are important in the communication between birds of specific species. A bird can listen to other birds and classify them as conspecific or heterospecific, neighbor or stranger, mate or non-mate, kin or non-kin [28]. It can also sing to other birds for mate attraction, danger alert, or territory defense [29]. Behavioral and ecological studies could benefit from automatically detecting and identifying species from acoustic recordings.

A denoising filter is usually needed to suppress the background noise and enhance the target bird call before extracting features from acoustic signals [30].

Different signal processing approaches are employed in analyzing the bird songs, such as Wigner-Ville distribution analysis [31], parametric representation [32], frequency component analysis [33], and so on. Researchers have also applied machine learning methods to bird song classification and recognition, such as back propagation and multivariate statistics [34], artificial neural networks [35],

evolving neural networks [36], dynamic time warping and hidden Markov models [37] [38], sinusoidal modeling of syllables [39], syllable pair histograms [40], and so on.

Methods for human speaker identification and speech recognition have also been applied to bird species identification and bird song recognition [41]. A typical bird call classification system usually includes feature extraction, acoustic model training and adaptation, and instance classification modules [42].

In feature extraction, audio signals are compressed to a sequence of feature vectors. When the distribution of the features is quantitatively modeled, the expectation-maximization (EM) algorithm can be used to estimate acoustic model parameters by iteratively maximizing the expectation of the likelihood from these features [43].

To improve the discriminability of the features, the original feature space can be mapped to new subspaces by certain projections. Different criteria are employed to search for optimal projections. Linear discriminant analysis (LDA) [44] computes the projection by maximizing the Fisher ratio value; heteroscedastic LDA (HLDA) [45] and multiple LDA (MLDA) [46] learn the projection by maximizing the likelihood from the transformed features; while fMPE [47] estimates the projection by minimizing phone error rate.

Changing parameters in feature extraction can also increase the discriminability of the features. The Mel-scaled filter bank is often used for feature extraction in automatic speech recognition (ASR) [48], although Graciarena et al. [49] manually changed the frequency range, the number of filters, and the frequency scale type of the filter bank for bird species identification.

1.3 Dissertation Outline

In Chapter 2, a statistical algorithm for F0 estimation under noisy conditions, SAFE, is proposed. In Chapter 3, a noise robust model-based unvoiced/voiced classifier is proposed to allow the SAFE algorithm to estimate F0 values on classified voiced regions. A new error metric for evaluating F0 tracking performance, F0 Frame Error, is also proposed to compare the performance of various F0 tracking algorithms in a unified framework. In Chapter 4, a Correlation-Maximization filter is proposed for acoustic signal denoising before extracting features for bird call classification. In Chapter 5, a filter bank Expectation-Maximization algorithm is proposed to improve the discriminability of the features extracted for bird call classification. In Chapter 6, a syllable pattern-based bird song detector is proposed for improving bird song detection in an audio stream. Finally, in Chapter 7, a summary of the dissertation is presented and ideas for future work are discussed.

Part I

Noise Robust F0 Estimation and Tracking

CHAPTER 2

SAFE: A Statistical Algorithm for F0 Estimation

In this chapter, voicing information is made available to F0 tracking algorithms. Therefore, it is possible to focus on reducing F0 estimation errors, i.e. GPEs, under noisy conditions.

According to the experimental results in this study, some of the publicly available methods, e.g. RAPT [6], TEMPO [9], Praat [8], can work well under relatively noise-free conditions. However, when the low-frequency band is contaminated by noise, an increase in F0 estimation errors is observed. Since it is possible that F0 harmonics in the middle or high frequency bands are not corrupted, it may be beneficial for an F0 estimation method to utilize these harmonics in determining F0. Current multi-band methods [14] [15] mainly retain F0 candidates obtained from the most reliable band, which is a 'hard-decision', while the Licklider's pitch perception model uses an empirically-based 'soft-decision' to merge the information from different bands [18]. Wu et al. [19] uses a 'soft-decision' approach to combine the information across bands. We propose a Statistical Algorithm for F0 Estimation (SAFE) which also utilizes a 'soft-decision' method. A data-driven approach is used to learn how the noise affects the amplitude and location of the peaks in the Signal-to-Noise Ratio (SNR) spectra of clean voiced speech. The likelihoods of F0 candidates are obtained by evaluating the peaks in

the SNR spectrum using the corresponding models learned from different bands. It is worth noting that Ying et al. [50] use a probabilistic method to estimate F0 distribution in order to avoid local optima in F0 estimation. Wang et al. [51] modeled the between-frame F0 transitions in a statistical approach to improve both F0 estimation and unvoiced/voiced decision.

In the following sections, the statistical effects of noise on clean voiced speech spectra are studied. This relationship between the noise and information source for F0 estimation is modeled in a probabilistic framework. In testing, the posterior probabilities of the F0 candidates are then calculated. In the experimental section, the performance of the proposed method under different noise types and SNRs is compared with prevailing F0 estimation methods.

A flowchart of SAFE is shown in Fig. 2.1. This chapter focuses on estimating fundamental frequency (F0) values over voiced frames that may be corrupted by quasi-stationary noise. Suppose that the range of F0 in human speech is from f_{0min} to f_{0max}, and the frequency resolution of F0 estimation is Δ. Then $\mathcal{S}_{\mathrm{F0}}$ is used to denote the set of all possible F0 values $\{f_{0min}, f_{0min}+\Delta, \cdots, f_{0max}\}$.

Given the power spectrum $\mathbf{Y}$ of a single observed noisy voiced frame under a stationary noise condition $\mathbf{N}$, the probability of f_0 being the fundamental frequency of that frame can be expressed as $P(f_0|\mathbf{Y},\mathbf{N})$. The most likely estimate, denoted by $\hat{f}_0$, should be:

$$\hat{f}_0 = \arg\max_{f_0 \in \mathcal{S}_{\mathrm{F0}}} P(f_0|\mathbf{Y},\mathbf{N}). \tag{2.1}$$

Let $\mathbf{Y}_f$ and $\mathbf{N}_f$ denote the power spectrum of the noisy voiced frame and noise at frequency f, respectively. Then the *a posteriori* SNR at frequency f denoted by γ_f is:

$$\gamma_f = 10\log_{10}\frac{\mathbf{Y}_f}{\mathbf{N}_f}. \tag{2.2}$$

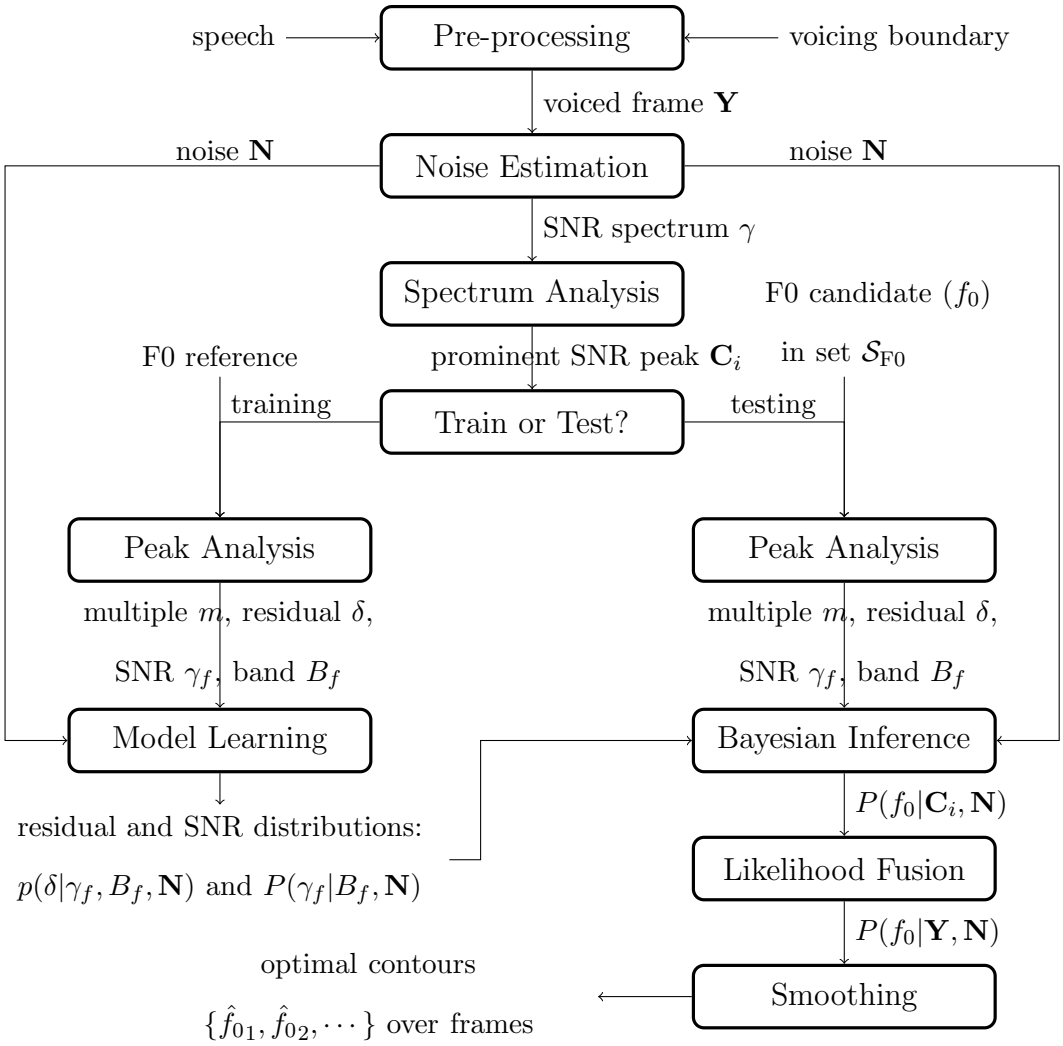

Figure 2.1: A flowchart of SAFE.

As quasi-stationary noise is assumed in this study, the noise spectrum for each utterance is estimated by averaging the initial 10 and final 10 frames of noisy speech. The frame shift is 10 ms, and the frame length is 40 ms.

The SNR γ_f is a measure of the spectral magnitude at frequency f being contaminated by noise. According to the source-filter theory of speech production, a voiced speech spectrum has a harmonic structure. Local SNR peaks (correspond to mainly harmonics) contain more information than valleys regarding F0. It is assumed that the information contained in the set of local SNR peaks $\{\mathbf{C}_1, \cdots, \mathbf{C}_M\}$ is sufficient to estimate F0, where M is the number of local SNR peaks. Thus, the posterior probability of f_0 is:

$$P(f_0|\mathbf{Y}, \mathbf{N}) = P(f_0|\mathbf{C}_1, \cdots, \mathbf{C}_M, \mathbf{N}). \tag{2.3}$$

In a ROVER system for automatic speech recognition [52], the posterior probabilities of a word from different sub-systems are combined with different weights. Inspired by ROVER, local SNR peaks can be assumed to be independent in inferring F0 given the noise shape and level. The overall posterior probability can be approximated as a weighted combination of posterior probabilities $P(f_0|\mathbf{C}_i, \mathbf{N})$:

$$P(f_0|\mathbf{Y}, \mathbf{N}) \approx \sum_{i=1}^{M} w_i P(f_0|\mathbf{C}_i, \mathbf{N}), \tag{2.4}$$

where w_i is the confidence measure of the i-th local SNR peak. If each local SNR peak is assumed to have an equal confidence score, then w_i is set to $1/M$. $(i = 1, 2, \cdots, M)$.

If the distribution of f_0 given the noise, i.e., $P(f_0|\mathbf{N})$, is assumed to be uniform when prior information is not available, then $P(f_0|\mathbf{C}_i, \mathbf{N})$ can be obtained from the Bayesian rule:

$$P(f_0|\mathbf{C}_i, \mathbf{N}) = \frac{p(\mathbf{C}_i|f_0, \mathbf{N})}{\sum_{f_0 \in \mathcal{S}_{\mathrm{F0}}} p(\mathbf{C}_i|f_0, \mathbf{N})}. \tag{2.5}$$

Let f denote the frequency of the local SNR peak $\mathbf{C}_i$. Because f is not usually equal to a multiple of f_0, f can be decomposed into a multiple m and a residual δ as follows:

$$m = \left[\frac{f}{f_0}\right], \quad \delta = \frac{f}{f_0} - m, \tag{2.6}$$

where $[\frac{f}{f_0}]$ denotes the nearest integer of $\frac{f}{f_0}$. Hence, the residual ranges from -0.5 to 0.5. If the fraction of $\frac{f}{f_0}$ is exactly 0.5, either rounding upwards or downwards does not change F0 estimation error rates in SAFE.

Given f_0 and noise $\mathbf{N}$, the local SNR peak $\mathbf{C}_i$ has the following attributes: multiple m, residual δ, *a posteriori* SNR γ_f, and frequency band index B_f in which the frequency f is. In other words, the peak $\mathbf{C}_i$ resides in band B_f. The reason why f is not adequate on its own is because there are not enough training samples for each frequency bin. Then we have:

$$\begin{aligned} p(\mathbf{C}_i|f_0,\mathbf{N}) &= p(m,\delta,\gamma_f,B_f|f_0,\mathbf{N}) \\ &= P(m|f_0,\mathbf{N})p(\delta|m,\gamma_f,B_f,f_0,\mathbf{N}) \\ &\quad p(\gamma_f|m,B_f,f_0,\mathbf{N})P(B_f|m,f_0,\mathbf{N}). \end{aligned} \tag{2.7}$$

We assume that the deviation of a local SNR peak from a multiple of f_0, caused by noise, will not exceed half f_0. Therefore, m is independent of the noise $\mathbf{N}$, i.e., $P(m|f_0,\mathbf{N}) = P(m|f_0)$. After the decomposition shown in Eq. 2.6, the residual δ can be assumed to be independent of m and f_0 given γ_f, B_f, and $\mathbf{N}$, i.e., $p(\delta|m,\gamma_f,B_f,f_0,\mathbf{N}) = p(\delta|\gamma_f,B_f,\mathbf{N})$. The local SNR γ_f is independent of m and f_0 given the band index B_f and noise condition $\mathbf{N}$, i.e., $p(\gamma_f|m,B_f,f_0,\mathbf{N}) = p(\gamma_f|B_f,\mathbf{N})$. Furthermore, $P(m|f_0)$ is assumed to be uniformly distributed. Since B_f can be assumed to be determined by m and f_0 regardless of noise, the Dirac

function $P(B_f|m, f_0, \mathbf{N})$ is assumed to be equal to 1. Then we can have:

$$\begin{aligned} & p(\mathbf{C}_i|f_0, \mathbf{N}) \\ = \;& D_1 \cdot p(\delta|\gamma_f, B_f, \mathbf{N}) p(\gamma_f|B_f, \mathbf{N}). \end{aligned} \tag{2.8}$$

where D_1 is a constant.

2.1 Prominent SNR Peaks

Before studying the distribution of the residual and local SNR peaks, it is important to select useful local SNR peaks for F0 estimation. Short and long-term smoothed SNRs denoted by $\gamma_f^{\mathtt{S}}$ and $\gamma_f^{\mathtt{L}}$ are obtained by smoothing γ_f with a Hamming window of length f_{0min} and f_{0max} in Hz, respectively. The Hamming window is used because of its relatively small side lobes. Since the short-term smoothing can reduce the number of false alarm local SNR peaks and retain F0 information, γ_f in Eq. 2.8 is replaced by $\gamma_f^{\mathtt{S}}$. To depict the relationship between the two smoothed SNRs, an SNR difference at the i-th local peak in $\gamma_f^{\mathtt{S}}$ denoted by ζ_i can be expressed as follows:

$$\zeta_i = \gamma_{f_i}^{\mathtt{S}} - \gamma_{f_i}^{\mathtt{L}}, \quad i = 1, \cdots, M^{\mathtt{S}}, \tag{2.9}$$

where $M^{\mathtt{S}}$ is the number of the local peaks in $\gamma_f^{\mathtt{S}}$. ζ_i is further normalized with respect to all the peaks in the frame as follows:

$$\bar{\zeta_i} = \frac{\zeta_i - \mu_\zeta}{\sigma_\zeta}, \quad i = 1, \cdots, M^{\mathtt{S}}, \tag{2.10}$$

where μ_ζ and σ_ζ are the mean and standard deviation of the sequence ζ_i. The i_{th} local SNR peak ($\mathbf{C}_i$) is regarded as a *prominent SNR peak* for F0 estimation only if $\bar{\zeta_i}$ is above a certain threshold. In this study, the threshold is empirically set to 0.33.

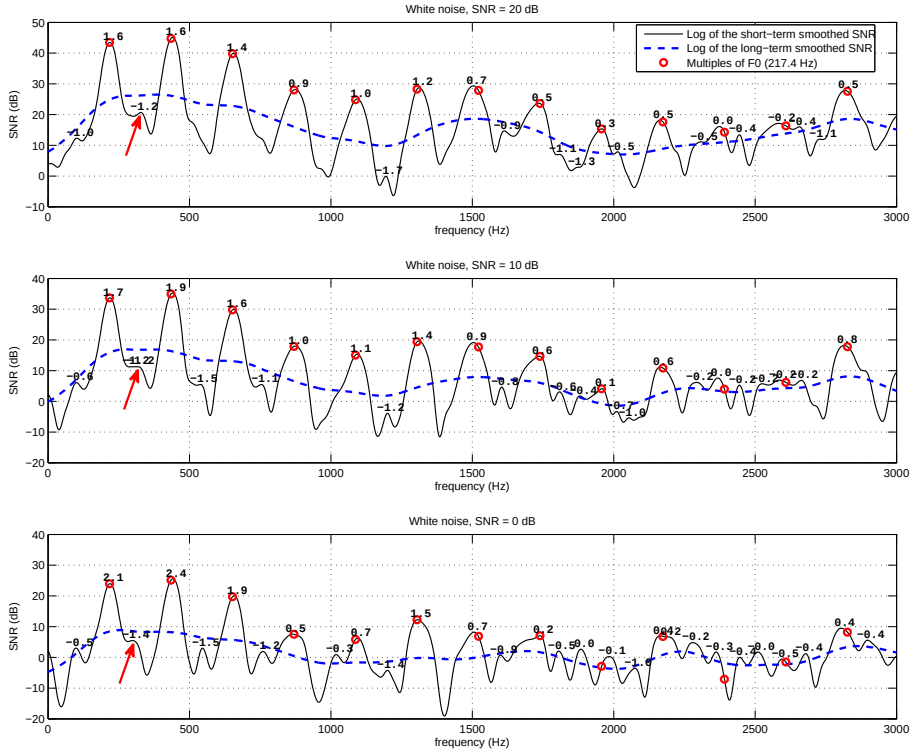

Figure 2.2: The SNR spectrum of a voiced frame of a female speaker corrupted by different levels of additive white noise (20, 10 and 0 dB). The number on top of each peak of the short-term smoothed SNR is the value of the normalized difference SNR $\bar{\zeta}_i$ of that peak. Arrows around 300 Hz indicate peaks with a lower $\bar{\zeta}_i$ than their adjacent prominent peaks.

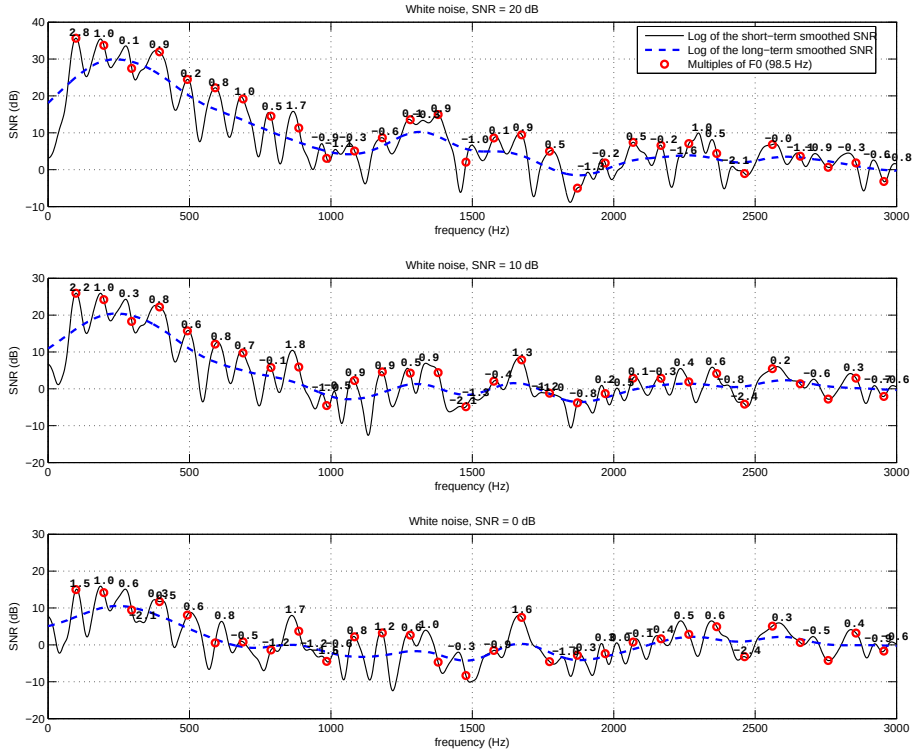

Figure 2.3: The SNR spectrum of a voiced frame of a male speaker corrupted by different levels of additive white noise (20, 10 and 0 dB). The number on top of each peak of the short-term smoothed SNR is the value of the normalized difference SNR $\bar{\zeta}_i$ of that peak.

Figs. 2.2 and 2.3 show the SNR spectra of a voiced frame of a female and a male speaker, respectively, corrupted by different levels of additive white noise (20, 10 and 0 dB). The number on top of each peak of the short-term smoothed SNR is the value of the normalized difference SNR $\bar{\zeta}_i$ of that peak. It can be seen that not all local SNR peaks reside in the vicinity of multiples of F0. Most false alarm or deviated peaks have a lower normalized SNR difference compared to the peaks near the multiples of F0. Take the false alarm local peaks around 300 Hz of the voiced frames in all panels of Fig. 2.2 for example. These peaks, indicated by arrows, have a lower $\bar{\zeta}_i$ than their adjacent prominent peaks in the three noise conditions.

As shown in Figs. 2.2 and 2.3, the lower a peak is compared to the long-term smoothed SNR, the more likely it is corrupted by the noise and shifted from its original location, and the less likely it is to be close to multiples of F0. Hence, prominent SNR peaks which are less corrupted by noise and less deviated from a multiple of F0 can provide reliable information for inferring F0s. When middle and high frequency bands are less corrupted by noise, it is possible that prominent peaks can exist in these bands, e.g., the peaks around 2800 Hz in female voiced frames and the peaks around 1700 Hz in male voiced frames under 20, 10 and 0 dB SNR conditions. Retaining these prominent peaks and discarding non-prominent peaks might improve the performance of F0 estimation.

As mentioned above, the higher a peak is, compared to long-term smoothed SNR, the more important it is in F0 inference. Therefore, the normalized SNR difference can be used for deciding the weights in Eq. 2.4 using a logistic regression function:

$$w_i = \frac{\dfrac{1}{1+\alpha e^{-\beta\bar{\zeta}_i}}}{\sum_{j=1}^{M^{\mathrm{s}}} \dfrac{1}{1+\alpha e^{-\beta\bar{\zeta}_j}}}, \quad i = 1, \cdots, M^{\mathrm{s}}. \tag{2.11}$$

where α and β are empirically set to 1.0 and 0.33.

As mentioned above, only prominent peaks are used in Eq. 2.4, i.e., M is changed to the number of prominent SNR peaks M^{S}.

2.2 Distribution of the Residuals

Recall that the residual δ is dependent on the local SNR value and the band index. To reduce the model complexity, it can be assumed that the distribution of the $p(\delta|\gamma_f, B_f, \mathbf{N})$ in Eq. 2.8 slightly changes when γ_f is rounded, i.e.,

$$p(\delta|\gamma_f, B_f, \mathbf{N}) \approx p(\delta|Q_{\gamma_f}, B_f, \mathbf{N}), \tag{2.12}$$

where Q_{γ_f} denotes the SNR bin which γ_f is rounded to. The intervals of the SNR bins in dB are spaced by 3.33 dB and are as follows: (-∞, 0], (0, 3.33], $\cdots$, (66.67, 70], (70, ∞).

The distributions of the residuals given different rounded SNR bins, frequency band index and noise conditions are shown in Fig. 2.4. Two white noise conditions: 20 and 0 dB SNRs are studied. This analysis is conducted over all the voiced frames in the KEELE corpus [53] with F0 ground truth values obtained from the simultaneously recorded laryngograph signal. In this study, three bands: 0-1000 Hz, 1000-2000 Hz, and 2000-3000 Hz, are employed to represent the low, middle, and high frequency bands, respectively. Note that all the residual distributions in Fig. 2.4 are derived only from the prominent peaks in the low frequency band. Most distributions are centered on zero, which means that these peaks can generate unbiased F0 estimates. It can be seen that under a certain noise condition, the higher the rounded SNR is, the smaller the variance of the residuals. Because having a smaller residual variance means that the frequencies of local SNR peaks are less likely to be affected by noise, local SNR peaks from higher

SNR bins are more reliable for F0 estimation. Under 20 dB conditions, no prominent peak has a local SNR higher than 56.67 dB; under 0 dB condition, the local SNRs of all prominent peaks are below 36.67 dB.

A comparison of the distributions of the residuals of the prominent and non-prominent peaks is shown in Fig. 2.5 for the white noise condition with 0 dB SNR. In the low frequency band, prominent peaks can have a local SNR as high as 36.67 dB, while the local SNRs of non-prominent peaks are below 26.67 dB. Furthermore, the residuals of the non-prominent peaks with low local SNRs are mostly distributed away from zero, which means that it is difficult to infer F0 from these non-prominent peaks. Although the residuals of the non-prominent peaks with high local SNRs are distributed around zero, the distributions have larger variances compared to the residuals of the prominent peaks with the same local SNR.

Curve-fitting or Gaussian mixture modeling can be used to model the distributions of the residuals; however, it is important to control the number of parameters in the model which enables training with limited data and prevent model over-fitting. A *Doubly truncated Laplacian distribution*, denoted by $p(\delta|\mu, b)$, is used for modeling $p(\delta|Q_{\gamma_f}, B_f, \mathbf{N})$, i.e. the distribution of residuals given the rounded SNR bin, band index and noise condition:

$$p(\delta|\mu, b) = \begin{cases} \frac{A}{2b} e^{-\frac{|\delta - \mu|}{b}} & -\frac{1}{2} \leq \delta \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}, \tag{2.13}$$

where μ and b represent the mean and the variance, respectively. A is set to $(1 - e^{-1/(2b)})^{-1}$ to ensure that $\int_\delta p(\delta|\mu, b) = 1$. Hence, only two free parameters (μ, b) need to be estimated.

Given a sequence of residuals $\{\delta_1, \cdots, \delta_N\}$ denoted by $\boldsymbol{\delta}$, (suppose all the

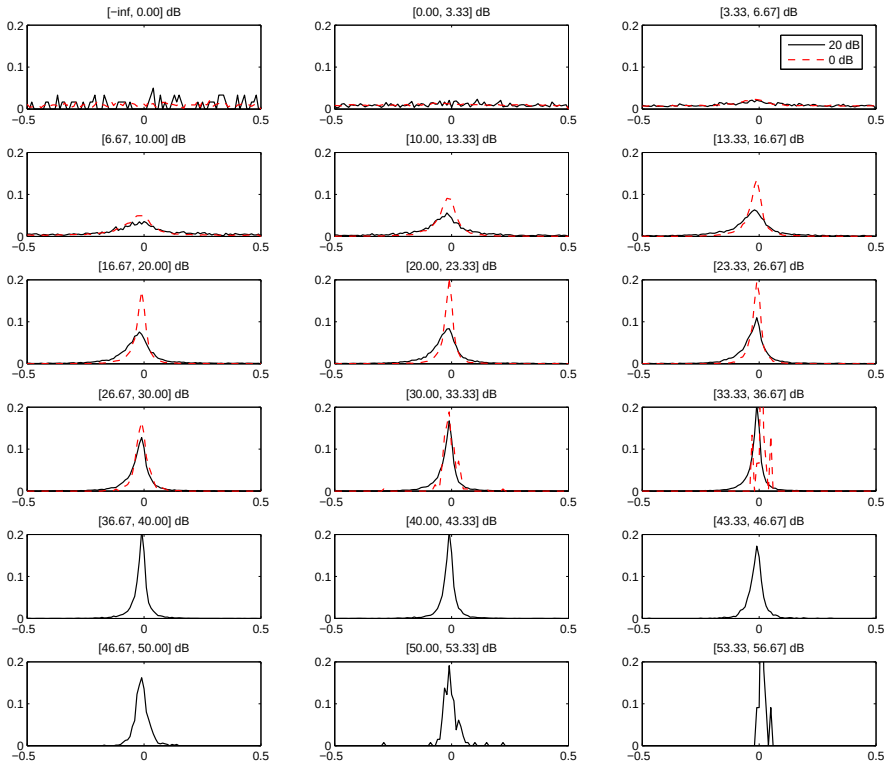

Figure 2.4: The distributions of the residuals given different rounded local SNRs for a 3.33 dB interval at the low frequency band (0-1000Hz). Different white noise conditions (20 and 0 dB global SNRs) are shown. The horizontal axes are the residuals with a bin size of 0.01. The vertical axes are the probabilities of occurrences. The title on each sub-figure shows the interval of rounded local SNR Q_{γ_f}.

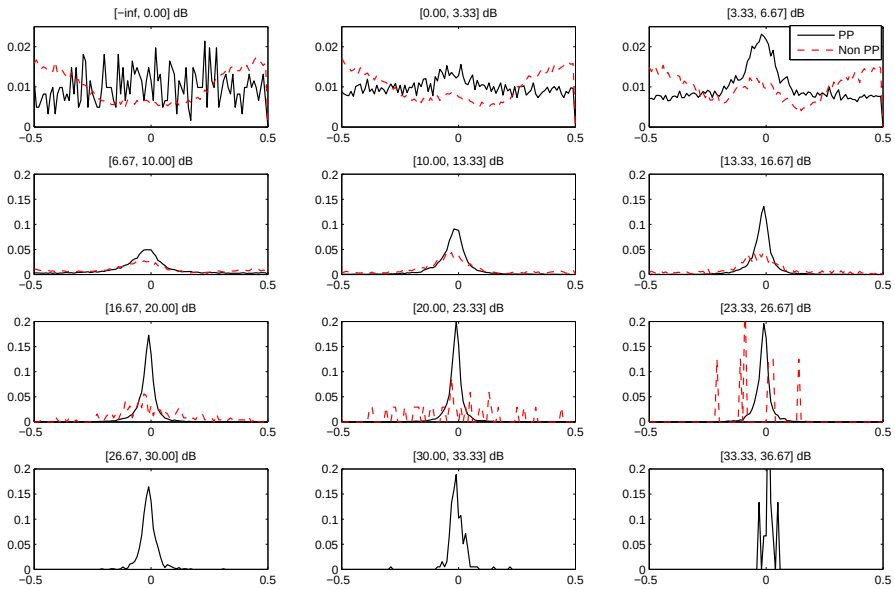

Figure 2.5: A comparision of the distributions of the residuals of prominent SNR peaks (PP) and non-prominent SNR peaks (Non PP) given different rounded local SNRs at the low frequency band (0-1000Hz). The noise condition is white noise at 0 dB global SNR. The horizontal axes are the residuals with a bin size of 0.01. The vertical axes are the probabilities of occurrences. The title on each sub-figure shows the interval of rounded local SNR Q_{γ_f}.

residuals are independent and identically distributed,) we have:

$$p(\boldsymbol{\delta}|\mu, b) = \prod_{i=1}^{N} p(\delta_i|\mu, b). \tag{2.14}$$

Let $\alpha = 1/(2b)$ and $\mathcal{L}(\boldsymbol{\delta}|\mu, \alpha) = \log p(\boldsymbol{\delta}|\mu, b)$. Then:

$$\begin{aligned} & \mathcal{L}(\boldsymbol{\delta}|\mu, \alpha) \\ = & \sum_{i=1}^{N} \log p(\delta_i|\mu, b) \\ = & N \log \alpha - N \log(1 - e^{-\alpha}) - 2\alpha \sum_{i=1}^{N} |\delta_i - \mu|. \end{aligned} \tag{2.15}$$

Under the maximum-likelihood criterion, the estimated mean and variance denoted by $\hat{\mu}$ and $\hat{b}$ (or $\hat{\alpha}$) should maximize the joint probability $p(\boldsymbol{\delta}|\mu, b)$ which is equivalent to maximizing $\mathcal{L}(\boldsymbol{\delta}|\mu, \alpha)$.

Since $\partial^2\mathcal{L}/\partial\mu^2 = -2\alpha \sum_{i=1}^{N} \delta(\delta_i - \mu) \leq 0$ when $\alpha > 0$, $\mathcal{L}$ achieves its maximum when $\partial\mathcal{L}/\partial\mu = 0$ for any α, i.e.:

$$-2\alpha \sum_{i=1}^{N} \operatorname{sgn}(\delta_i - \hat{\mu}) = 0. \tag{2.16}$$

Since $\partial^2\mathcal{L}/\partial\alpha^2 = 1/(e^{\alpha} - 1) - 1/\alpha^2 < 0$ when $\alpha > 0$, $\mathcal{L}$ achieves its maximum when $\partial\mathcal{L}/\partial\alpha = 0$ and $\mu = \hat{\mu}$, i.e.:

$$\frac{N}{\hat{\alpha}} - \frac{N}{e^{\hat{\alpha}} - 1} - 2 \sum_{i=1}^{N} |\delta_i - \hat{\mu}| = 0. \tag{2.17}$$

To solve $\hat{\mu}$, let $\tilde{\boldsymbol{\delta}} = \{\tilde{\delta}_1, \cdots, \tilde{\delta}_N\}$ denote the sorted sequence of the sequence $\boldsymbol{\delta}$ in an ascending order, we have one feasible solution of $\hat{\mu}$:

$$\hat{\mu} = \begin{cases} \tilde{\delta}_{\frac{N+1}{2}} & N \text{ is odd} \\ \frac{1}{2}(\tilde{\delta}_{\frac{N}{2}} + \tilde{\delta}_{\frac{N}{2}+1}) & N \text{ is even} \end{cases}. \tag{2.18}$$

Note that when N is even, any value between $\tilde{\delta}_{\frac{N}{2}}$ and $\tilde{\delta}_{\frac{N}{2}+1}$ can satisfy Eq. 2.16. Eq. 2.18 guarantees that the number of residuals that are greater than $\hat{\mu}$ is equal to the number of residuals that are less than $\hat{\mu}$. Eq. 2.17 can be then simplified as:

$$\frac{N}{\hat{\alpha}} - \frac{N}{e^{\hat{\alpha}} - 1} - 2\sum_{i=1}^{N} |\delta_i| = 0. \tag{2.19}$$

Although there is no close-form solution to Eq. 2.17, Newton's method can be used to search for $\hat{\alpha}$. Note that $\hat{b} = 1/(2\hat{\alpha})$. When a bin with a high rounded SNR does not have training instances, the mean and variance are not estimated. In testing, if the SNR peaks have higher values than the training set, the mean is set to 0, and the variance is set to a small value, e.g., 0.01.

There is one similarity between SAFE and Wu et al.'s method [19]: the use of Laplacian distribution for data modeling. The meaning and range of the modeled random variables are different. SAFE models the residual derived from the prominent peak in the SNR spectrum. The residual ranges from -0.5 to 0.5. Wu et al.'s method models the time lag derived from the peak in the correlogram. The time lag ranges from $-\infty$ to ∞.

The logarithms of the averaged estimated variances of the residual distributions for different bands are shown in Fig. 2.6. Averaging is across all noise levels: clean, 20 dB, 10 dB, 5 dB, 0 dB, -5 dB. The noise type is white noise. It can be seen that the variance of the lower frequency band at a certain rounded SNR bin is smaller than the variance of the higher frequency band. When the variance of the estimated residual distribution is small given a frequency band, it means that the probability of accurately estimating F0 in that band is high. As mentioned above, it is still possible to use the prominent peaks lying in the middle and high frequency bands to improve F0 estimation. Note that the higher the rounded local SNR, the smaller the variance is.

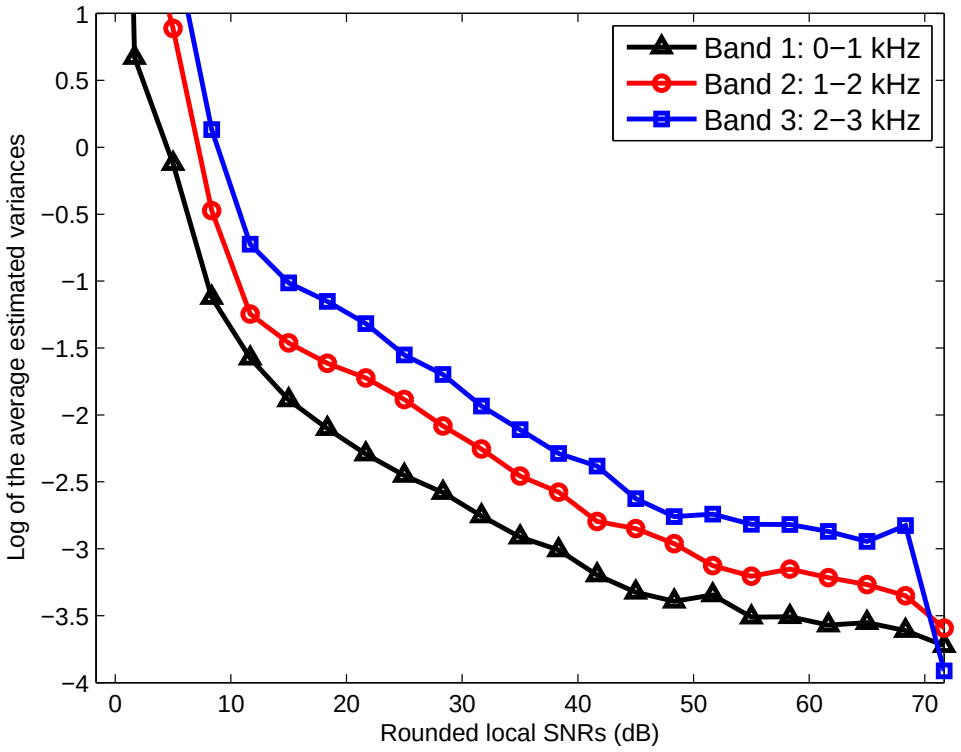

Figure 2.6: A comparison of the log of the averaged estimated variances of the residual distributions under different frequency bands (low, middle, high). The noise condition is white noise. Estimated variances from different noise levels (clean, 20 dB, 10 dB, 5 dB, 0 dB, -5 dB) are averaged.

In Fig. 2.7, the estimated means of the residual distributions for different bands under clean and noisy conditions are compared. The noise types are white and babble noise. The means under noisy conditions at different SNRs (20 dB, 10 dB, 5 dB, 0 dB, -5 dB) are averaged. It can be seen that the estimated means are not exactly equal to zero under both clean and noisy conditions if local SNR is less than 55 dB. F0 estimation actually benefits from learning a Laplacian distribution with a non-zero mean which better fits the real distribution of the data.

2.3 Distribution of the local SNRs

In the previous section, local SNRs of the prominent peaks are rounded. It can be assumed that this rounding does not significantly change the $p(\gamma_f|B_f, \mathbf{N})$ in Eq. 2.8, i.e.:

$$p(\gamma_f|B_f, \mathbf{N}) \approx D_2 P(Q_{\gamma_f}|B_f, \mathbf{N}), \tag{2.20}$$

where D_2 is a constant. The distribution can be learned by using a histogram-like approach based on the training set.

The distributions of the rounded local SNRs of the prominent peaks under different bands and noise conditions are shown in Fig. 2.8. The distribution under noisy conditions at different SNRs (20 dB, 10 dB, 5 dB, 0 dB, -5 dB) are averaged. It can be seen that the peaks of noisy speech are more likely to be distributed in bins with low SNRs compared to clean speech, which can be one of the reasons why estimating F0 values is difficult under noisy conditions. For either clean or noisy condition, the rounded local SNRs of the prominent peaks from the low frequency band are also more likely to be concentrated in high SNR bins compared to the middle and high frequency bands.

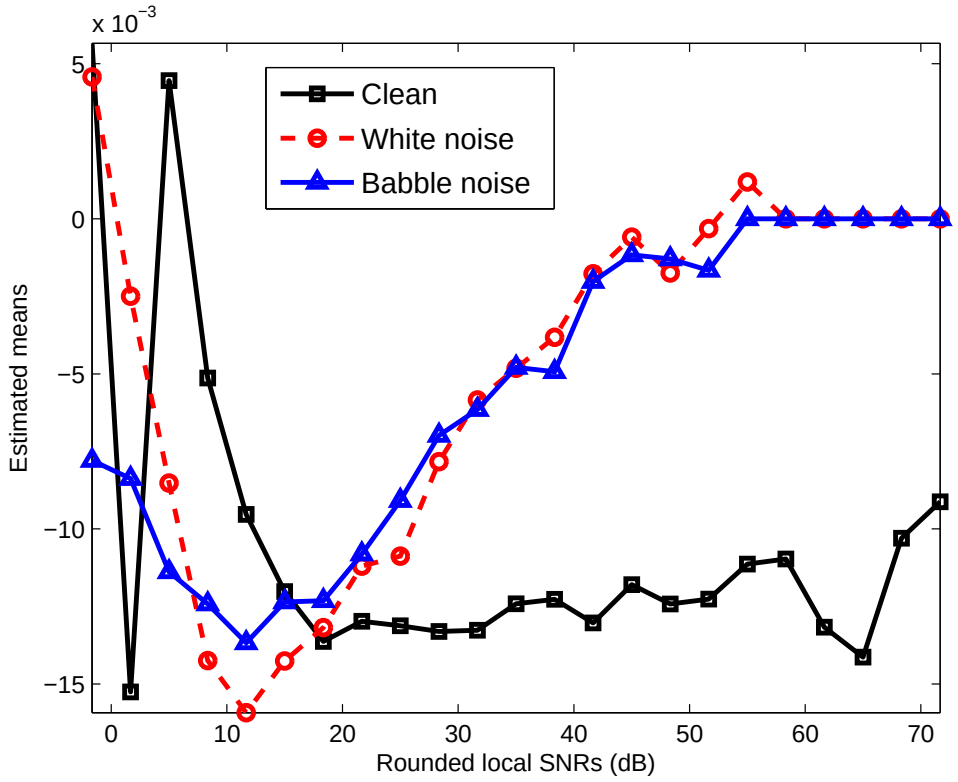

Figure 2.7: A comparison of the averaged estimated means of the distributions of residuals under different noise conditions using the KEELE corpus. Estimated means from different noise levels (clean, 20 dB, 10 dB, 5 dB, 0 dB, -5 dB) and different frequency bands (low, middle, high) are averaged.

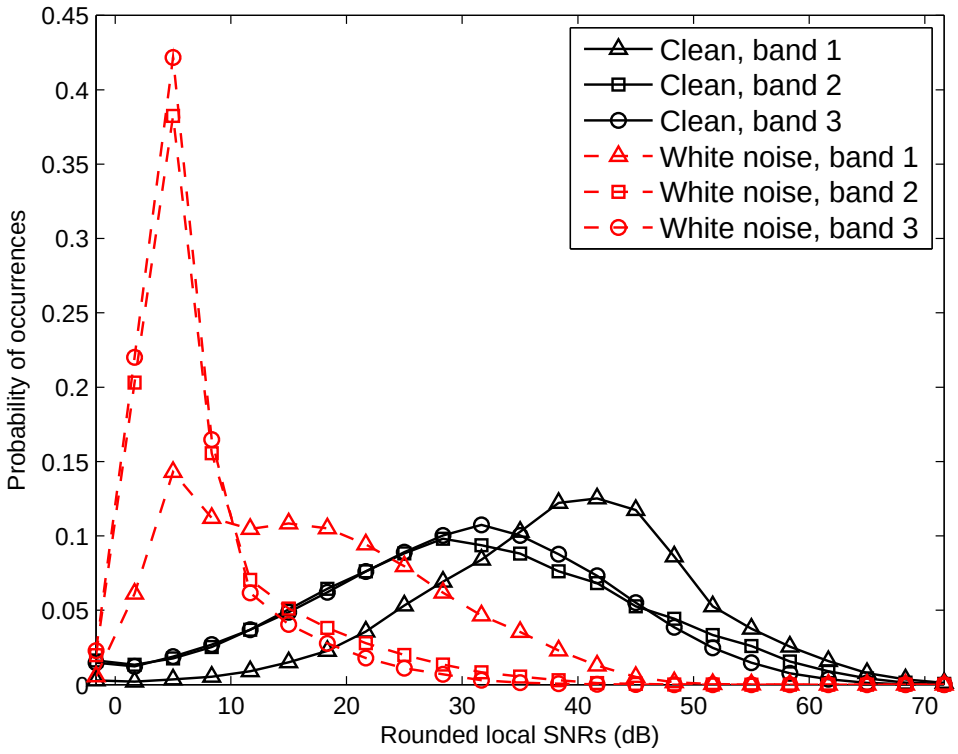

Figure 2.8: A comparison of the distributions of rounded local SNRs under different frequency bands (low-1, middle-2, high-3). The noise condition is white noise. The distributions under different noise levels (20 dB, 10 dB, 5 dB, 0 dB, -5 dB) are averaged.

2.4 Post-Processing

For an utterance, the posterior probabilities, $P(f_0|\mathbf{Y},\mathbf{N})$, for each frame are obtained by calculating Eq. 2.4. Then, a dynamic programming approach, the same as that used in RAPT, was used to smooth the tracked F0 contour and to allow octave jumps at a certain cost [6].

The focus of the proposed method is to reduce F0 estimation error under both clean and noisy conditions. However, voicing boundaries can affect the results of F0 tracking [54]. Hence, each F0 tracking algorithm is forced to estimate F0 values over all the voiced frames regardless of the SNRs.

The F0 trackers (RAPT, Praat, TEMPO, WWB) also output voiced/unvoiced decisions. If the ground truth and the F0 tracker agree that a frame is voiced or unvoiced, the F0 value is not changed. If a ground truth unvoiced frame is assumed to be voiced, the F0 value is set to be 0. If a ground truth voiced frame N_c is assumed to be unvoiced, f_{0N_c} is estimated by using an interpolation-based method:

$$f_{0N_c} = f_{0N_l} + \frac{N_c - N_l}{N_r - N_l}(f_{0N_r} - f_{0N_l}), \tag{2.21}$$

where N_l and N_r denote the left and right closest frame to the current frame N_c among the frames that both the ground truth and F0 tracker agree to be voiced. One exception of this interpolation is that if frame N_c is in the first or last assumed unvoiced segment by the F0 tracker in a ground truth voiced segment, the f_{0N_c} is set to be either f_{0N_r} or f_{0N_l} depending on whether the right or left frame is closer.

An example of F0 estimation made with SAFE is shown in Fig. 2.9. The segment corresponds to the beginning of the utterance of the second female speaker in the KEELE corpus. The noise condition is babble noise at 0 dB SNR. Each

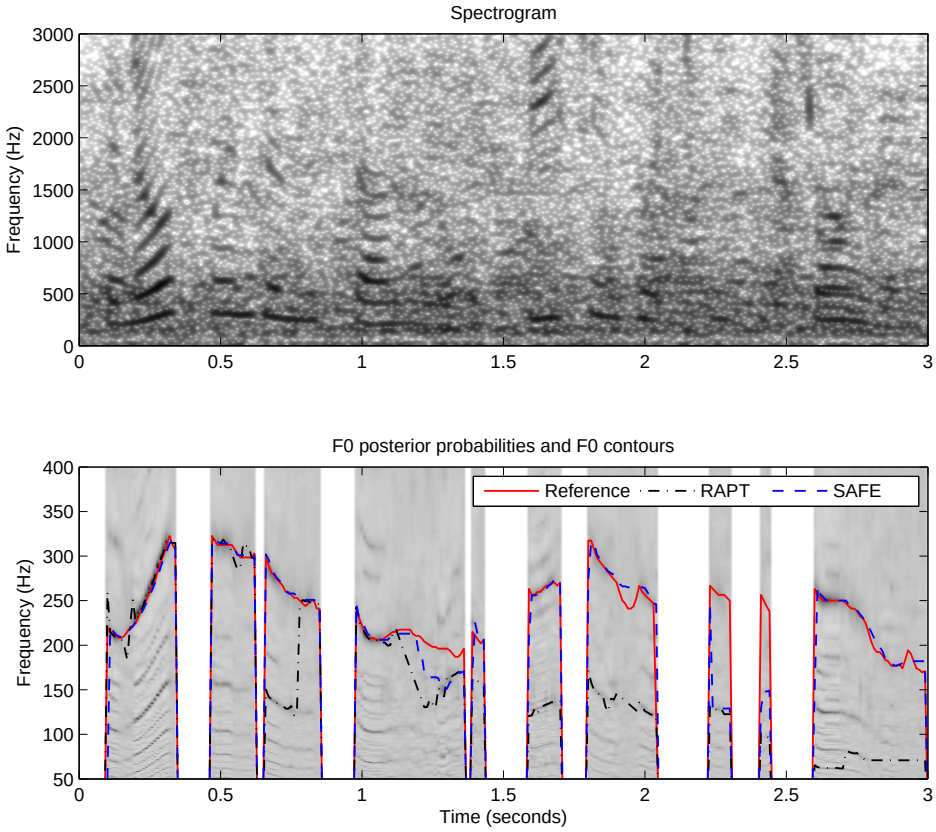

Figure 2.9: The spectrogram, F0 posterior probabilities from SAFE, and F0 contours from RAPT and SAFE of a segment of an utterance from the second female speaker (f2nw0000) in the KEELE corpus under babble noise condition at 0 dB SNR.

vertical strip in the bottom panel shows the F0 posterior probabilities over the voiced frame. The darker a point is, the higher the probability that F0 corresponds to that frequency. Since RAPT has the lowest GPE among all the F0 estimators, and SAFE uses the same cost function as RAPT for the dynamic programming-post processing, only the tracked F0 of RAPT and SAFE are shown in Fig. 2.9. It can be observed from the spectrogram in the top panel that the babble noise is mostly concentrated on the low frequency band. The babble noise can corrupt the harmonic structure of the voiced frame by suppressing, inserting, or shifting the spectral peaks in the original clean speech. These distortions may cause estimation errors. For some regions in which the target speech has high energy at high frequencies, e.g., around 1.6 s, the prominent peaks in the middle and high frequency bands, which are less affected by noise, may be used to infer the F0 value.

2.5 Experiments

In this section, we compare GPE, MFPE, and SDFPE using the KEELE [53] and CSTR [55] corpora. The 5 minute 37 seconds KEELE corpus contains a simultaneous recording of speech and laryngograph signals for a phonetically-balanced text which was read by 5 male and 5 female speakers. The 5 minute 32 seconds CSTR corpus is composed of laryngograph and speech signals from one male and one female speaker. Each speaker read 50 sentences in the CSTR corpus. Ground truth F0s were obtained by running an autocorrelation method on the laryngograph signal in addition to some manual correction.

Speech signals are downsampled from 20000 Hz to 16000 Hz for both corpora. Noise is artificially added to the corpora to test the robustness of the F0 trackers under different noise conditions. The program FaNT [56] with the default

command line option (-u -m snr_8khz) was used to employ white and babble noise segments from the NOISEX92 [57] corpus to the speech signals to generate utterances with SNR of 20, 10, 5, 0, and -5 dB. The white noise is acquired by sampling high-quality analog noise generator. The babble noise is acquired by recording 100 people speaking in a canteen with a room radius over 2m.

The parameters of SAFE are as follows: FFT size is 16384; frequency resolution is 1 Hz; frame length and step size are 0.04 and 0.01 seconds, respectively; f_{0min} and f_{0max} are 50 and 400 Hz, respectively; the lengths of the short-term and long-term windows for spectrum smoothing are 50 and 400 in Hz, respectively. A peak is regarded as a prominent peak if the normalized difference SNR $\bar{\zeta}_i$ is greater than an empirically determined threshold of 0.33; the ranges of the low, middle, and high frequency bands are 0-1, 1-2, and 2-3 kHz, respectively; local SNRs of the peaks are rounded to the nearest value in the following sequence $10r/3$, where $r = 0, 1, \cdots, 21$. The weighting factors in Eq. 2.4 are all set to the reciprocal of the number of the prominent peaks in that frame.

For the KEELE corpus, a 5-fold cross-validation scheme is applied. For each fold under a certain noise level, the speech of one male and one female speaker are used for testing, the residual and SNR models are trained from the remaining speech and its ground truth. Since 54% of the KEELE corpus is voiced speech, if the frame step size is 0.01 seconds, each fold has about 14000 frames for training. Since there are 23 rounded local SNR bins, if each voiced frame has 10 prominent peaks on average, each residual model has about 6000 samples for training. Because some bins with high SNRs might have fewer training instances, e.g., 5% of the average - 300 samples, it is still possible to robustly train a doubly-truncated Laplacian distribution with only two free parameters.

A comparison of the GPEs of RAPT, Praat, TEMPO, YIN, Wu et al.'s

Table 2.1: The GPEs (%) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the KEELE corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution. Bold numbers represent the lowest GPE in each column.

SNR (dB)	Clean	20	10	5	0	-5
		KEELE White Noise				
RAPT	2.62	2.69	3.10	4.09	7.69	17.83
Praat	3.22	3.16	4.28	6.11	11.53	30.91
TEMPO	2.98	3.41	4.27	5.57	12.79	22.64
YIN	2.94	2.94	3.20	3.96	6.70	14.48
WWB	4.22	4.27	5.21	5.57	6.42	8.87
SAFE (EW, LFB)	3.13	3.09	3.74	4.39	4.72	6.29
SAFE (EW, $\mu = 0$)	3.00	3.04	3.38	3.71	4.10	5.16
SAFE (EW)	2.98	3.01	3.35	3.66	4.06	5.01
SAFE	**2.03**	**2.18**	**2.36**	**2.68**	**2.92**	**4.65**
		KEELE Babble Noise				
RAPT		2.87	7.19	15.99	29.76	58.40
Praat		3.18	8.33	17.97	35.26	54.06
TEMPO		4.69	13.99	26.98	43.98	65.15
YIN		3.27	8.89	19.71	36.75	57.35
WWB		6.76	12.48	21.20	32.84	55.40
SAFE (EW, LFB)		3.23	6.01	10.21	20.64	47.21
SAFE (EW, $\mu = 0$)		3.14	4.75	7.68	16.23	39.62
SAFE (EW)		3.10	4.72	7.44	15.88	**39.23**
SAFE		**2.32**	**3.91**	**6.66**	**15.52**	39.35

Table 2.2: The GPEs (%) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the CSTR corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution. Bold numbers represent the lowest GPE in each column.

SNR (dB)	Clean	20	10	5	0	-5
		CSTR White Noise				
RAPT	2.45	2.46	3.04	3.94	6.73	17.72
Praat	2.27	2.27	2.99	4.35	11.84	27.54
TEMPO	2.27	2.29	2.87	5.07	11.64	31.65
YIN	2.25	2.25	2.36	3.34	5.20	12.33
WWB	2.75	3.00	4.00	4.83	5.35	7.64
SAFE (EW, LFB)	2.49	2.52	2.97	3.49	3.93	4.14
SAFE (EW, $\mu = 0$)	2.40	2.41	2.69	3.10	3.24	3.68
SAFE (EW)	2.45	2.46	2.73	3.25	3.34	3.76
SAFE ($\mu = 0$)	**1.42**	**1.47**	**1.63**	**1.63**	**2.10**	**2.91**
		CSTR Babble Noise				
RAPT		2.86	8.36	24.41	46.41	64.52
Praat		2.65	10.55	27.15	46.32	64.24
TEMPO		3.56	15.24	33.10	54.43	66.38
YIN		2.36	10.09	27.53	51.15	68.22
WWB		4.82	14.15	30.09	49.05	66.00
SAFE (EW, LFB)		2.69	5.37	9.97	23.59	63.20
SAFE (EW, $\mu = 0$)		2.61	4.14	7.73	19.32	57.17
SAFE (EW)		2.63	4.23	8.23	20.74	59.54
SAFE ($\mu = 0$)		**1.51**	**2.50**	**5.65**	**19.27**	**52.80**

method (WWB), and SAFE on the KEELE corpus is shown in Tables 2.1 and 2.2. Note that Yegnanarayana et al.'s [11] results are not included, because silence was added to the KEELE corpus in their experiments. There are two configurations of Praat: autocorrelation (default) or cross-correlation. The cross-correlation configuration is used, since it consistently provided better results. The default settings were used for RAPT, Praat, TEMPO, YIN, and WWB, except that the voicing thresholds were optimized. The implementation of WWB was provided by Prof. Dan Ellis and his group at Columbia University. Three configurations of SAFE were compared: standard (SAFE), only with information from the low frequency band as the prevailing F0 tracking algorithms (SAFE (LFB)), and with zero mean residual estimation (SAFE ($\mu = 0$)). It can be seen that all F0 trackers have GPEs lower than 3.5% in quiet. All algorithms suffer from performance degradation when the SNR drops. As expected, it is more difficult to accurately estimate F0 in the babble noise condition compared to the white noise condition with the same SNR. The SAFE algorithm has the lowest GPE when the SNR is at or below 5 dB under white noise, or at or below 10 dB under babble noise. It can be concluded from Tables 2.1 and 2.2 that discarding information from middle and high frequency bands can cause an increase in GPE, especially for babble noise which is usually concentrated at low frequencies. Forcing the means of the estimated residual distributions to be zero can also result in an increase in GPE.

To determine the generalizablity of SAFE, the model trained from the KEELE corpus is used for the CSTR corpus. According to the performances of the F0 algorithms shown in Tables 2.1 and 2.2, it can be seen that F0 estimation for the CSTR corpus is easier under white noise, but harder under babble noise compared to the KEELE corpus. Although there is mismatch between the KEELE and CSTR corpora, SAFE still has the lowest GPE under SNR conditions for both.

The mismatch can explain why SAFE ($\mu = 0$) has a lower GPE compared to the standard SAFE. Thus, it may be more appropriate to use SAFE ($\mu = 0$) when prior information of the testing set is not available.

The MFPEs for the KEELE and CSTR corpora are shown in Tables 2.3 and 2.4. It can be seen that the best configuration of SAFE has less than 1 Hz MFPEs under all noise conditions. Other F0 trackers have less than 3 Hz MFPEs under most noise conditions. Note that 3 Hz is only 1.2% of the average of all possible F0s which is 225 Hz. That means all F0 trackers do not make significantly biased F0 estimation under clean and most noisy conditions. For the KEELE corpus, the means of the residuals are slightly less than zero most of the time as shown in Fig. 2.7. Thus, the standard SAFE which considers the bias is supposed to have slightly lower F0 estimation than the zero mean version of SAFE. Due to the mismatch between KEELE and CSTR corpora, the negative bias causes the MFPEs of the standard SAFE to be more deviated from zero compared to the zero mean version of SAFE on the CSTR corpus.

The SDFPEs on KEELE and CSTR corpus are shown in Tables 2.5 and 2.6. It can be seen that the SDFPEs of SAFE are slightly higher (1-2 Hz) than other F0 estimators under some conditions. Since the MFPE and SDFPE are calculated over the frames in which the F0 tracker does not have gross F0 estimation errors (less than 20% gross error), the number of frames for calculating the SDFPE over different F0 trackers under the same noise condition is different. It is known that F0 estimation accuracy is higher over less noisy frames [54]. Given a certain noise condition, if an estimator only correctly estimates F0 over a few frames that have high frame-level SNRs, it could have relatively low MFPE and SDFPE, but a high GPE. Therefore, having higher MFPEs or SDFPEs does not necessarily mean that SAFE is less accurate in F0 estimation.

Table 2.3: The MFPE (Hz) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the KEELE corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution.

SNR (dB)	Clean	20	10	5	0	-5
		KEELE White Noise				
RAPT	0.79	0.60	0.60	0.32	-0.18	-1.87
Praat	0.19	0.21	-0.14	0.67	-1.93	-4.08
TEMPO	0.41	0.36	0.27	0.08	-1.26	-2.16
YIN	0.55	0.56	0.54	0.53	0.53	0.43
WWB	2.86	2.87	2.74	2.67	2.35	2.05
SAFE (EW, LFB)	-0.40	-0.40	-0.43	-0.47	-0.66	-0.61
SAFE (EW, $\mu = 0$)	0.15	0.34	0.34	0.28	0.04	-0.05
SAFE (EW)	-0.36	-0.46	-0.50	-0.57	-0.72	-0.86
SAFE	-0.14	-0.16	-0.24	-0.28	-0.43	-0.49
		KEELE Babble Noise				
RAPT		0.74	0.47	0.23	-0.35	-0.24
Praat		0.24	0.21	0.05	0.16	0.50
TEMPO		0.34	-0.06	-1.19	-0.09	1.22
YIN		0.66	0.83	0.93	1.11	1.03
WWB		2.66	2.34	1.95	1.35	0.89
SAFE (EW, LFB)		-0.42	-0.52	-0.51	-0.33	0.10
SAFE (EW, $\mu = 0$)		0.21	0.04	-0.12	-0.19	-0.12
SAFE (EW)		-0.49	-0.65	-0.78	-0.71	-0.47
SAFE		-0.21	-0.27	-0.47	-0.29	0.12

Table 2.4: The MFPE (Hz) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the CSTR corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution.

SNR (dB)	Clean	20	10	5	0	-5
		CSTR White Noise				
RAPT	-0.06	-0.27	-0.22	-0.31	-0.59	-2.07
Praat	-0.77	-0.78	-0.97	-1.34	-2.79	-4.71
TEMPO	-0.85	-0.73	-0.76	-0.97	-1.21	-2.66
YIN	-0.39	-0.40	-0.44	-0.47	0.60	-0.62
WWB	2.73	2.67	2.49	2.34	2.19	1.93
SAFE (EW, LFB)	-1.28	-1.32	-1.39	-1.40	-1.45	-1.53
SAFE (EW, $\mu = 0$)	-0.78	-0.53	-0.50	-0.53	-0.62	-0.81
SAFE (EW)	-1.39	-1.43	-1.46	-1.49	-1.59	-1.69
SAFE ($\mu = 0$)	-0.41	-0.14	-0.15	-0.17	-0.25	-0.40
		CSTR Babble Noise				
RAPT		-0.19	-0.34	-0.18	-0.35	-0.14
Praat		-0.79	-0.72	-0.44	-0.30	0.13
TEMPO		-0.54	-0.71	-0.77	0.51	0.69
YIN		-0.36	-0.14	-0.06	0.04	0.28
WWB		2.05	1.55	1.24	0.77	0.34
SAFE EW, (LFB)		-1.40	-1.45	-1.39	-1.13	-0.47
SAFE (EW, $\mu = 0$)		-0.65	-0.78	-0.81	-0.92	-0.42
SAFE (EW)		-1.46	-1.55	-1.52	-1.40	-0.69
SAFE ($\mu = 0$)		-0.33	-0.37	-0.46	-0.42	-0.23

Table 2.5: The SDFPE (Hz) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the KEELE corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution.

SNR (dB)	Clean	20	10	5	0	-5
		KEELE White Noise				
RAPT	4.41	4.50	4.75	5.54	6.62	9.92
Praat	3.69	3.71	4.89	6.05	8.96	12.74
TEMPO	5.04	5.19	5.84	7.25	9.43	11.52
YIN	4.45	4.47	4.60	4.82	5.21	5.59
WWB	5.65	5.59	5.61	5.75	6.02	6.82
SAFE (EW, LFB)	5.63	5.62	5.63	5.70	5.99	6.48
SAFE (EW, $\mu = 0$)	5.48	5.49	5.51	5.54	5.95	6.43
SAFE (EW)	5.53	5.56	5.62	5.67	6.07	6.50
SAFE	4.05	3.87	4.06	4.28	4.78	5.64
		KEELE Babble Noise				
RAPT		4.85	5.96	6.83	8.57	9.39
Praat		3.85	4.86	5.79	7.03	8.86
TEMPO		5.92	9.06	12.01	13.29	11.79
YIN		4.71	5.30	5.81	6.76	7.94
WWB		5.62	6.08	6.61	7.17	8.15
SAFE (EW, LFB)		5.60	6.09	6.67	7.65	8.40
SAFE (EW, $\mu = 0$)		5.56	6.04	6.64	7.48	9.35
SAFE (EW)		5.60	6.07	6.70	7.58	9.34
SAFE		4.03	5.05	5.82	7.20	8.93

Table 2.6: The SDFPE (Hz) of the RAPT, Praat, TEMPO, YIN, WWB, and SAFE using the CSTR corpus. **EW**: use equal weighting in Eq. 2.4. **LFB**: only the low frequency band (0-1000 Hz) is used. $\boldsymbol{\mu}$**=0**: a zero mean is used in the doubly truncated Laplacian distribution.

SNR (dB)	Clean	20	10	5	0	-5
		CSTR White Noise				
RAPT	5.49	5.78	6.02	6.57	7.92	10.67
Praat	6.04	6.09	6.54	7.56	10.22	14.38
TEMPO	6.76	7.28	7.74	8.55	10.41	13.29
YIN	6.28	6.29	6.35	6.46	6.68	6.75
WWB	6.86	6.83	6.79	6.90	7.09	7.61
SAFE (EW, LFB)	8.10	8.00	7.97	7.93	7.92	8.31
SAFE (EW, $\mu = 0$)	7.85	7.81	7.82	7.80	7.89	8.19
SAFE (EW)	7.89	7.85	7.74	7.71	7.87	8.19
SAFE ($\mu = 0$)	5.93	5.80	5.96	6.08	6.39	6.96
		CSTR Babble Noise				
RAPT		5.84	6.86	7.47	7.84	8.78
Praat		6.06	6.36	6.45	6.85	7.87
TEMPO		7.76	10.86	13.96	14.98	12.50
YIN		6.33	6.25	5.96	5.59	6.25
WWB		6.14	5.84	5.85	5.74	6.46
SAFE (EW, LFB)		8.03	8.12	8.19	8.34	7.09
SAFE (EW, $\mu = 0$)		7.64	7.97	8.19	8.55	7.62
SAFE (EW)		7.59	7.91	8.16	8.49	7.49
SAFE ($\mu = 0$)		5.91	6.41	7.03	7.92	7.27

2.6 Conclusions

Prominent Signal-to-Noise Ratio (SNR) peaks constitute a simple and an effective information source for F0 inference under both clean and noisy conditions. The statistical framework of F0 estimation is promising in modeling the effect of additive noise on clean speech spectra given F0. In addition to low frequencies, middle and high frequency bands (1-3 kHz) provide supplemental useful information for F0 inference. The proposed SAFE algorithm is more effective in reducing the GPE compared to prevailing F0 trackers especially at low SNRs, and is robust in maintaining low Mean and Standard Deviation of the Fine Pitch Errors.

CHAPTER 3

Unvoiced/Voiced Classification and F0 Tracking

In this chapter, an algorithm to reduce the unvoiced/voiced (U/V) classification errors in F0 tracking, i.e. VDEs, under noisy conditions is introduced. We will show in the experimental section that reducing VDEs can eventually result in a reduction of FFEs.

To improve the accuracy and overcome the instability of U/V detection methods that rely on thresholds, we introduce a model-based U/V classification frontend whose output can be taken as an U/V mask for any F0 tracker. With the help of the model-based method, parameters are automatically learned and adjusted during model training and unsupervised adaptation. Reliable U/V boundary information results in improved F0 tracking.

There have been several model-based techniques for Voice Activity Detection (VAD) [58] [59] [60], but they primarily distinguish voiced frames from unvoiced frames.

The flowchart of the proposed U/V classifier and its relationship to the subsequent F0 tracker are illustrated in Fig. 3.1.

Two methods for U/V classification are introduced: Hidden Markov Model (HMM), and Gaussian Mixture Model (GMM)-Based U/V Classification. The two U/V classifiers are discussed in the following.

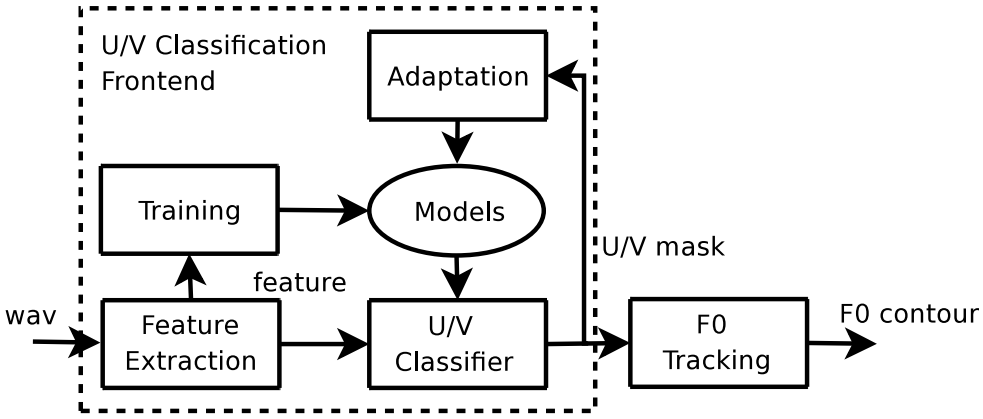

Figure 3.1: U/V Classification Frontend for F0 Trackers

3.1 Hidden Markov Models-Based Unvoiced/Voiced Classification

Two acoustic models were trained, one for unvoiced sounds (U) and the other for voiced sounds (V). The mapping to U and V sounds is shown in Table 3.1. The phone symbols shown in the table are used in the TIMIT phone level transcription. 'pau' is a pause, 'epi' is an epenthetic silence, 'h#' is the begin/end marker (non-speech events).

For feature extraction, both Mel-Frequency Cepstral Coefficients (MFCCs) and the noise robust AFE [61] frontend are used. The U/V models are left-to-right HMMs with 3 emitting states, and 256 Gaussian components per mixture model. A word net containing unvoiced and voiced nodes with a bigram language model attached to the directed arcs between the nodes was constructed. The U/V decision can be adjusted by tuning the language model. For example, increasing $P(voiced)$ or $P(voiced|unvoiced)$ would make the decoder prone to making more voiced hypotheses.

The training set is an American English corpus (TIMIT, approximately 4 hours). The test set (KEELE) is based on speech by British English speakers.

Table 3.1: Phonemes and Sounds to U and V Dictionary. The phonemes are used in the TIMIT phone level transcription.

	Unvoiced	Voiced
Stops	p(cl) t(cl) k(cl)	b(cl) d(cl) g(cl) dx
Affricates & Fricatives	ch s f th sh	jh z v zh dh
Nasals & Vowels	-	m n ng em en eng nx iy ih eh ey ae aa aw ay ah ao oy ow uh uw ux er ax ix axr ax-h
Semivowels & Glides	hh hv	l r el w y
Others	epi h pau	-

Hence, we need to apply offline unsupervised Maximum Likelihood Linear Regression (MLLR) speaker adaptation to adapt the initial SI models to speaker dependent (SD) models [62]. In SD model adaptation for speaker s, depending on the amount of adaptation data, either global or regression tree style adaptation can be used. When a small amount of adaptation data is available, a global adaptation transform can be obtained. The linear transform can be applied to all the Gaussians in the model set. As more adaptation data are available, a binary regression class tree can be used to group Gaussians that are close in feature space together. The purpose of the Gaussian grouping is to allow the adaptation of distributions for which there are no observations. Since each grouped Gaussian can obtain an adaptation transform, which is more specific compared to the global style adaptation, improved performance is expected. The regression tree used in U/V classification task is shown in Fig. 3.2, the regression tree is composed of a

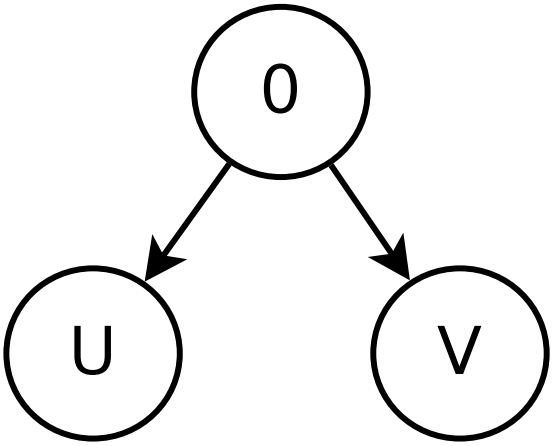

Figure 3.2: The regression tree used in the adaptation of U/V Classification. **0**: the root node in which all the Gaussians are grouped. **U/V**: the leaf node in which all the Gaussians of all the emitting states in the U or V HMM are grouped.

base node connected to two leaves which are unvoiced (U) and voiced (V). For a speaker, the global style adaptation uses all the data to train a global transformation. Regression tree based adaptation needs to use the decoding results to attach the data to leaf node U/V, and then use the attached data to train a transformation for the leaf node U/V.

3.2 Gaussian Mixture Models-Based Unvoiced/Voiced Classification

It should be mentioned that the phonemes and sounds to U and V dictionary shown in Table 3.1 can not guarantee the derived voicing labels of TIMIT database to be 100% accurate. To alleviate this problem, we also used the KEELE database which has F0 annotations for training unvoiced/voiced models. For each frame, MFCCs and their first and second order derivatives with cepstral mean normalization are extracted. Because the KEELE database (less than 6 minutes)

is much smaller than the TIMIT database (about 4 hours), less complex models with fewer parameters, i.e., Gaussian Mixture Models (GMMs), are trained instead of HMMs to ensure that there are enough samples for parameter estimation. There are 8 Gaussian components per mixture model. In testing, the voicing probability of each frame is obtained by evaluating the feature vector of that frame on GMMs. A Hamming window of length 11 frames is then applied to the voicing probabilities over frames to have a smoother U/V decision results. The length of the Hamming window is empirically set.

3.3 F0 Frame Error and GPE-VDE Curve

It is desirable for an F0 tracking algorithm to reduce the VDE and GPE at the same time. The error of an F0 tracking method is usually presented as an error pair: (GPE, VDE). But some algorithms have low GPE, but high VDE, compared to other algorithms. We propose an error metric called the F0 Frame Error (FFE) which takes both GPE and VDE into consideration. We plot the GPE-VDE curve as a Receiver Operating Characteristics (ROC) curve to show the trade-off between GPE and VDE. With the help of the FFE and the GPE-VDE curve, we can compare the performance of F0 trackers in a unified framework.

When the tracked F0 contour is compared to the ground truth, there can only exist 3 possible types of gross errors in any frame i:

- U→V Error: an unvoiced frame is classified as a voiced frame;
- V→U Error: a voiced frame is classified as an unvoiced frame;
- F0 Value Estimation Error: the estimated F0 value deviate too much from the ground truth.

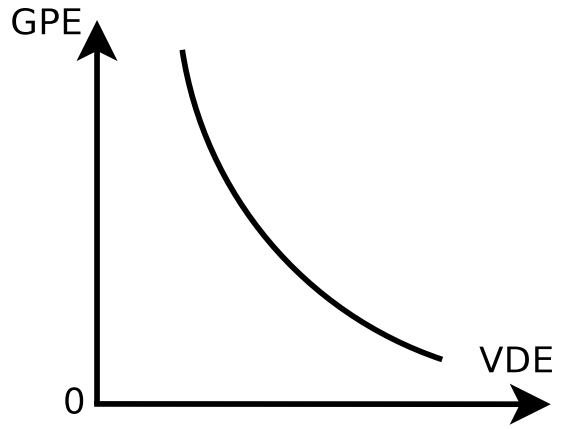

Figure 3.3: A sketch of a GPE-VDE curve

In Fig. 1.1, the F0 tracker made U→V errors over N_2 frames, F0 value estimation errors over N_4 frames, and V→U errors over N_6 frames. We propose an F0 Frame Error (FFE) metric which sums the three types of errors mentioned above:

$$\begin{aligned} FFE &= \frac{\text{\# of error frames}}{\text{\# of total frames}} \times 100\% \\ &= \frac{N_{U\to V} + N_{V\to U} + N_{F0E}}{N} \times 100\%. \end{aligned} \tag{3.1}$$

FFE is also a combination of GPE and VDE:

$$\begin{aligned} FFE &= \frac{N_{F0E}}{N} \times 100\% + \frac{N_{U\to V} + N_{V\to U}}{N} \times 100\%. \\ &= \frac{N_{VV}}{N} \times GPE + VDE \end{aligned} \tag{3.2}$$

Therefore, FFE takes both GPE and VDE into consideration making the comparison of different F0 trackers possible.

We also propose showing a GPE-VDE curve which is effective in showing the relationship between the two parameters. A sketch of a possible GPE-VDE curve is shown in Fig. 3.3. When optimizing the parameters of the F0 tracker, we can obtain a set of (GPE, VDE) pairs. $(\text{GPE}_i, \text{VDE}_i)$ is a minimum point if and only

if there exists no j that satisfies $\mathrm{GPE}_j < \mathrm{GPE}_i$ and $\mathrm{VDE}_j < \mathrm{VDE}_i$ at the same time. When plotting all the minimum points, we can obtain a GPE-VDE curve.

The F0 Frame Error (FFE) and GPE-VDE curve can be used to evaluate the F0 tracking algorithms in a unified framework.

3.4 Experiments

In this section, we compare the VDEs of HMM and GMM-based U/V classifiers using the KEELE and CSTR corpus.

The noise addition procedure has been described in the experimental section of Chapter 2. In U/V classification, the training and testing noise types and SNR levels are matched.

3.4.1 Using HMM-based Unvoiced/Voiced Classifier

Table 3.2 shows the VDEs of the proposed HMM-based U/V classifier with different features before and after adaptation. Unsupervised speaker adaptation is effective in minimizing the mismatch between training and test data. AFE features are always better than MFCC features before and after adaptation. For the white noise cases, the VDE of the regression class tree based adaptation (RSD) is lower than that of global adaptation (GSD). In the babble noise case, the GSD resulted in slightly better performance for AFE features. This could be because babble noise is more correlated with the underlying speech signal than white noise is.

The U/V classification result was then used as a mask for F0 trackers. Since RAPT and Praat do not have the option of directly using an U/V mask, the effect of the mask is only tested on TEMPO. The U/V decoder using AFE features and

Table 3.2: VDEs (%) of the U/V Classifier Using the KEELE Corpus (SNR = 0 dB, **SI**: speaker independent models, **GSD/RSD**: global style/regression tree style adapted models, **MFCC** and **AFE** are the features used in the classifier)

VDE	White Noise		Babble Noise	
	MFCC	AFE	MFCC	AFE
SI	11.57	10.84	30.70	26.27
GSD	10.98	9.81	27.61	**22.48**
RSD	**10.18**	**9.14**	**27.23**	23.54

SD models is used for both noise conditions. To take advantage of the decoder that has the lowest VDE, regression tree style adaptation is used under white noise, but global style adaptation is used under babble noise.

For each F0 tracking package, 500 - 1000 configurations are tested where different parameters are adjusted (e.g., the correlation window length, voicing thresholds). The performance of the F0 tracker under each configuration corresponds to certain values for GPE, VDE, and FFE as shown in Table 3.3. 'M+' denotes the U/V mask by the model-based classifier. In white and babble noise, the lowest GPE is achieved by Praat, and the lowest VDE by M+TEMPO. Note that minimizing the FFE results in a significant reduction in GPE. Take TEMPO in white noise for example, when we shift our objective from minimizing the VDE to FFE, the VDE slightly increases from 14.52% to 14.69%, but the GPE significantly decreases from 15.87% to 4.93%. That is also true for RAPT, Praat, and TEMPO in babble noise. Compared to TEMPO, the FFE of M+ drops by 24.4% in white noise, and 27.6% in babble noise. It could be inferred that only minimizing the VDE can not guarantee the minimization of the overall FFE, but reducing the VDE is helpful for lowering FFE. Note that the GPE for

Table 3.3: GPEs, VDEs and FFEs (%) on KEELE Database corrupted by white and babble noise at 0 dB SNR, **M+**: U/V mask provided by model-based classifier trained on TIMIT database corrupted by white and babble noise at 0 dB SNR, **mV/mF**: when VDE/FFE is minimized. Bold numbers denote the lowest error rate in each column.

		White Noise			Babble Noise		
		GPE	VDE	FFE	GPE	VDE	FFE
RAPT	mV	3.19	20.00	21.04	31.56	28.21	37.58
	mF	2.83	20.02	20.94	8.51	30.65	32.79
Praat	mV	**2.10**	19.72	20.41	31.82	29.32	38.69
	mF	**2.10**	19.72	20.41	**5.31**	32.67	33.86
TEMPO	mV	15.87	14.52	20.59	58.05	36.51	50.35
	mF	4.93	14.69	16.56	8.11	40.16	41.24
M+TEMPO	mV	7.10	9.14	**12.52**	18.65	**22.48**	**29.86**
	mF	7.10	**9.14**	**12.52**	18.65	**22.48**	**29.86**

M+TEMPO is higher than TEMPO when minimizing the FFE.

In the GPE-VDE curves shown in Figs. 3.4 and 3.5, it can be observed that for every F0 tracker without the U/V mask, GPE decreases when VDE increases. As shown in Eqs. 1.13 and 1.14, when the VDE increases, it may be due to an increase in the V→U errors resulting in a reduction in N_{VV}. Although the N_{VV} decreases, the N_{F0E} decreases more, for it is easier to estimate the F0 value over the remaining voiced frames with a higher SNR. Since the ratio of N_{F0E} to N_{VV} decreases, the GPE decreases. Take TEMPO in white noise for example, when the VDE increases from 14.69% to 21.92%, the V→U error rate increases from to 27.05% to 41.60%, the U→V error rates shift from 1.25% to 0.50%, and the GPE

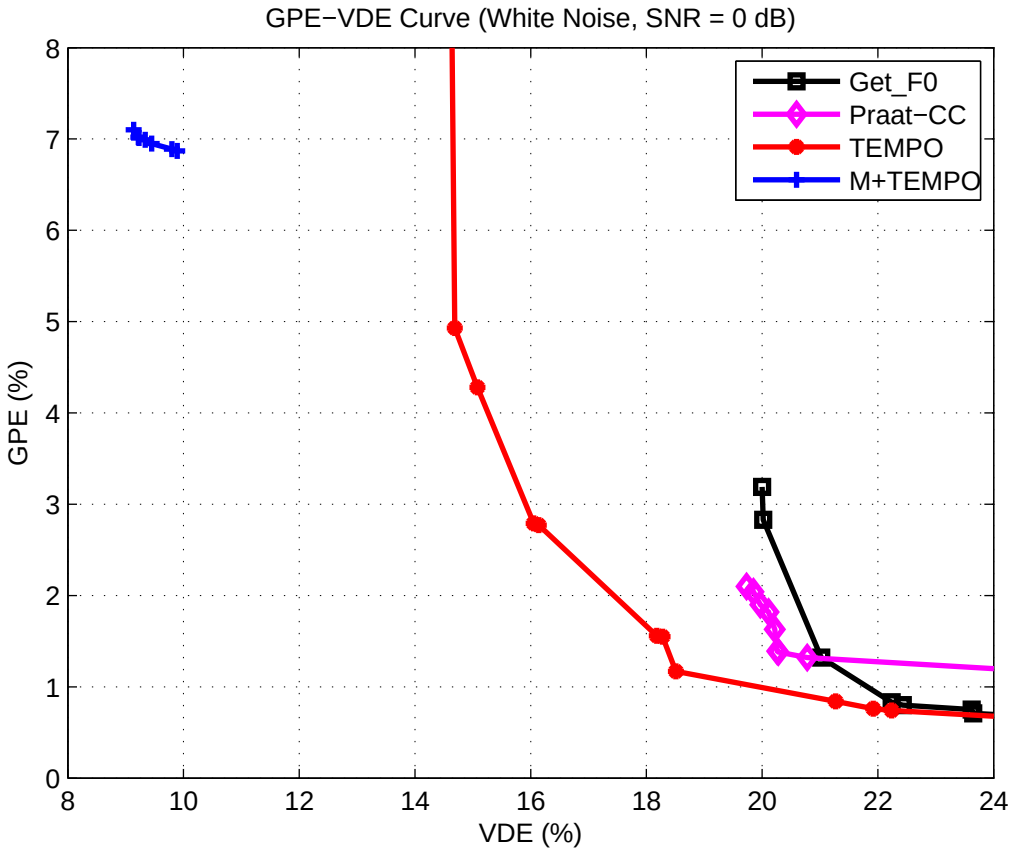

Figure 3.4: GPE-VDE curves on KEELE database corrupted by white noise at 0 dB SNR. (**M+**: using U/V classifier output as a mask)

decreases from 4.93% to 0.76%. But for F0 trackers with U/V masks, the VDE is more stable, and the GPE does not change much. Because the F0 tracker has to estimate F0 for every voiced frame indicated by the mask, even if it is a frame with a low SNR. Take M+TEMPO in white noise for example, when the VDE increases from 9.14% to 9.89%, the V→U error rate increases from to 8.60% to 10.63%, the U→V error rate decrease from 9.73% to 9.08%, the GPE slightly decreases from 7.10% to 6.87%.

It is also shown in Figs. 3.4 and 3.5 that integrating our model-based U/V classifier into an F0-tracking algorithm can improve its voicing decision accuracy.

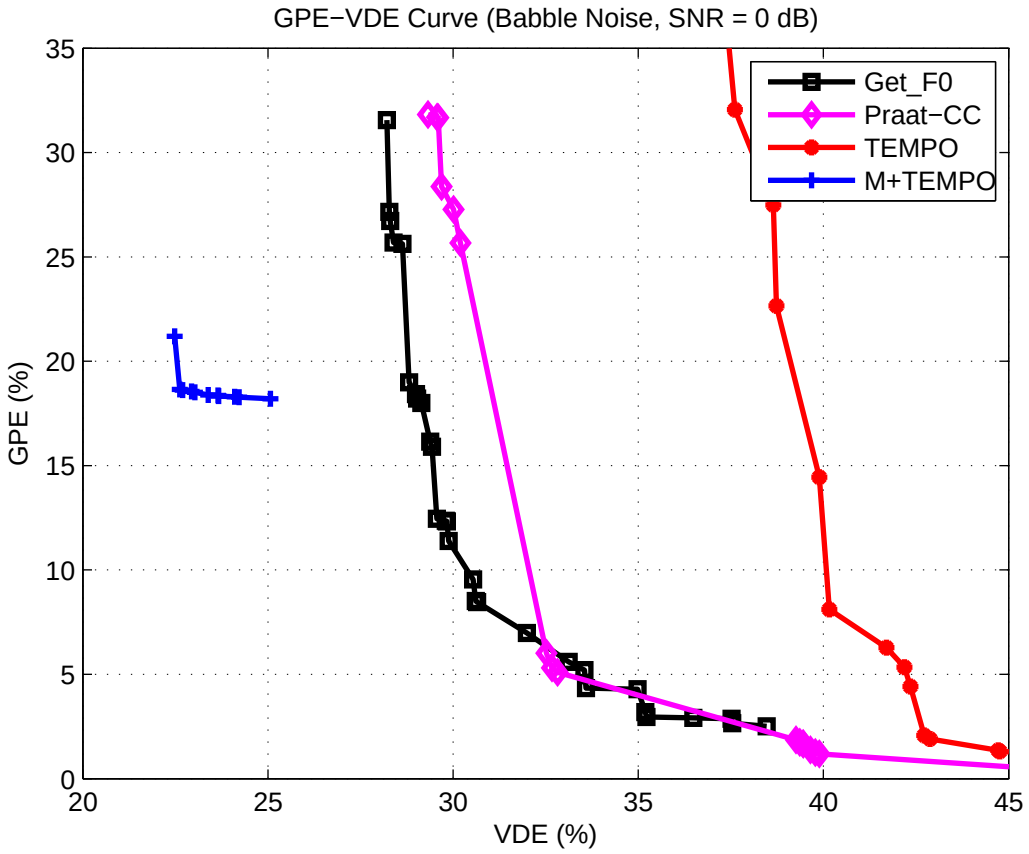

Figure 3.5: GPE-VDE Curves on KEELE Database corrupted by babble noise at 0 dB SNR. (**M+**: using U/V classifier output as a mask)

Table 3.4: A comparision of GPEs, VDEs and FFEs of RAPT, Praat, TEMPO, and SAFE algorithms on KEELE and CSTR corpus. Clean condition. Bold numbers denote the lowest error rate in the FFE column.

Clean	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE Database				
RAPT	2.04	4.97	5.18	5.07	6.08
Praat	1.89	6.78	3.99	5.44	6.35
TEMPO	1.09	6.67	6.12	6.41	6.94
SAFE	1.97	3.83	4.44	4.60	**5.58**
	CSTR Database				
RAPT	2.05	4.10	6.06	5.32	5.96
Praat	1.23	5.20	4.55	5.41	5.85
TEMPO	0.82	5.49	6.17	5.92	6.21
SAFE	1.04	2.88	6.54	5.19	**5.68**

Take TEMPO and M+TEMPO in white noise for example, after applying the U/V mask, the minimum VDE decreases from 14.52% to 9.14%.

3.4.2 Using GMM-based Unvoiced/Voiced Classifier

In this section, the F0 tracking results of SAFE algorithm using the mask generated by the GMM-based U/V classifier are compared with RAPT, Praat, and TEMPO. The algorithm WWB [19] was not included because its U/V classifier failed under noisy conditions. Note that on KEELE corpus, the U/V masks are generated in a 5-fold cross-validation scheme as mentioned in the previous chapter. On CSTR corpus, the U/V masks are generated by using the GMM-based U/V classifier trained on the KEELE corpus. For generating U/V masks

Table 3.5: A comparision of GPEs, VDEs and FFEs of RAPT, Praat, TEMPO, and SAFE algorithms on KEELE and CSTR corpus. White and babble noise conditions, SNR = 20 dB. Bold numbers denote the lowest error rate in the FFE column.

SNR = 20 dB	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE White Noise				
RAPT	1.58	8.89	2.90	6.02	6.77
Praat	1.72	7.62	3.39	5.60	6.43
TEMPO	0.99	7.62	6.00	6.85	7.32
SAFE	2.06	5.20	4.23	4.74	**5.75**
	KEELE Babble Noise				
RAPT	1.89	7.54	5.19	6.42	7.33
Praat	1.70	7.95	4.50	6.30	7.12
TEMPO	1.19	12.41	6.96	9.80	10.34
SAFE	1.85	7.33	4.59	6.02	**7.00**
	CSTR White Noise				
RAPT	1.75	6.14	5.87	5.97	6.54
Praat	1.26	5.27	5.82	5.78	6.23
TEMPO	0.71	6.01	6.07	6.05	6.29
SAFE	1.24	3.83	6.37	5.43	**5.87**
	CSTR Babble Noise				
RAPT	2.14	5.72	9.49	8.09	8.84
Praat	1.67	6.63	6.16	6.34	6.93
TEMPO	0.90	9.56	7.16	8.05	8.36
SAFE	1.21	5.91	6.74	6.43	**6.86**

Table 3.6: A comparision of GPEs, VDEs and FFEs of RAPT, Praat, TEMPO, and SAFE algorithms on KEELE and CSTR Corpus. White and babble noise conditions, SNR = 10 dB. Bold numbers denote the lowest error rate in the FFE column.

SNR = 10 dB	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE White Noise				
RAPT	1.21	22.02	1.23	12.07	12.56
Praat	1.27	18.08	1.45	10.12	10.66
TEMPO	1.00	11.95	3.30	7.81	8.27
SAFE	1.97	4.97	4.93	4.86	**5.83**
	KEELE Babble Noise				
RAPT	4.93	15.80	6.25	11.23	13.39
Praat	4.78	14.70	11.33	13.09	15.21
TEMPO	1.30	39.59	4.29	22.69	23.10
SAFE	2.82	13.99	6.86	10.58	**11.84**
	CSTR White Noise				
RAPT	1.07	16.11	1.21	6.76	7.10
Praat	1.25	12.64	1.36	6.56	6.97
TEMPO	0.99	8.23	5.55	6.56	6.90
SAFE	1.46	2.82	7.58	5.80	**6.33**
	CSTR White Noise				
RAPT	6.54	12.97	10.07	11.16	13.28
Praat	8.33	11.96	15.21	14.00	16.73
TEMPO	1.71	40.32	4.87	18.08	18.46
SAFE	2.58	11.23	10.57	10.82	**11.67**

Table 3.7: A comparision of GPEs, VDEs and FFEs of RAPT, Praat, TEMPO, and SAFE algorithms on KEELE and CSTR Corpus. White and babble noise conditions, SNR = 5 dB. Bold numbers denote the lowest error rate in the FFE column.

SNR = 5 dB	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE White Noise				
RAPT	0.56	39.72	0.61	20.99	21.17
Praat	0.95	33.31	0.56	17.63	17.96
TEMPO	1.05	21.33	1.94	12.05	12.48
SAFE	2.42	6.10	4.67	5.41	**6.59**
	KEELE Babble Noise				
RAPT	11.61	27.64	11.56	19.94	24.32
Praat	12.67	27.29	16.18	21.97	26.77
TEMPO	2.81	67.92	2.82	36.75	37.22
SAFE	4.93	20.12	10.90	15.71	**17.76**
	CSTR White Noise				
RAPT	0.65	33.97	0.60	13.03	13.19
Praat	0.92	26.67	0.51	10.26	10.51
TEMPO	1.06	15.59	2.43	7.33	7.67
SAFE	1.51	3.29	8.80	6.75	**7.29**
	CSTR White Noise				
RAPT	22.66	23.75	10.81	15.63	22.07
Praat	23.90	22.58	15.96	18.43	25.32
TEMPO	5.87	68.44	4.10	28.07	28.76
SAFE	5.04	17.87	16.11	16.77	**18.31**

Table 3.8: A comparision of GPEs, VDEs and FFEs of RAPT, Praat, TEMPO, and SAFE algorithms on KEELE and CSTR Corpus. White and babble noise conditions, SNR = 0 dB. Bold numbers denote the lowest error rate in the FFE column.

SNR = 0 dB	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE White Noise				
RAPT	0.42	67.67	0.15	35.34	35.41
Praat	0.73	58.95	0.08	30.76	30.92
TEMPO	1.49	41.20	0.94	21.92	22.38
SAFE	2.15	10.73	5.67	8.31	**9.31**
	KEELE Babble Noise				
RAPT	21.64	49.32	8.09	29.58	35.29
Praat	27.38	45.87	14.71	30.95	38.67
TEMPO	8.90	88.87	2.19	47.37	47.88
SAFE	13.31	26.76	21.25	24.12	**29.20**
	CSTR White Noise				
RAPT	0.42	63.50	0.18	23.78	23.83
Praat	1.22	56.28	0.05	21.00	21.20
TEMPO	2.05	37.11	1.05	14.48	14.96
SAFE	1.88	4.37	10.30	8.09	**8.76**
	CSTR Babble Noise				
RAPT	42.75	44.92	11.87	24.18	32.96
Praat	42.32	41.27	17.91	26.61	35.87
TEMPO	20.92	90.17	3.91	36.05	36.82
SAFE	17.82	23.38	29.46	27.20	**32.29**

Table 3.9: A comparision of GPEs, VDEs and FFEs on KEELE and CSTR Corpus. White and babble noise conditions, SNR = -5 dB. Bold numbers denote the lowest error rate in the FFE column.

SNR = -5 dB	GPE (%)	V→UE (%)	U→VE (%)	VDE (%)	FFE (%)
	KEELE White Noise				
RAPT	0.17	93.18	0.00	48.56	48.57
Praat	1.88	87.79	0.00	45.75	45.87
TEMPO	1.45	70.68	0.21	36.94	37.16
SAFE	3.07	15.82	6.62	11.41	**12.76**
	KEELE Babble Noise				
RAPT	45.57	74.13	13.33	45.02	51.16
Praat	44.17	67.90	19.96	44.94	52.33
TEMPO	30.65	95.38	2.52	50.92	51.65
SAFE	40.66	25.64	39.94	32.49	**48.25**
	CSTR White Noise				
RAPT	34.67	0.00	93.03	0.01	34.67
Praat	1.72	85.39	0.00	31.82	31.91
TEMPO	6.13	71.62	0.24	26.83	27.48
SAFE	2.42	8.05	11.97	10.51	**11.34**
	CSTR Babble Noise				
RAPT	57.31	68.26	12.43	33.24	40.01
Praat	58.14	62.10	19.19	35.18	43.39
TEMPO	48.22	96.36	3.94	38.37	39.03
SAFE	56.57	16.19	39.14	30.59	**36.15**

on CSTR corpus, the U/V GMMs are trained on the KEELE corpus. For each F0 tracking package, default parameters are used. The performance of the F0 trackers on KEELE and CSTR corpora under clean, 20 dB, 10 dB, 5 dB, 0 dB, and -5 dB noise conditions are shown in Tables 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9. Note that on the CSTR corpus, the zero mean version of the SAFE algorithm, i.e., SAFE($\mu = 0$), is used.

Under clean and noisy conditions, it can be seen that the SAFE algorithm always has the lowest FFE. Under noisy conditions, the relative performance gains of the SAFE algorithm in FFE against other algorithms are greater than that under the clean condition. For example, using the KEELE corpus, in the clean condition, FFEs of SAFE and RAPT are 5.58% and 6.08%, respectively. The relative gain is 8.2%; under white noise 0 dB condition, the FFEs of SAFE and RAPT are 9.31% and 35.31%, respectively. The relative gain is 73.6%. Therefore, the SAFE algorithm with the GMM-based U/V classifier can obtain more noise robust F0 tracking results compared with prevailing F0 trackers.

A comparison of the GPEs of the estimation and tracking version of the SAFE algorithm on the KEELE and CSTR corpora under clean and noisy conditions is shown in Table 3.10. Note that the GPEs are improved for the most part, using the F0 tracking version.

3.5 Conclusions

The model-based U/V classifier can output robust U/V masks for F0 trackers under both white and babble noise conditions which is helpful for reducing the overall FFE. Minimizing the FFE is more effective than minimizing the VDE alone. The SAFE algorithm using masks generated from the GMM-based U/V

Table 3.10: A comparison of the GPEs (%) of the estimation and tracking version of the SAFE algorithm using the KEELE and CSTR corpora.

GPE (%)	Clean	20	10	5	0	-5
		KEELE White Noise				
Estimation	2.03	2.18	2.36	2.68	2.92	4.65
Tracking	1.97	2.06	1.97	2.42	2.15	3.07
		KEELE Babble Noise				
Estimation		2.32	3.91	6.66	15.52	39.35
Tracking		1.85	2.82	4.93	13.31	40.66
		CSTR White Noise				
Estimation	1.42	1.47	1.63	1.63	2.10	2.91
Tracking	1.04	1.24	1.46	1.51	1.88	2.42
		CSTR Babble Noise				
Estimation		1.51	2.50	5.65	19.27	52.80
Tracking		1.21	2.58	5.04	17.82	56.57

classifier has lower FFEs compared with prevailing F0 tracking algorithms under both clean and noisy conditions.

Part II

Noise Robust Bird Song Classification and Detection

CHAPTER 4

A Correlation-Maximization Denoising Filter

In this chapter, we automatically identify the bird species according to the recorded calls from 5 species of Antbirds (Barred Antshrike (BAS), Dusky Antbird (DAB), Great Antshrike (GAS), Mexican Antthrush (MAT), Dot-winged Antwren (DWA)) in a Mexican rainforest. The acoustic data were collected by researchers from the Ecology and Evolutionary Biology department at UCLA [38] [63].

Because the data were collected through a hand-held directional microphone, most of the noise in this study is additive noise. Wiener filtering can efficiently remove quasi-stationary additive noise [30]; however, according to our observation, it will also enhance the background chirps and other non-stationary noises. To alleviate this problem, we propose a 'correlation-maximization' filter which was inspired by the matched filter [64], originally designed for detection purposes. The periodicity of the chirps in the bird call is employed to develop a denoising filter which enhances the periodic structure of the target call by maximizing the value of a correlation-based function. Therefore, the proposed filter is called a Correlation-Maximization denoising filter. The coefficients of the filter are obtained through a gradient search approach which maximizes the value of a correlation-based function.

In the following sections, we analyze the characteristics of the Antbird calls, briefly review the Wiener filter and matched filter, design a Correlation-Maximization denoising filter, and develop a statistically-based bird call classification system.

We also discuss the advantages of the Correlation-Maximization filter over Wiener filtering in bird call denoising.

4.1 Antbird Call Analysis

Antbird calls were collected at a rainforest in Mexico by using a directional microphone.

Spectrograms of some Antbird calls are shown in Figures 4.1 - 4.5 . The following properties of the bird calls are observed:

Each call is composed of chirps. The chirp can be viewed as the smallest unit in the bird calls. Spectrograms of chirps are similar within a call, which implies that an Antbird repeats a similar vocalizing pattern in each call.

The number of chirps in the calls varies, even within the same species. In our database, an Antbird call is about 0.5 - 5 seconds long and contains 10 - 30 chirps.

The spectrum of the chirp has a harmonic structure. This structure is because the vocalization of the chirp can be viewed as a periodic airflow from the syrinx passing through the trachea. Much like the vocal folds in humans, the vibrating tympaniform membrane in the syrinx can control the airflow from the bronchus, which enables the bird to change tones (fundamental frequencies) of the chirps [65] [29]. Typically, the frequency range of the tones is between 500 to 6000 Hz. The transfer function of the trachea can be changed by loosening and tightening the surrounding muscles. Most high frequency components in the calls are weak because of radiation effects.

The recorded bird calls are corrupted by different kinds of noise. It is natural to find non-target calls overlapping in time with target calls. Cicadas, frog, and

other non-stationary animal sounds are common in a rainforest. Water flowing, wind blowing, and other quasi-stationary background noises are also common for a rainforest with a diverse landscape. Microphone friction, power buzzing and other instrumental noises are not avoidable for a hand-held recording device.

The changing location of the microphone also results in a variation in Signal-to-Noise Ratio (SNR). The highly varied acoustic condition can impose difficulty in training acoustic models and recognizing songs.

Considering the presence of the noise and varying SNRs, it is a necessity to perform denoising before extracting features. In the following section, different denoising methods are discussed.

4.2 Wiener Filtering

A prevailing denoising filter is the Wiener filter which estimates the additive noise spectrum and adaptively updates the frequency response of the denoising filter [30].

Suppose that the clean signal $x[n]$ and the noisy signal $y[n]$ are wide sense stationary, and $x[n]$ and the additive noise $v[n]$ are uncorrelated. After minimizing the mean square error, we have the relationship between the spectrum of estimated signal $\hat{x}[n]$ and noisy signal $y[n]$ denoted by $\hat{X}(f)$ and $Y(f)$:

$$\mathbb{E}[|\hat{X}(f)|^2] = |H(f)|^2 \mathbb{E}[|Y(f)|^2] \tag{4.1}$$

where $H(f)$ denotes the frequency response of the denoising filter $h(n)$, $\mathbb{E}[\cdot]$ denotes mathematical expectation operation. The estimate of the SNR at frequency f denoted by $\widehat{\mathrm{SNR}}(f)$ can be expressed as:

$$\widehat{\mathrm{SNR}}(f) = \frac{\mathbb{E}[|\hat{X}(f)|^2]}{\mathbb{E}[|\hat{V}(f)|^2]} \tag{4.2}$$

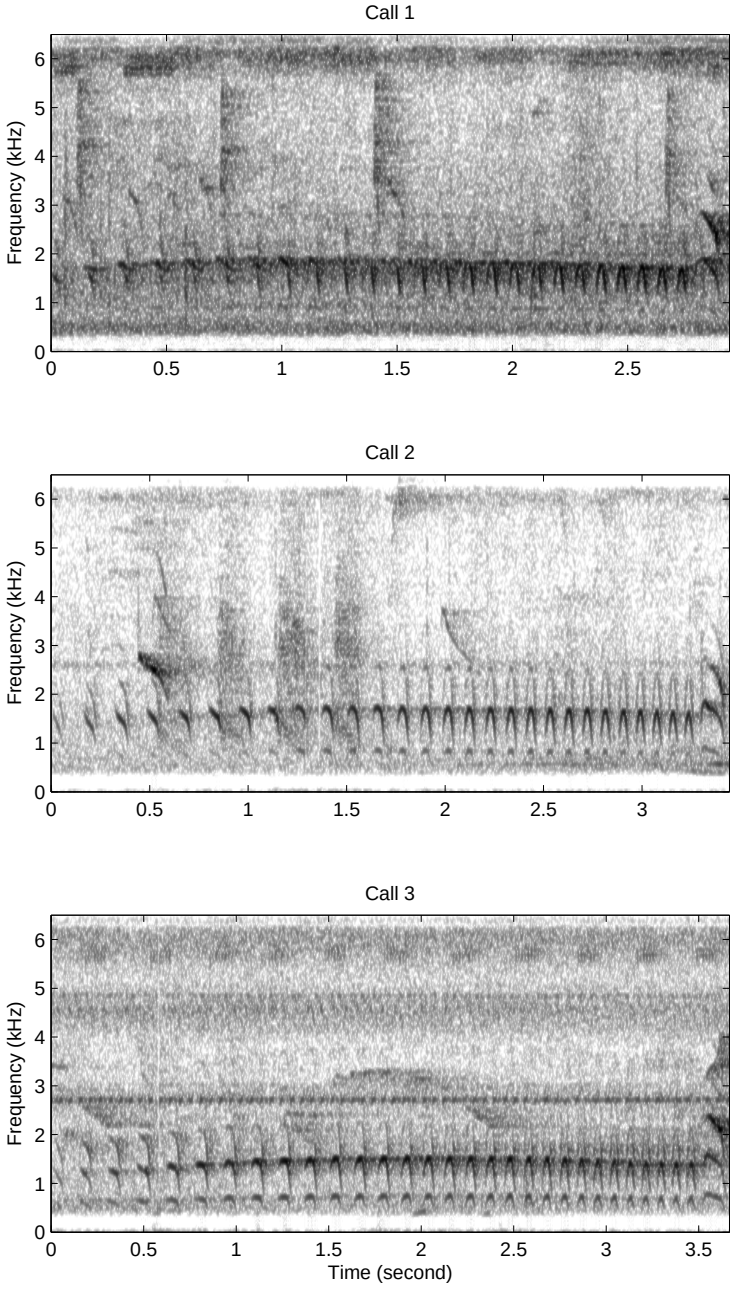

Figure 4.1: Spectrograms of 3 Barred Antshrike (BAS) calls

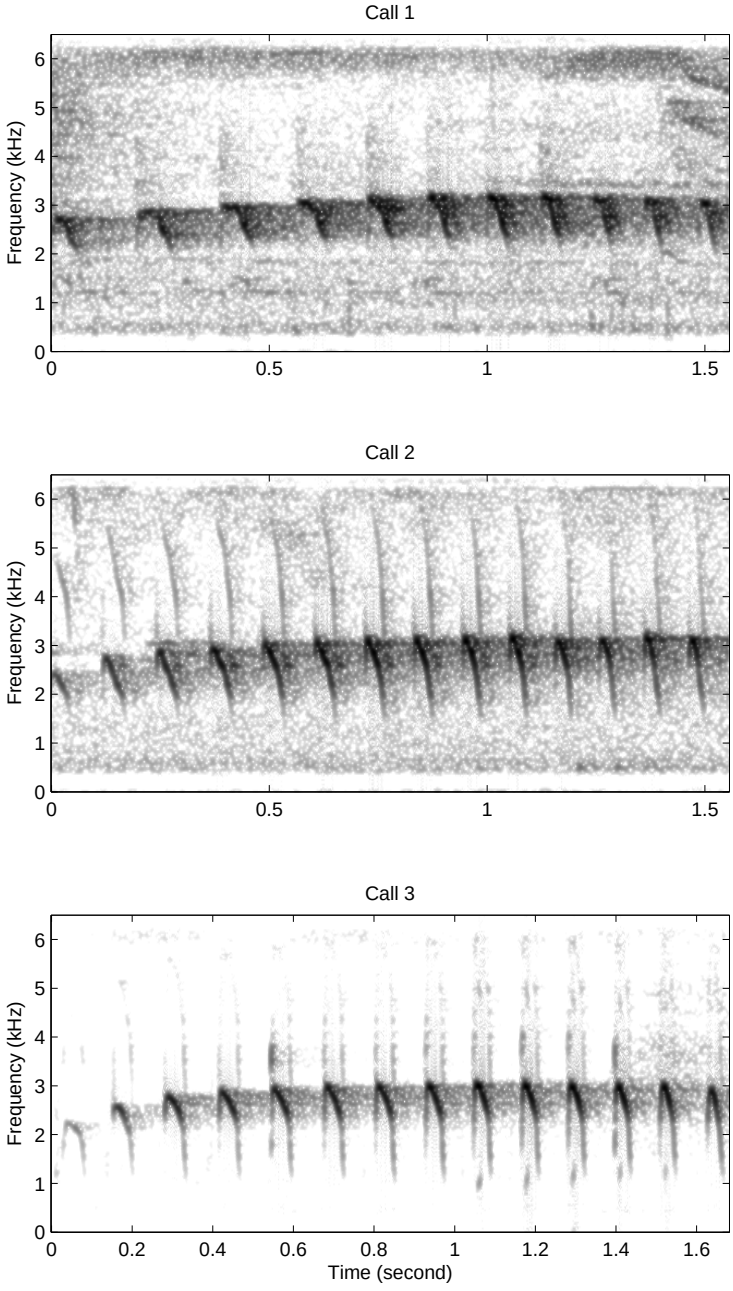

Figure 4.2: Spectrograms of 3 Dusky Antbird (DAB) calls

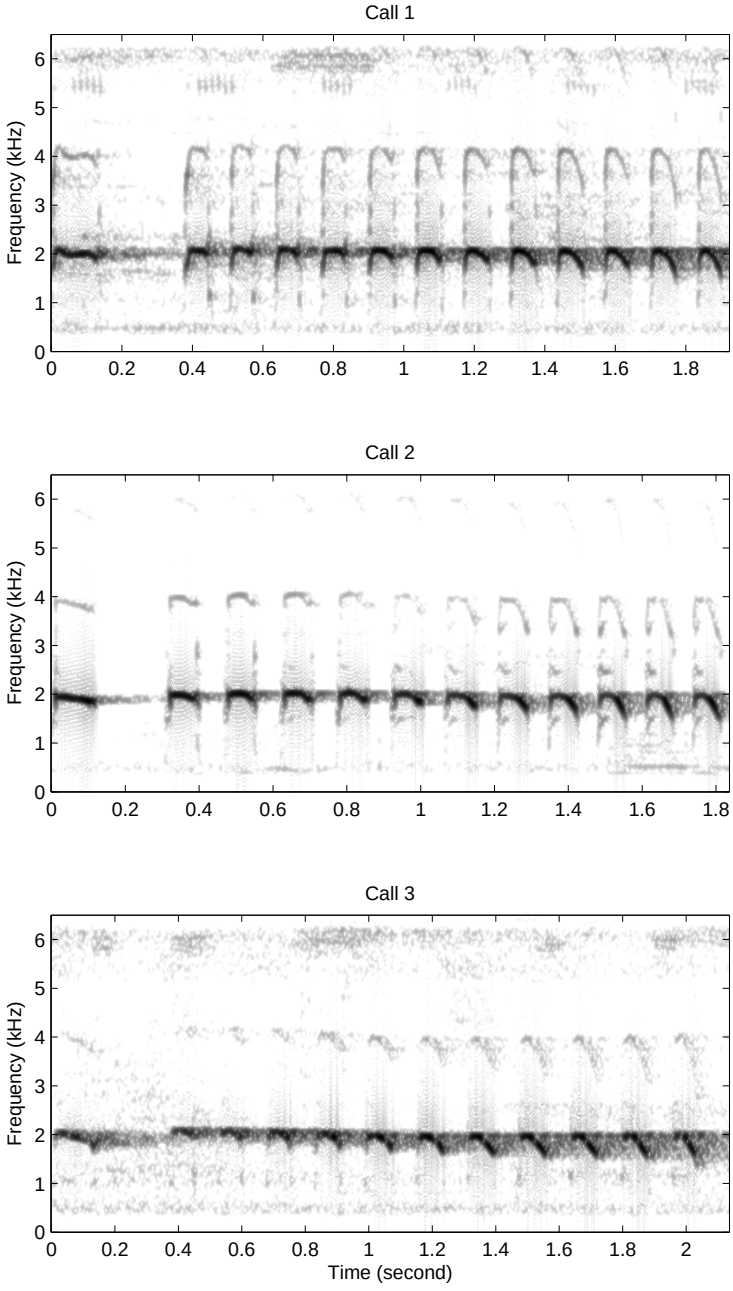

Figure 4.3: Spectrograms of 3 Mexican Antthrush (MAT) calls

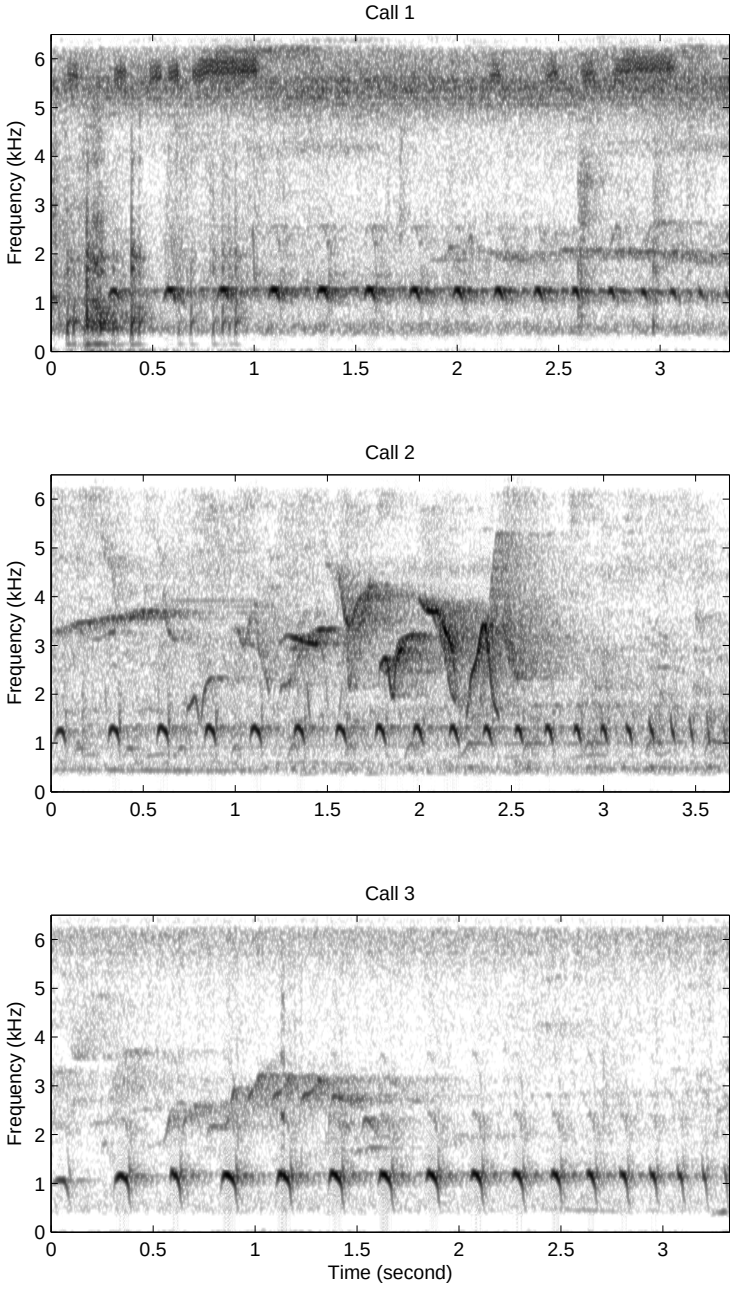

Figure 4.4: Spectrograms of 3 Great Antshrike (GAS) calls

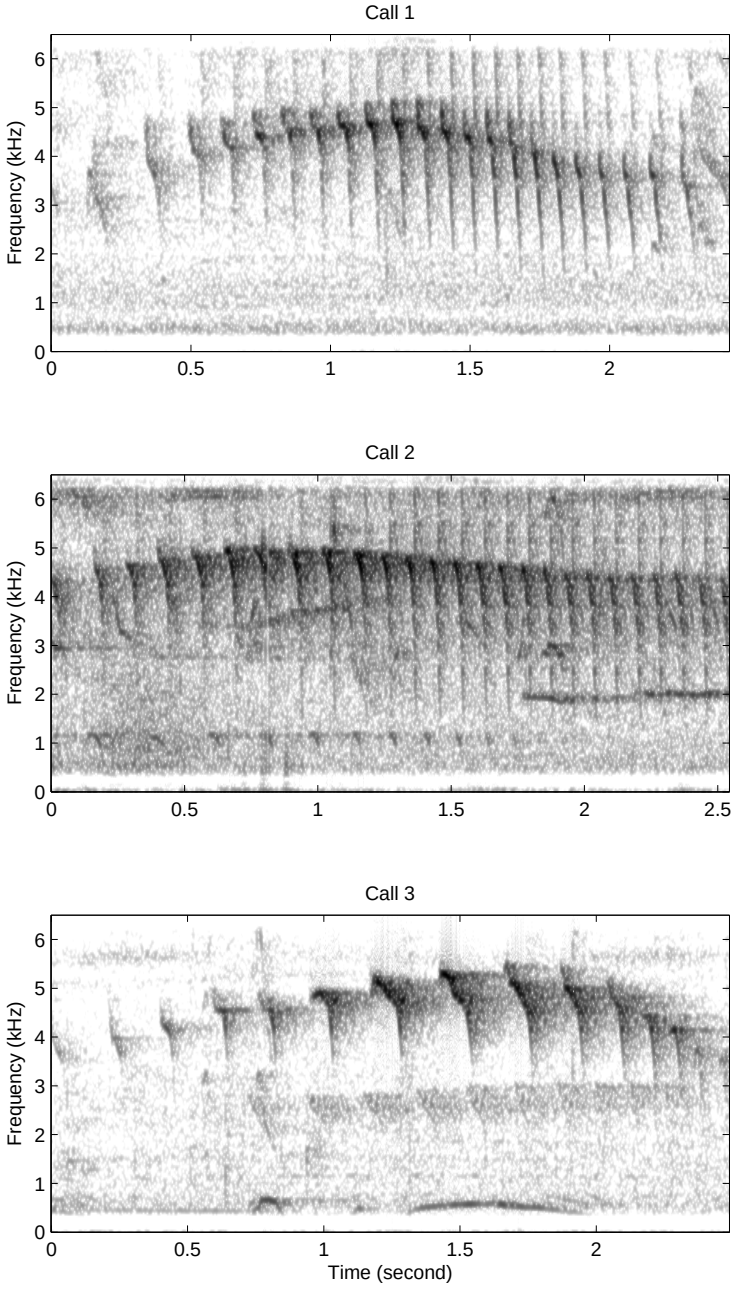

Figure 4.5: Spectrograms of 3 Dot-winged Antwren (DWA) calls

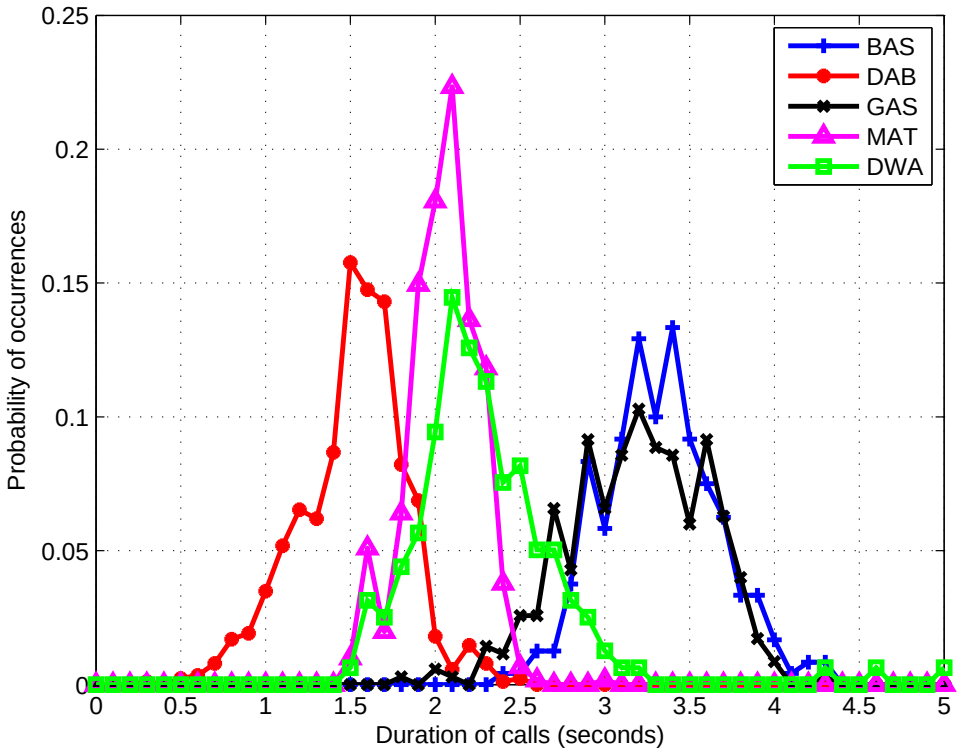

Figure 4.6: A histogram of bird-call duration

where $\hat{V}(f)$ is the estimated spectrum of the noise signal $v[n]$. Note that $\mathbb{E}[|Y(f)|^2] = \mathbb{E}[|\hat{X}(f)|^2] + \mathbb{E}[|\hat{V}(f)|^2]$ assuming $\hat{X}(f)$ and $\hat{V}(f)$ are orthogonal. The estimated clean spectrum can be expressed as:

$$\mathbb{E}[|\hat{X}(f)|^2] = \frac{\widehat{\mathrm{SNR}}(f)}{1 + \widehat{\mathrm{SNR}}(f)} \mathbb{E}[|Y(f)|^2] \tag{4.3}$$

Therefore, the non-causal Wiener filter converts the denoising problem into an SNR estimation problem [66].

According to our observation, the Wiener filter sometimes fails to identify background chirps as noise and boosts both the target and non-target chirps. It is necessary to develop a denoising filter which can not only enhance the target chirps but also suppress non-target chirps.

4.3 Matched Filtering

In matched filtering, a known signal, or template, is correlated with an unknown signal to detect the presence of the template in the unknown signal [64]. Given a clean signal $x[n]$, the impulse response of the matched filter denoted by $h[n]$ is defined as:

$$h[n] = kx[m - n] \tag{4.4}$$

where k and m are arbitrary constants. The matched filter is the optimal linear filter for maximizing the SNR in the presence of additive noise.

Let $v[n]$ denote the additive noise. The corrupted signal denoted by $y[n]$ can be expressed as: $y[n] = x[n] + v[n]$. The output of the matched filter denoted by

$\hat{x}[n]$ can be expressed as:

$$\begin{aligned} \hat{x}[n] &= \sum_{r=-\infty}^{\infty} y[r]h[n-r] \\ &= k\sum_{r=-\infty}^{\infty} x[r]x[r+m-n] + k\sum_{r=-\infty}^{\infty} v[r]x[r+m-n] \end{aligned} \tag{4.5}$$

Suppose $x[n]$ and $v[n]$ are both wide sense stationary and independent of each other, we have:

$$y[n] = kR_x[m-n] \tag{4.6}$$

where $R_x[m-n]$ denotes the autocorrelation function of $x[n]$. When $n=m$, the correlation function $R_x[m-n]$ is maximized.

The matched filter is usually used in radar and telecommunication for detection purpose. It is assumed that the clean signal, or the reference signal, is known. In the bird call denoising task, there is no need to perform the denoising if the reference signal is known; however, it is still possible to borrow the 'Correlation-Maximization' idea from the Eq. 4.6 to design a denoising filter which can enhance the target chirps and suppress non-target chirps discriminatively. The details of the proposed filter is discussed in the following section.

4.4 A Correlation-Maximization Denoising Filter

According to our observations, every Antbird call is quasi-periodic in terms of the interval between chirps, and the intervals slowly decrease with time. An example is shown in Fig. 4.7.

In the following, we will discuss how to search an optimal correlation denoising filter which can enhance this periodic structure.

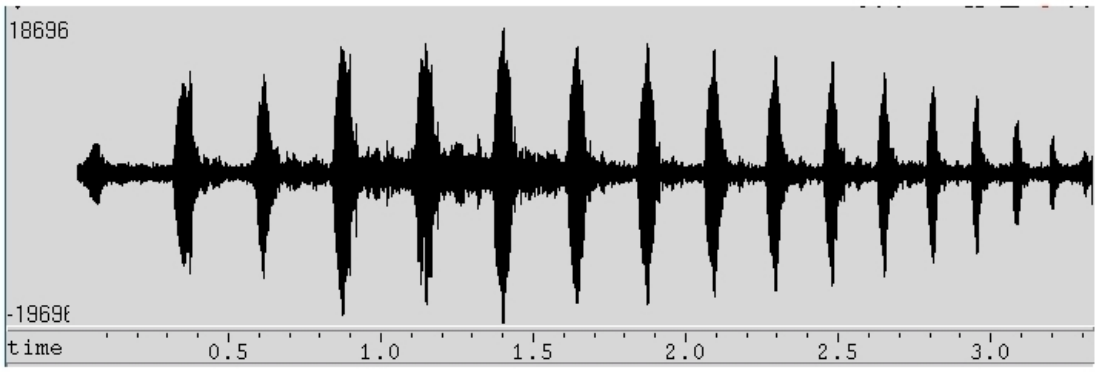

Figure 4.7: The waveform of a Great Antshrike (GAS) call

4.4.1 Search Chirp Interval Using a Correlation Function

Suppose a bird call $x[n]$ is corrupted by additive noise $v[n]$. An FIR filter denoted by $h[n]$ of L taps is used to obtain the estimate of the clean signal denoted by $\hat{x}[n]$ from the observed noisy signal denoted by $y[n]$. Then we have:

$$\hat{x}[n] = \sum_{k=1}^{L} h[k]y[n-k] \tag{4.7}$$

$y[n]$ is decomposed into M frames with a frame step size of Δ and a frame length of N, and we assume that $y[n]$ and $x[n]$ are wide sense stationary in each frame. Since the spectral distributions of different frames in a bird call are similar, a single $\mathbf{h}$ is assumed for each bird call. Therefore at frame m, the cross correlation function of $\hat{x}[n]$ at lag k denoted by $\phi_{\hat{x}}^{m}[0,k]$ can be expressed as:

$$\begin{aligned}\phi_{\hat{x}}^{m}[0,k] &= \sum_{n=m\Delta}^{m\Delta+N-1-k} \hat{x}[n]\hat{x}[n+k] \\ &= \sum_{p=1}^{L} h[p] \sum_{q=1}^{L} h[q] \sum_{n=m\Delta}^{m\Delta+N-1-k} y[n-p]y[n+k-q].\end{aligned} \tag{4.8}$$

Note that the lag k has K possible values, $k = k_0, k_1, ...k_{K-1}$. We can define an $L \times L$ cross correlation function matrix $\mathbf{\Phi}_y^m[0,k]$ for frame m at lag k. The element of $\mathbf{\Phi}_y^m[0,k]$ in row p and column q is expressed as:

$$\mathbf{\Phi}_y^m[0,k]_{pq} = \sum_{n=m\Delta}^{m\Delta+N-1-k} y[n-p]y[n+k-q]. \tag{4.9}$$

Therefore we have

$$\phi_{\hat{x}}^m[0,k] = \mathbf{h}^T \mathbf{\Phi}_y^m[0,k]\mathbf{h} \tag{4.10}$$

where $\mathbf{h} = [h[0], h[1], \cdots, h[L]]^T$ denotes the coefficients of the FIR filter. To confine the dynamic range of $\phi_{\hat{x}}^m[0,k]$, the normalized cross correlation function $\bar{\phi}_{\hat{x}}^m[0,k]$ is expressed as follows:

$$\begin{aligned}
\bar{\phi}_{\hat{x}}^m[0,k] &= \frac{\sum_{n=m\Delta}^{m\Delta+N-1-k} \hat{x}[n]\hat{x}[n+k]}{\sqrt{\sum_{n=m\Delta}^{m\Delta+N-1-k} \hat{x}^2[n]}\sqrt{\sum_{n=m\Delta}^{m\Delta+N-1-k} \hat{x}^2[n+k]}} \\
&= \frac{\phi_{\hat{x}}^m[0,k]}{\sqrt{\phi_{\hat{x}}^m[0,0]\phi_{\hat{x}}^m[k,k]}} \\
&= \frac{\mathbf{h}^T\mathbf{\Phi}_y^m[0,k]\mathbf{h}}{\sqrt{\mathbf{h}^T\mathbf{\Phi}_y^m[0,0]\mathbf{h}}\sqrt{\mathbf{h}^T\mathbf{\Phi}_y^m[k,k]\mathbf{h}}}.
\end{aligned} \tag{4.11}$$

Note that $\bar{\phi}_{\hat{x}}^m[0,k] \in [-1,1]$.

It is possible to find the chirp interval in each frame over the denoised signal $\hat{x}[n]$. Dynamic programming can be used to minimize the distortion induced by background noise in the chirp interval search [6]. Because the objective of the dynamic programming is to search the path which has a minimum accumulative cost. The local cost of frame m at lag k is defined as $-\bar{\phi}_{\hat{x}}^m[0,k]$. Since the chirp interval is gradually decreasing over time, the cost of transitioning from lag k_i to

k_j denoted by $d(k_i, k_j)$is defined as follows:

$$d(k_i, k_j) = e^{\alpha|k_i - \delta - k_j|} - 1 \qquad i, j = 0, \cdots, K-1 \tag{4.12}$$

where α and δ are pre-set empirically. α is a scaling factor, and δ is an estimate of how fast the chirp interval changes per second. This exponential function can impose more penalty than its linear counterpart on the transition cost in order to prevent chirp intervals from greatly varying between two consecutive frames. Note that when $k_j = k_i - \delta$, $d(k_i, k_j) = 0$.

A trellis structure of $K \times M$ for dynamic programming is developed, where M is the number of total frames, K is the number of possible candidates at each frame. $\mathbf{s} = [s_1, s_2, \cdots, s_M]$ is used to denote an arbitrary valid path in the trellis.

4.4.2 Search The Optimal Denoising Filter

We search the filter coefficients in a grid by minimizing a correlation-based cost function.

It can be assumed that an optimal filter $\mathbf{h}$ can enhance the periodic structure of the target bird call and remove the additive noise so that the minimum accumulative cost is achieved in the chirp interval search over the denoised signal. This assumption can be expressed as:

$$\mathbf{h}^* \quad = \quad \arg\min_{\mathbf{h}} \mathcal{F}(\mathbf{h}, \mathbf{s}) \tag{4.13}$$

where $\mathbf{h}^*$ denotes the optimal denoising filter, the accumulative cost $\mathcal{F}(\mathbf{h}, \mathbf{s})$ which is the summation of the accumulative local and transition costs is expressed as:

$$\mathcal{F}(\mathbf{h}, \mathbf{s}) = \sum_{m=1}^{M} -\bar{\phi}_{\hat{x}}^{m}[0, s_m] + \sum_{m=1}^{M-1} d(s_m, s_{m+1}). \tag{4.14}$$

The gradients of $\mathcal{F}(\mathbf{h}, \mathbf{s})$ w.r.t. $\mathbf{h}$ can be expressed as:

$$
\begin{aligned}
& \nabla_{\mathbf{h}} \mathcal{F}(\mathbf{h}, \mathbf{s}) \qquad (4.15) \\
= \; & \nabla_{\mathbf{h}} \{ - \sum_{m=1}^{M} \frac{\mathbf{h}^T \mathbf{\Phi}_y^m[0, s_m] \mathbf{h}}{\sqrt{\mathbf{h}^T \mathbf{\Phi}_y^m[0,0] \mathbf{h}} \sqrt{\mathbf{h}^T \mathbf{\Phi}_y^m[s_m, s_m] \mathbf{h}}} \} + \mathbf{0} \\
= \; & - \sum_{m=1}^{M} \frac{1}{\sqrt{\mathbf{h}^T \mathbf{\Phi}_y^m[0,0] \mathbf{h} \cdot \mathbf{h}^T \mathbf{\Phi}_y^m[s_m, s_m] \mathbf{h}}} \cdot \\
& \big[\frac{\mathbf{\Phi}_y^m[0, s_m] + \mathbf{\Phi}_y^m[0, s_m]^T}{\mathbf{h}^T \mathbf{\Phi}_y^m[0, s_m] \mathbf{h}} - \frac{1}{2} \frac{\mathbf{\Phi}_y^m[0,0] + \mathbf{\Phi}_y^m[0,0]^T}{\mathbf{h}^T \mathbf{\Phi}_y^m[0,0] \mathbf{h}} \\
& - \frac{1}{2} \frac{\mathbf{\Phi}_y^m[s_m, s_m] + \mathbf{\Phi}_y^m[s_m, s_m]^T}{\mathbf{h}^T \mathbf{\Phi}_y^m[s_m, s_m] \mathbf{h}} \big] \mathbf{h}
\end{aligned}
$$

Therefore, the gradient descent method can be used to search the optimal filter $\mathbf{h}^*(\mathbf{s})$ for a path $\mathbf{s}$. The minimum cost is achieved when $\nabla_{\mathbf{h}} \mathcal{F}(\mathbf{h}, \mathbf{s}) = \mathbf{0}$. Note that $\mathbf{s}$ is independent of $\mathbf{h}$. The final optimal filter $\mathbf{h}^*$ can be searched using a brute-force method:

Algorithm 4.4.1: BRUTE-FORCE FILTER SEARCH ($\mathbf{h}$)

Set the iteration time $= I$, the iteration stopping threshold $= \epsilon$.

for all valid $\mathbf{s}$

do
- Initialize $\mathbf{h}_0(\mathbf{s}) = [1, 0, \cdots, 0]^T$.
- **for** $i = 0$ **to** I
  - **do**
    - $\mathbf{h}_{i+1}(\mathbf{s}) = \mathbf{h}_i(\mathbf{s}) - t_i \nabla_{\mathbf{h}} \mathcal{F}(\mathbf{h}_i(\mathbf{s}), \mathbf{s})$,
    - where t_i is the step size;
    - **if** $||\mathbf{h}_{i+1}(\mathbf{s}) - \mathbf{h}_i(\mathbf{s})||/||\mathbf{h}_i(\mathbf{s})|| < \epsilon$
      - **then** $\mathbf{h}^*(\mathbf{s}) = \mathbf{h}_i(\mathbf{s})$, **break** .
- **if** i equals I
  - **then** $\mathbf{h}^*(\mathbf{s}) = \mathbf{h}_I(\mathbf{s})$.

$\mathbf{s}^* = \arg\min_{\mathbf{s}} \mathcal{F}(\mathbf{h}^*(\mathbf{s}), \mathbf{s})$

$\mathbf{h}^* = \mathbf{h}^*(\mathbf{s}^*)$, **exit** .

Symbol List:

$\mathbf{s}^*$: the optimal path;

$\mathbf{h}^*$: the optimal filter;

$\mathbf{h}_i(\mathbf{s})$: the searched filter at i_{th} iteration given the path $\mathbf{s}$;

$\mathbf{h}^*(\mathbf{s})$: the optimal filter given the path $\mathbf{s}$;

4.4.3 Speed Up The Search: N-best Search

Instead of a grid search, we propose to search through a trellis similar to the N-best search in ASR.

There are K^M possible paths in a $K \times M$ trellis. Suppose the average iteration time of the gradient search is $\bar{I}$, this brute-force approach needs $K^M \times \bar{I}$ iterations which is computationally unacceptable.

If the gradient search stopped at iteration i, the optimal path among all the valid paths denoted by $\mathbf{s}_i^*$ can be expressed as:

$$\mathbf{s}_i^* = \arg\min_{\mathbf{s}} \mathcal{F}(\mathbf{h}_i(\mathbf{s}), \mathbf{s}). \tag{4.16}$$

Since $\mathbf{s}_i^*$ may not be equal to $\mathbf{s}^*$ (obtained from the previous section), we need to search through all possible paths; however, we can assume that $\mathbf{s}^*$ is within a path subset during each iteration. The subset is composed of the top N-best paths which are the output of the dynamic programming on the trellis. That means the gradient descent search only needs to be applied to the N-best paths, not all the paths at each iteration. Then the brute-force search approach can be improved into an N-best search:

Algorithm 4.4.2: N-BEST FILTER SEARCH($\mathbf{h}$)

Set the iteration time $= I$, the iteration stopping threshold $= \epsilon$;

Set the N-best path number $= J$.

for $j = 0$ **to** J

  do Initialize an N-best (J) filter list $\mathbf{h}_0^j = [1, 0, \cdots, 0]^T$.

for $i = 0$ **to** I

do
- **for** $j = 1$ **to** J
  - **do**
    - Use $\mathbf{h}_i^j$ to build j_{th} trellis by calculating $\bar{\phi}_{\hat{x}}^m[k]$.
    - Search N-best (J) paths $\mathbf{s}_i^{(j,k)}$ in j_{th} trellis.
    - **for** $k = 1$ **to** J
      - **do**
        - $\mathbf{h}_{i+1}^{(j,k)} = \mathbf{h}_i^j - t_i \nabla_{\mathbf{h}} \mathcal{F}(\mathbf{h}_i^j, \mathbf{s}_i^{(j,k)})$,
        - where t_i is the step size.
- Sort $\mathbf{h}_{i+1}^{(j,k)}$, $j, k = 1, \cdots, J$, according to the values of $\mathcal{F}(\mathbf{h}_{i+1}^{(j,k)}, \mathbf{s}_i^{(j,k)})$ in ascending order to obtain $\tilde{\mathbf{h}}_{i+1}^l$, $l = 1, \cdots, J^2$
- **for** $j = 1$ **to** J
  - **do** $\mathbf{h}_{i+1}^j = \tilde{\mathbf{h}}_{i+1}^j$.
- **if** $\max\limits_{j=1\ldots J} \dfrac{||\mathbf{h}_{i+1}^j - \mathbf{h}_i^j||}{||\mathbf{h}_i^j||} < \epsilon$
  - **then** $\mathbf{h}^* = \mathbf{h}_i^1$, **exit** .

if $i == I$

  then $\mathbf{h}^* = \mathbf{h}_I^1$, **exit** .

Symbol List:

$\mathbf{h}^*$: the optimal filter; $\mathbf{h}_i^j$: the j_{th} N-best filters at i_{th} iteration;

$\mathbf{s}_i^{(j,k)}$: the k_{th} best path in the j_{th} trellis at i_{th} iteration;

$\mathbf{h}_{i+1}^{(j,k)}$: the searched filter given $\mathbf{h}_i^{(j,k)}$ and $\mathbf{s}_i^{(j,k)}$;

$\tilde{\mathbf{h}}_{i+1}^l$: a sorted filter list of $\mathbf{h}_{i+1}^{(j,k)}$.

Table 4.1: Number of bird calls in the training and test sets. BAS: Barred Antshrike; DAB: Dusky Antbird; GAS: Great Antshrike; MAT: Mexican Antthrush; DWA: Dot-winged Antwren.

	BAS	DAB	GAS	MAT	DWA	Total
Training	240	888	350	609	159	2246
Testing	120	444	175	304	77	1120

Although $J \times \bar{I}$ trellis building and dynamic programming operations are newly introduced in this N-best search approach, the total average gradient search iterations is reduced to $J^2 \times \bar{I}$ compared to the $K^M \times \bar{I}$ iterations in the brute-force search approach when the M is large. Typically, for Antbird calls, $K = 49$, $1 \leq M \leq 50$, $J = 20$.

4.5 Experiments

The Antbird call corpus contains 3366 bird calls from 5 species. We split the corpus into a training and testing set with a ratio of 2:1 as shown in Table 4.5. The training set is 85 minutes long and the testing set is 42 minutes long. The calls are 0.5 - 5.0 seconds long.

The original single-channel acoustic signal is collected at a sampling rate of 44.1 kHz. The frequency range of the bird calls is from 500 to 6000 Hz. Thus, we use a band-pass filter with cutoff frequencies between 360 Hz and 6500 Hz to remove irrelevant frequency components. The signal is then downsampled to 16 kHz.

In the Correlation-Maximization denoising filter, the number of the filter taps (L) is 20. Since an analysis frame should contain at least two chirps to extract

the chirp interval, and the bird chirp length ranges from 60 to 300 ms, the frame length is 600 ms, i.e. 9600 samples. The frame step size is 100 ms, and the correlation lag step is 5 ms. The number of lags is (300 - 60)/5 + 1 = 49. The maximum number of iterations (I) in the gradient search, and the number of N-best paths (J) are both 20. According to the experimental results, increasing L, I, or J does not boost the classification accuracy but does increase the computational cost.

A 39-dimension feature composed of the first 13 MFCCs and first and second derivatives is computed every frame for model training and testing.

In the GMM classifier, each species' model is set to have 256 Gaussians. In the HMM-based classifier, each species' model also has 256 Gaussians per state. The recognition network is the same as the one used in isolated word recognition, in which each species corresponds to a word node. Choosing the correct state number may enable finer modeling of a bird call. Since the number of chirps in a bird call varies and the state numbers are the same for all 5 species, state number 6, which is the minimum number of chirps in all the bird calls, is used for each species model and it also results in the lowest classification error rate among state numbers 1 to 9.

Classification results are shown in Table 4.2. The HMM-based classifier results in better performance than the GMM classifier when using the same features. After applying the Correlation-Maximization denoising filter, classification error rates of both GMM and HMM-based classifiers are lower than their counterparts using the Wiener filer. Since the Correlation-Maximization filter uses a long frame length to estimate the slow-varying noise and to capture interval periodicity, and the Wiener filter employs a relatively short frame length to track the fast changing noise, it is possible that cascading the two filters can further reduce error rates.

Table 4.2: Classification error rate (%) on the test set. **W+/CM+**: feature extraction using the output of the Wiener/Correlation-Maximization based denoising filter

	GMM	HMM
MFCC	8.7	5.4
W+MFCC	5.9	4.9
CM+MFCC	5.3	4.6
CM+W+MFCC	**4.7**	**4.1**

Table 4.3: The confusion matrix of using CM+W+MFCC feature and HMM based classifier on the test set; **RE**: the number of errors divided by the total number of calls in the row; **PE**: the number of errors in the row divided by the total number of calls.

	BAS	DAB	GAS	MAT	DWA	RE(%)	PE (%)
BAS	120	0	0	0	0	0.0	0.0
DAB	1	430	7	5	1	3.2	1.2
GAS	6	3	149	17	0	14.9	2.3
MAT	0	0	0	304	0	0.0	0.0
DWA	0	4	2	0	71	7.8	0.5

As shown in Table 4.3, the confusion matrix is used to analyze the classification errors. The calls of BAS, MAT, and DAB are less likely to be misclassified as other species compared to those of GAS and DWA. The GAS→MAT errors (1.5%) accounted for more than 35% in the total errors (4.1%).

A GAS bird call is used to illustrate the difference between the Wiener and Correlation-Maximization filter. From Fig. 4.8 (a), other bird chirps are observed from 0.6 to 1.6 seconds, and background noise can act as adverse factors to the classification task. As shown in Fig. 4.8 (b), both target and non-target bird chirps are enhanced after Wiener filtering. That is because Wiener filter can not denoise discriminatively. It can be seen from Fig. 4.8 (c) that the Correlation-Maximization filter can suppress the non-target chirps while enhancing the target chirps. That is because the Correlation-Maximization filter is supposed to only enhance the periodic structure of the target bird call. It is also shown in Fig. 4.8 (d) that both non-target bird call and background noise are suppressed when cascading the Wiener filter and the Correlation-Maximization filter.

The frequency response of the optimal Correlation-Maximization denoising filter for this bird call is shown in Fig. 4.9. The filter has a pass-band from 800 to 1750 Hz, which can enhance the frequency components of the target bird call, a stop-band from 2600 to 8000 Hz, and a dip around 2800 Hz can minimize the interference introduced by background noise and other bird chirps. Other filters were developed for other bird calls.

We also studied how the classification error rates changes when the training set size and the number of Gaussians per state in the HMM-based classifier are changed, which is shown in Fig. 4.10. In this study, the number of states is fixed to 6 and the features are CM+W+MFCC. For a training set ratio less than 1, the sub training set is created by randomly selecting calls in each species according to

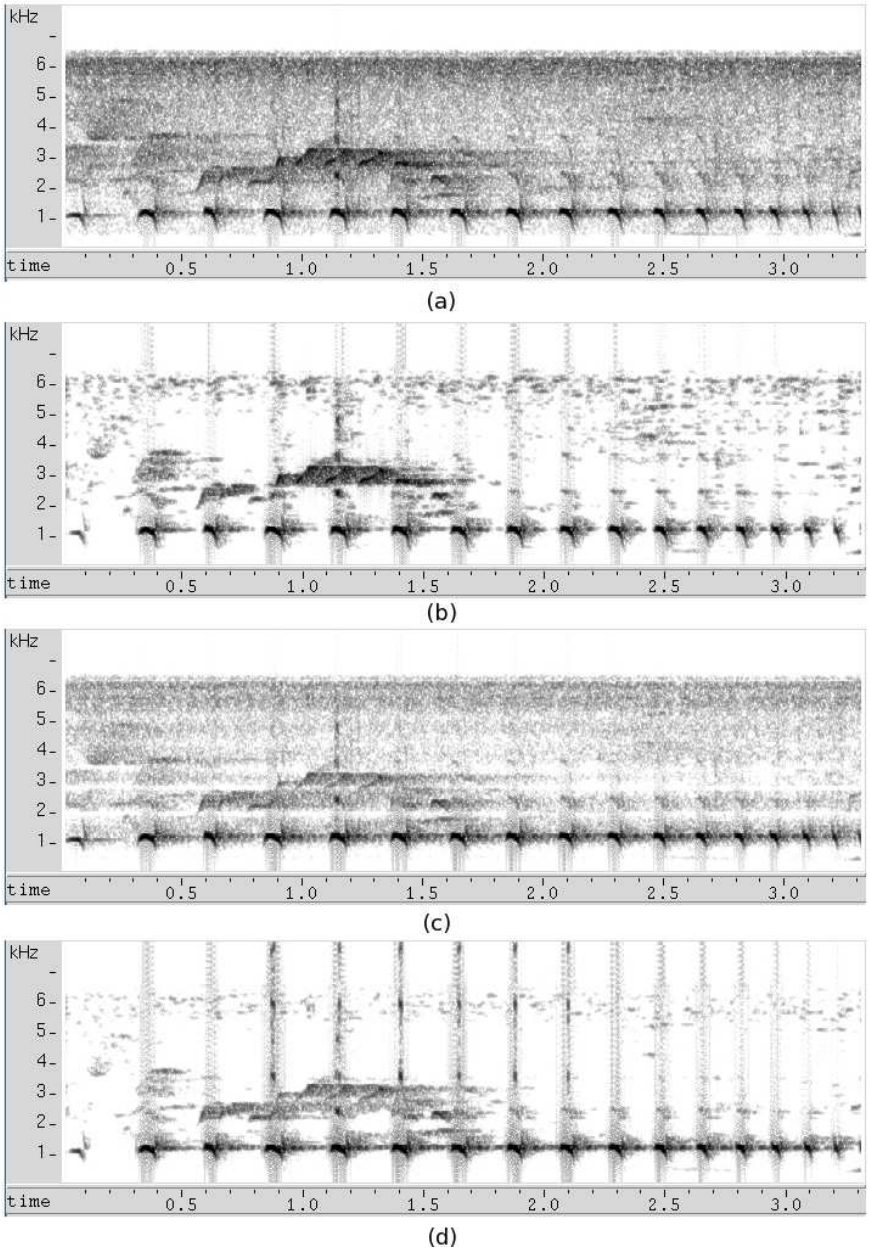

Figure 4.8: A Great Antshrike (GAS) call: (a) original spectrogram; (b) spectrogram after Wiener filtering; (c) spectrogram after Correlation-Maximization filtering; (d) spectrogram after Wiener and Correlation-Maximization filtering.

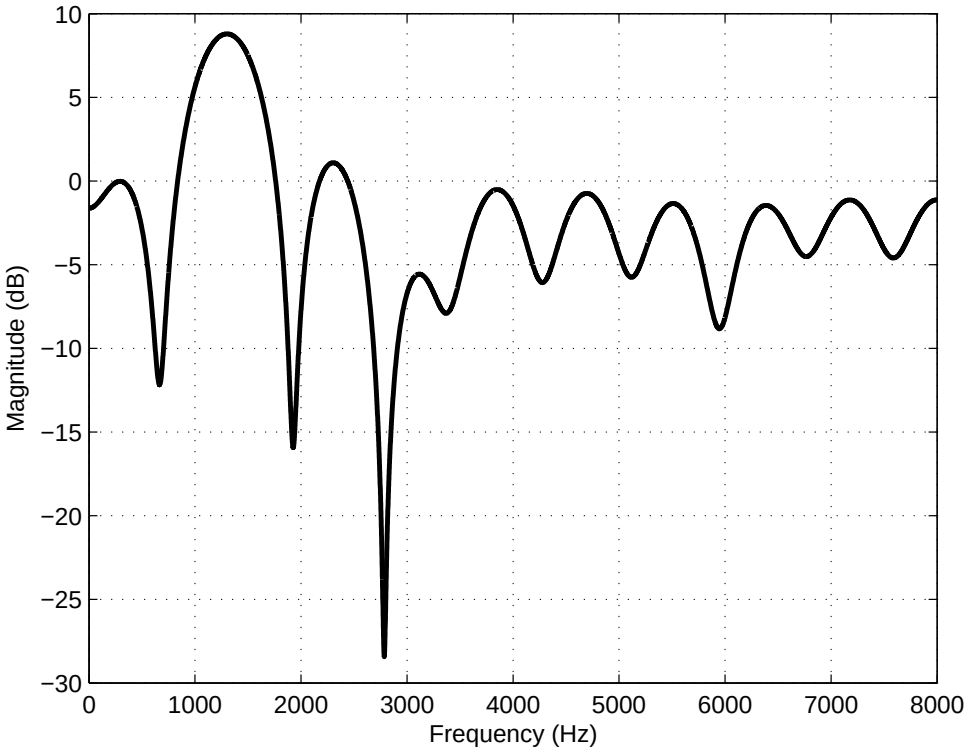

Figure 4.9: The frequency response of the Correlation-Maximization filter for a GAS call.

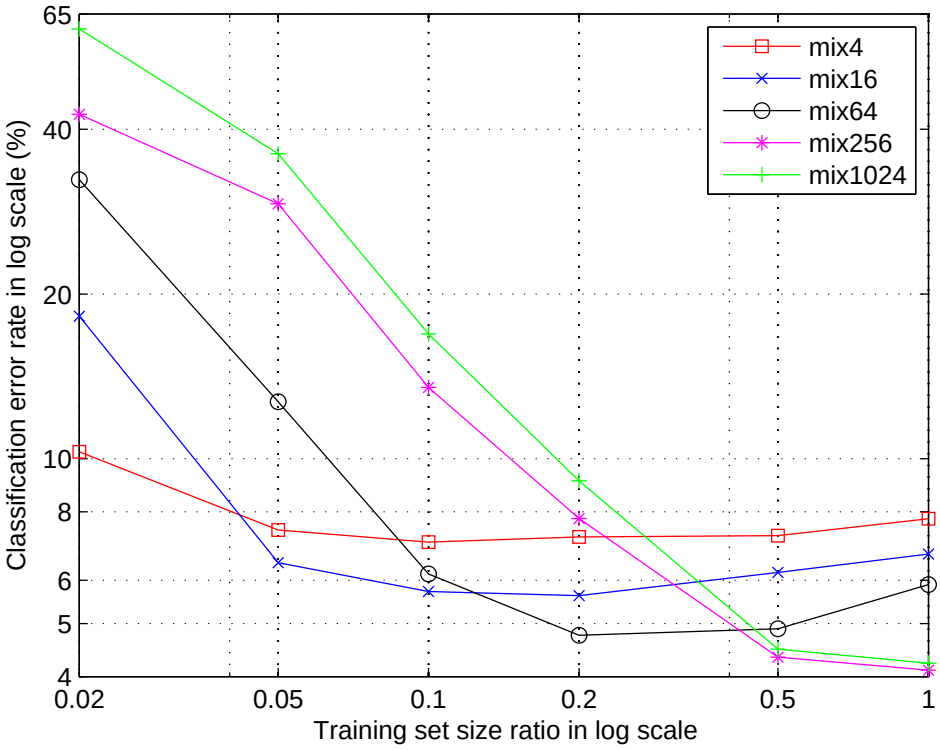

Figure 4.10: The relationship between the ratio of the training set size to the original one, the number of Gaussians per state, and the classification error rate. The feature is fixed as CM+W+MFCC. The number of states in HMM is fixed to 6.

the ratio. To reduce the uncertainty of the classification error rates, the training and testing routines are repeated 20 times. The mean of the error rates is used to represent the error rate under the training set ratio. It can be observed that the optimal number of Gaussians per state decreases as the size of training set decreases. The phenomena is probably because of the model overfitting. Thus, it is important to control the model complexity when the size of the training set is changed.

4.6 Conclusions

The Correlation-Maximization denoising filter is effective in enhancing target bird calls with a quasi-periodic structure in the time domain and suppressing non-target bird calls and other non-stationary noises, which results in a reduction in classification error rate.

Compared to the Wiener filter, the Correlation-Maximization filter avoids estimating the SNR by using the periodicity of the target bird call, and it does not assume that the noise is stationary. Combining both filters can further improve the classification accuracy compared to the system only using the Correlation-Maximization filter.

CHAPTER 5

fbEM: a Filter Bank EM Algorithm

In this chapter, the optimal center frequencies and bandwidths of the filter bank used for Mel-frequency cepstral coefficients (MFCCs) extraction are searched in an efficient statistically-based approach. Since the auxiliary function in the EM algorithm is extended for optimizing not only model parameters, but also parameters of the filter bank used in feature extraction, the proposed algorithm is called the filter bank EM (fbEM) algorithm. Note that statistically-based non-uniform DFT analysis/synthesis filter banks have been explored before to reduce spectral-domain distortion in speech coding [67].

The organization of the chapter is as follows. First, joint filter bank and model parameters optimization using the fbEM algorithm is presented. Then, experimental results on an Antbird corpus are analyzed.

5.1 Optimizing the Filter Bank in Feature Extraction

The procedure and parameters of cepstral feature extraction are the same as the MFCC extraction except for the parameters of the filter bank. In the new filter bank shown in Fig. 5.1, it is assumed that the number of filters is fixed as L, the shape of each filter is triangular, the gain of each filter is the same, and the center frequency of each filter is equal to the low and high cut-off frequencies of its right and left filters, respectively. $\boldsymbol{\alpha} = [\alpha_0, \cdots, \alpha_{L+1}]^T$ is used to represent the

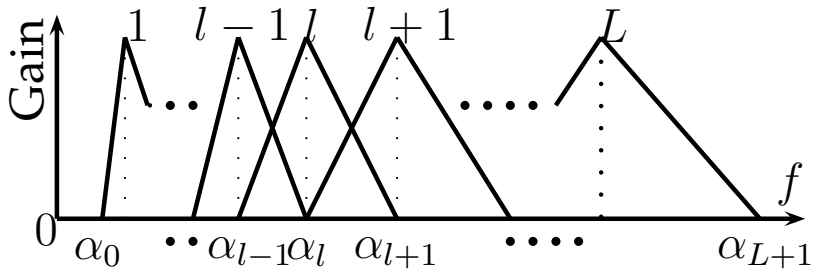

Figure 5.1: The frequency response of the filter bank used in feature extraction. L is the number of filters. The letter on top of each filter denotes the filter index.The gain of each filter is the same.

parameters of the filter bank, where α_l, $l = 1 \cdots L$, denotes the center frequency of filter l, α_0 and α_{L+1} denote the low and high cut-off frequencies of the filter bank, respectively. Audio signals denoted by $\mathbf{x}$ is compressed to a sequence of column feature vectors denoted by $\mathbf{Y}$ which can be represented as $\{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$, where T is the number of the frames. The procedure of feature extraction can be viewed as a function denoted by $f_{\boldsymbol{\alpha}}$ from $\mathbf{x}$ to $\mathbf{Y}$, i.e. $\mathbf{Y} = f_{\boldsymbol{\alpha}}(\mathbf{x})$.

If the feature sequence is assumed to be independent and identically distributed within each class, a Gaussian mixture model (GMM) can be used to model the distribution of the homogeneous data.

Let $\boldsymbol{\alpha}$ and $\mathcal{M}$ denote the current feature extraction and model parameters, respectively. Let $\bar{\boldsymbol{\alpha}}$ and $\bar{\mathcal{M}}$ denote the feature extraction and model parameters to be estimated, respectively. $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathcal{M}}$ are used to denote the estimated feature extraction and model parameters, respectively.

The proposed fbEM algorithm is described in Algorithm 1.

Algorithm 1. fbEM: joint filter bank and model parameter optimization using the EM algorithm

Step 1: Initialization: initialize the filter bank parameters $\boldsymbol{\alpha}$; extract feature $\mathbf{Y}$, i.e. $f_{\boldsymbol{\alpha}}(\mathbf{x})$ from acoustic signals $\mathbf{x}$; train an initial model $\mathcal{M}$ from $\mathbf{Y}$ using the conventional EM algorithm.

Step 2: Constrained Filter bank optimization without updating the model $\mathcal{M}$:

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= \arg\max_{\bar{\boldsymbol{\alpha}}} \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\}) \\ \text{s.t.} \quad & \alpha_{\min} \leq \bar{\alpha}_0 < \cdots < \bar{\alpha}_{L+1} \leq \alpha_{\max} \end{aligned} \tag{5.1}$$

$\hat{\boldsymbol{\alpha}}$ is solved as follows: initialize $\bar{\boldsymbol{\alpha}}$ to be $\boldsymbol{\alpha}$, then update $\bar{\boldsymbol{\alpha}}$:

$$\bar{\boldsymbol{\alpha}} \leftarrow \bar{\boldsymbol{\alpha}} + \eta \frac{\partial \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\})}{\partial \bar{\boldsymbol{\alpha}}}$$

where η denotes the step size, and $\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\})$ is an auxiliary function defined in Eq. 5.5. Extract $\bar{\mathbf{Y}}$ using the updated $\bar{\boldsymbol{\alpha}}$, i.e. $\bar{\mathbf{Y}} = f_{\bar{\boldsymbol{\alpha}}}(\mathbf{x})$, repeat until the increment of $\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\})$ falls below a certain threshold. Then let $\hat{\boldsymbol{\alpha}} = \bar{\boldsymbol{\alpha}}$, $\hat{\mathbf{Y}} = \bar{\mathbf{Y}}$.

Step 3: Estimate model parameters without updating the filter bank $\hat{\boldsymbol{\alpha}}$ and feature $\hat{\mathbf{Y}}$, i.e. $f_{\hat{\boldsymbol{\alpha}}}(\mathbf{x})$:

$$\hat{\mathcal{M}} \quad = \quad \arg\max_{\bar{\mathcal{M}}} \mathcal{Q}(\{\hat{\boldsymbol{\alpha}}, \mathcal{M}\}, \{\hat{\boldsymbol{\alpha}}, \bar{\mathcal{M}}\}) \tag{5.2}$$

which is the same as the conventional EM algorithm [43].

Step 4: Convergence or keep iterating: if

$$\frac{|\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\hat{\boldsymbol{\alpha}}, \hat{\mathcal{M}}\}) - \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\boldsymbol{\alpha}, \mathcal{M}\})|}{|\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\boldsymbol{\alpha}, \mathcal{M}\})|} \geq \epsilon \tag{5.3}$$

where ϵ denotes the threshold, then $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}$, $\mathcal{M} = \hat{\mathcal{M}}$, go to Step 2; else stop and exit.

As shown in Algorithm 1, the auxiliary function $\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \bar{\mathcal{M}}\})$ of the fbEM algorithm has both feature extraction and model parameters as variables. In conventional EM algorithm, the auxiliary function only has model parameters as variables. In fbEM algorithm, since $\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\boldsymbol{\alpha}, \mathcal{M}\}) \leq \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\hat{\boldsymbol{\alpha}}, \mathcal{M}\})$ $\leq \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\hat{\boldsymbol{\alpha}}, \hat{\mathcal{M}}\})$, which are illustrated in Steps 2 and 3, the increase of the auxiliary function is guaranteed.

The details of Algorithm 1 are as follows.

5.1.1 Filter bank $\boldsymbol{\alpha}$ and model $\mathcal{M}$ initialization

In Step 1, it is important to choose a good initial guess to the solution for an iterative method like the EM algorithm. Graciarena et al [49] showed that a Mel-scaled filter bank results in a higher bird call verification accuracy compared to the linear-scaled counterpart. In this chapter, the parameters of a Mel-scaled filter bank are used as the initial guess for $\boldsymbol{\alpha}$.

Note that the parameters of the initial GMMs are trained from the MFCC features using the conventional EM algorithm [43].

5.1.2 Compute the auxiliary function $\mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\})$

Because there is no closed-form solution for $\hat{\boldsymbol{\alpha}}$ in Eq. 5.1, the gradient ascent method is employed in Step 2.

Let $\mathbf{y}_t^{(r)}$ denote the features extracted using the filter bank $\boldsymbol{\alpha}$ at frame t. The

current $\boldsymbol{\alpha}$ is either initialized in Step 1, or obtained from Step 2 of the previous iteration. The probability of $\mathbf{y}_t^{(r)}$ belonging to mixture m of class r denoted by $\gamma_m^{(r)}(t)$ can be calculated as:

$$\gamma_m^{(r)}(t) = \frac{\omega_m^{(r)} \mathcal{N}(\mathbf{y}_t^{(r)}; \boldsymbol{\mu}_m^{(r)}, \boldsymbol{\Sigma}_m^{(r)})}{\sum_{m'=1}^{M} \omega_{m'}^{(r)} \mathcal{N}(\mathbf{y}_t^{(r)}; \boldsymbol{\mu}_{m'}^{(r)}, \boldsymbol{\Sigma}_{m'}^{(r)})} \tag{5.4}$$

where M denotes the number of Gaussians in each GMM, $\omega_m^{(r)}$, $\boldsymbol{\mu}_m^{(r)}$, and $\boldsymbol{\Sigma}_m^{(r)}$ are the weight, mean, and covariance matrix of the Gaussian mixture m of class r, obtained from the initialization or Step 3 of the previous iteration, $\mathcal{N}(\cdot)$ means Gaussian distribution.

Assuming that the discrete cosine transform (DCT) in feature extraction eliminates the dependencies among features from different dimensions, the covariance matrix of each Gaussian is a diagonal matrix. Suppose static (s), derivative (d), and acceleration (a) cepstral features are extracted, i.e. $\bar{\mathbf{y}}_t^T = [\bar{\mathbf{y}}_t^{s^T} \bar{\mathbf{y}}_t^{d^T} \bar{\mathbf{y}}_t^{a^T}]^T$.

In Step 2, the auxiliary function can be expressed as:

$$\begin{aligned} & \mathcal{Q}(\{\boldsymbol{\alpha}, \mathcal{M}\}, \{\bar{\boldsymbol{\alpha}}, \mathcal{M}\}) \\ = & \sum_{r=1}^{R} \sum_{m=1}^{M} \sum_{t=1}^{T^{(r)}} \gamma_m^{(r)}(t) \mathcal{N}(\bar{\mathbf{y}}_t^{(r)}; \boldsymbol{\mu}_m^{(r)}, \boldsymbol{\Sigma}_m^{(r)}) \\ = & -\frac{1}{2} \sum_{g \in \{s,d,a\}} \sum_{r=1}^{R} \sum_{m=1}^{M} \sum_{t=1}^{T^{(r)}} \left[\gamma_m^{(r)}(t) (\bar{\mathbf{y}}_t^{g^{(r)}} - \boldsymbol{\mu}_m^{g^{(r)}})^T \boldsymbol{\Sigma}_m^{g^{-1}(r)} (\bar{\mathbf{y}}_t^{g^{(r)}} - \boldsymbol{\mu}_m^{g^{(r)}}) \right] + C \end{aligned} \tag{5.5}$$

where R denotes the number of classes, $T^{(r)}$ denotes the number of frames in class r, $\bar{\mathbf{y}}_t^{(r)}/\bar{\mathbf{y}}_t^{s^{(r)}}/\bar{\mathbf{y}}_t^{d^{(r)}}/\bar{\mathbf{y}}_t^{a^{(r)}}$ denotes the whole/static/derivative/acceleration features at frame t extracted using the filter bank $\bar{\boldsymbol{\alpha}}$, C denotes a term that is invariant to $\bar{\boldsymbol{\alpha}}$.

5.1.3 Compute $\partial\mathcal{Q}(\{\boldsymbol{\alpha},\mathcal{M}\},\{\bar{\boldsymbol{\alpha}},\mathcal{M}\})/\partial\bar{\boldsymbol{\alpha}}$

By using the chain rule, we have

$$\begin{aligned}&\frac{\partial\mathcal{Q}(\{\boldsymbol{\alpha},\mathcal{M}\},\{\bar{\boldsymbol{\alpha}},\mathcal{M}\})}{\partial\bar{\boldsymbol{\alpha}}}\\ =\ &-\sum_{g\in\{s,d,a\}}\sum_{r=1}^{R}\sum_{m=1}^{M}\sum_{t=1}^{T^{(r)}}\left[\gamma_m^{(r)}(t)\frac{\partial\bar{\mathbf{y}}_t^{g^{(r)}}}{\partial\bar{\boldsymbol{\alpha}}}\boldsymbol{\Sigma}_m^{g^{-1(r)}}(\bar{\mathbf{y}}_t^{g^{(r)}}-\boldsymbol{\mu}_m^{g^{(r)}})\right]\end{aligned}\tag{5.6}$$

At frame t, let $\bar{e}_{t_l}$ denote the energy out of the l_{th} filter, and $\bar{\mathbf{e}}_t=[\bar{e}_{t_1}\cdots\bar{e}_{t_L}]$ denote the energy output of the filter bank. When $\bar{\mathbf{e}}_t$ is taken as input, the static cepstral coefficient $\bar{\mathbf{y}}_t^s$ is the output of the three cascaded feature extraction sub-procedures: logarithm, DCT and cepstral liftering:

$$\bar{\mathbf{y}}_t^s=\mathbf{M}_{\text{CEP\_LFT}}^T\mathbf{M}_{\text{DCT}}^T\log\bar{\mathbf{e}}_t\tag{5.7}$$

where $\mathbf{M}_{\text{CEP\_LFT}}$ is a $D\times D$ diagonal matrix:

$$[\mathbf{M}_{\text{CEP\_LFT}}]_d=1+\frac{d-1}{2}\sin\frac{\pi(d-1)}{N},\quad d=1\cdots D,\tag{5.8}$$

where d denotes the diagonal index, D denotes the dimension of the static features, N denotes the cepstral liftering coefficient; $\mathbf{M}_{\text{DCT}}$ is an $L\times D$ matrix:

$$[\mathbf{M}_{\text{DCT}}]_{l,d}\ =\ \sqrt{\frac{2}{L}}\cos\frac{\pi(l-0.5)(d-1)}{L},\quad l=1\cdots L,\quad d=1\cdots D\tag{5.9}$$

where l denotes the row index, d denotes the column index, L denotes the number of the filters.

Re-applying the chain rule, $\partial\bar{\mathbf{y}}_t^s/\partial\bar{\boldsymbol{\alpha}}$ can be expressed as:

$$\frac{\partial\bar{\mathbf{y}}_t^s}{\partial\bar{\boldsymbol{\alpha}}}=\frac{\partial\bar{\mathbf{e}}_t}{\partial\bar{\boldsymbol{\alpha}}}\frac{\partial\log\bar{\mathbf{e}}_t}{\partial\bar{\mathbf{e}}_t}\mathbf{M}_{\text{DCT}}\mathbf{M}_{\text{CEP\_LFT}}\tag{5.10}$$

$\partial\bar{\mathbf{e}}_t/\partial\bar{\boldsymbol{\alpha}}$ is solved as follows. Let $H_l[f]$ denote the frequency response of the

triangular filter l in the filter bank shown in Fig. 5.1, we have:

$$H_l[f] = \begin{cases} \dfrac{f - \bar{\alpha}_{l-1}}{\bar{\alpha}_l - \bar{\alpha}_{l-1}} & \bar{\alpha}_{l-1} \leq f < \bar{\alpha}_l \\ \dfrac{f - \bar{\alpha}_{l+1}}{\bar{\alpha}_l - \bar{\alpha}_{l+1}} & \bar{\alpha}_l \leq f < \bar{\alpha}_{l+1} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

where f denotes the frequency. Let $S_t[f]$ denote the power spectrum at frame t, the energy output of l_{th} filter can be expressed as: $\bar{e}_{t_l} = \sum_{f=0}^{F_s/2} H_l[f] S_t[f]$, where F_s is the sampling frequency. $\partial \log \bar{\mathbf{e}}_t / \partial \bar{\mathbf{e}}_t$ in Eq. 5.10 is an $L \times L$ diagonal matrix:

$$\Big[\frac{\partial \log \bar{\mathbf{e}}_t}{\partial \bar{\mathbf{e}}_t}\Big]_l = \frac{1}{\bar{e}_{t_l}}, \quad l = 1 \cdots L \tag{5.12}$$

where l denotes the diagonal index. $\partial \bar{\mathbf{e}}_t / \partial \bar{\boldsymbol{\alpha}}$ in Eq. 5.10 is an $(L+2) \times L$ band matrix:

$$\Big[\frac{\partial \bar{\mathbf{e}}_t}{\partial \bar{\boldsymbol{\alpha}}}\Big]_{p,l} = \begin{cases} \sum\limits_{f=\bar{\alpha}_{l-1}}^{\bar{\alpha}_l} \dfrac{f - \bar{\alpha}_l}{(\bar{\alpha}_{l-1} - \bar{\alpha}_l)^2} S_t[f] & p = l \\ \Big[\sum\limits_{f=\bar{\alpha}_l}^{\bar{\alpha}_{l+1}} \dfrac{f - \bar{\alpha}_{l+1}}{(\bar{\alpha}_l - \bar{\alpha}_{l+1})^2} - \sum\limits_{f=\bar{\alpha}_{l-1}}^{\bar{\alpha}_l} \dfrac{f - \bar{\alpha}_{l-1}}{(\bar{\alpha}_l - \bar{\alpha}_{l-1})^2} \Big] S_t[f] & p = l+1 \\ \sum\limits_{f=\bar{\alpha}_l}^{\bar{\alpha}_{l+1}} \dfrac{f - \bar{\alpha}_l}{(\bar{\alpha}_{l+1} - \bar{\alpha}_l)^2} S_t[f] & p = l+2 \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

$$p = 1 \cdots L+2, \quad l = 1 \cdots L$$

where p denotes the row index, and l denotes the column index.

Since the derivative features are calculated as:

$$\bar{\mathbf{y}}_t^d = \frac{\sum_{\theta=1}^{\Theta_d} \theta (\bar{\mathbf{y}}_{t+\theta}^s - \bar{\mathbf{y}}_{t-\theta}^s)}{2 \sum_{\theta=1}^{\Theta_d} \theta^2} \tag{5.14}$$

where Θ_d denotes the coefficient for computing the derivative features, after calculating $\partial \bar{\mathbf{y}}_t^s / \partial \bar{\boldsymbol{\alpha}}$, $\partial \bar{\mathbf{y}}_t^g / \partial \bar{\boldsymbol{\alpha}} \quad (g = d)$ in Eq. 5.6 can be computed as:

$$\frac{\partial \bar{\mathbf{y}}_t^d}{\partial \bar{\boldsymbol{\alpha}}} = \frac{\sum_{\theta=1}^{\Theta_d} \theta \big(\frac{\partial \bar{\mathbf{y}}_{t+\theta}^s}{\partial \bar{\boldsymbol{\alpha}}} - \frac{\partial \bar{\mathbf{y}}_{t-\theta}^s}{\partial \bar{\boldsymbol{\alpha}}} \big)}{2 \sum_{\theta=1}^{\Theta_d} \theta^2} \tag{5.15}$$

Since the acceleration features are obtained from the derivative features in the same way as obtaining the derivative features from the static features, $\partial\bar{\mathbf{y}}_t^g/\partial\bar{\boldsymbol{\alpha}}$ $(g = a)$ in Eq. 5.6 is calculated as:

$$\frac{\partial\bar{\mathbf{y}}_t^a}{\partial\bar{\boldsymbol{\alpha}}} = \frac{\sum_{\theta=1}^{\Theta_a}\theta\big(\frac{\partial\bar{\mathbf{y}}_{t+\theta}^d}{\partial\bar{\boldsymbol{\alpha}}} - \frac{\partial\bar{\mathbf{y}}_{t-\theta}^d}{\partial\bar{\boldsymbol{\alpha}}}\big)}{2\sum_{\theta=1}^{\Theta_a}\theta^2} \tag{5.16}$$

where Θ_a denotes the coefficient for computing the acceleration features.

5.1.4 Solve $\hat{\mathcal{M}}$

Let $\hat{\mathbf{y}}_t$ denote the features extracted from the optimal filter bank $\hat{\boldsymbol{\alpha}}$ obtained from Step 2 of the current iteration. In Step 3, the estimated model parameters, i.e. $\hat{\mathcal{M}}^{(r)}$, can be expressed as:

$$\hat{\omega}_m^{(r)} = \frac{\sum_{t=1}^{T^{(r)}}\gamma_m^{(r)}(t)}{T^{(r)}} \tag{5.17}$$

$$\hat{\boldsymbol{\mu}}_m^{(r)} = \frac{\sum_{t=1}^{T^{(r)}}\gamma_m^{(r)}(t)\hat{\mathbf{y}}_t^{(r)}}{\sum_{t=1}^{T}\gamma_m^{(r)}(t)} \tag{5.18}$$

$$\hat{\boldsymbol{\Sigma}}_m^{(r)} = \frac{\sum_{t=1}^{T^{(r)}}\gamma_m^{(r)}(t)(\hat{\mathbf{y}}_t^{(r)} - \hat{\boldsymbol{\mu}}_m^{(r)})(\hat{\mathbf{y}}_t^{(r)} - \hat{\boldsymbol{\mu}}_m^{(r)})^T}{\sum_{t=1}^{T}\gamma_m^{(r)}(t)} \tag{5.19}$$

where $\hat{\omega}_m^{(r)}$, $\hat{\boldsymbol{\mu}}_m^{(r)}$, and $\hat{\boldsymbol{\Sigma}}_m^{(r)}$ are the estimated weight, mean, and covariance matrix of the Gaussian mixture m of class r. These model parameters will be used in Step 2 of the next iteration as shown in Eq. 5.4. Note that in Step 1, the model parameters are estimated the same way as Step 3, except replacing $\hat{\mathbf{y}}_t$ with initial MFCC features.

5.2 Experiments

The Antbird call corpus contains 3366 bird calls from 5 species: Barred Antshrike (BAS), Dusky Antbird (DAB), Great Antshrike (GAS), Mexican Antthrush (MAT),

Dot-winged Antwren (DWA) [38]. The training set is 85 minutes long and the testing set is 42 minutes long. The calls are 0.5 - 5.0 seconds long. Examples of bird calls are shown in [38]. The frequency range of the bird calls is from 500 to 6000 Hz. The signal is downsampled from 44.1 kHz to 16 kHz. The low and high cut-off frequencies of the filter bank, $\alpha_{\min}$ and $\alpha_{\max}$, are set to 360 and 6500 Hz, respectively, to remove irrelevant frequency components for bird call classification [68].

Two feature extraction methods are compared: the standard MFCC extraction with a Mel-scaled filter bank, and the improved MFCC extraction with an optimized filter bank obtained from Algorithm 1. The number of filters in the filter bank, L, is set to 26. The cepstral liftering coefficient, N, is set to 22. The dimension of the static, derivative, and acceleration features, D, is set to 13. The coefficients for computing the derivative and acceleration features, Θ_d and Θ_a, are both set to 2. The frame step size is 10 ms, and the frame length is 25 ms. In the GMM classifier, the number of Gaussians in each species' model, M, is set to 256. In the filter bank optimization, the convergence threshold, ϵ, is set to 10^{-3}.

The baseline system using MFCC features has a classification error rate of 8.7%. By using the new features extracted using the optimal filter bank obtained from Algorithm 1, the error rate is reduced to 6.2%. The p-value of significance test is 0.024, which means that the proposed method is statistically significant for a significance level of 0.05. The optimization converges at the 6-th iteration, while the lowest classification error rate is achieved at the 4-th iteration. Model overfitting can be the explanation.

The confusion matrix of results obtained by using the Mel-scaled and optimized filter banks are shown in Table 5.1. The calls of BAS, MAT, and DWA are less likely to be misclassified as other species compared to those of DAB and

GAS. The optimized filter bank effectively reduced the DAB and GAS classification errors by 1.0% and 0.4%, respectively.

Let $\alpha_l^0/\hat{\alpha}_l$ and $B_l^0/\hat{B}_l$ denote the center frequency and bandwidth of l_{th} filter in the Mel-scaled/optimal filter bank, respectively. Note that in the triangular filter bank shown in Fig. 5.1, we have:

$$B_l^0 = \alpha_{l+1}^0 - \alpha_{l-1}^0 \tag{5.20}$$

$$\hat{B}_l = \hat{\alpha}_{l+1} - \hat{\alpha}_{l-1} \tag{5.21}$$

To show the percentages of the center frequencies and bandwidths of the optimal filter bank that are shifted, compared to the corresponding ones in the Mel-scaled filter bank, two difference measures regarding the l_{th} filter denoted by Δ_l^α and Δ_l^B, are defined as follows:

$$\Delta_l^\alpha = (\hat{\alpha}_l/\alpha_l^0 - 1) \times 100\% \tag{5.22}$$

$$\Delta_l^B = (\hat{B}_l/B_l^0 - 1) \times 100\% \tag{5.23}$$

In a Mel-scaled filter bank, the distances of the center frequency of l_{th} filter to its left and right counterparts are $\alpha_l^0 - \alpha_{l-1}^0$ and $\alpha_{l+1}^0 - \alpha_l^0$. The smaller the distances are, the higher the frequency resolution at frequencies near α_l^0 is [48]. Since $B_l^0 = [\alpha_{l+1}^0 - \alpha_l^0] + [\alpha_l^0 - \alpha_{l-1}^0]$, the bandwidth of the filter can be used as a measure of the frequency resolution at frequencies near the center frequency of the filter. The same conclusion can be drawn from the optimal filter bank.

A comparison of frequency parameters of the Mel-scaled and optimal filter banks are shown in Table 5.2. In the optimal filter bank, the bandwidth sequence $\{\hat{B}_0, \cdots, \hat{B}_L\}$ is no longer monotonically increasing compared to the Mel-scaled filter bank. As mentioned before, the shifting of the center frequencies and changing of the bandwidths compared to their counterparts in the Mel-scaled filter bank cause the frequency resolutions at different frequencies to change. In

Table 5.1: The confusion matrix of the species classification results on the test set. The numbers without parentheses are obtained by using the Mel-scaled filter bank. The numbers in parentheses denote the changes after using the optimized filter bank. For example, GAS was confused as MAT 32 times with Mel-scaled filter bank, but the confusion times were reduced by 11 after optimization.

		Classified (#)				
		BAS	DAB	GAS	MAT	DWA
Classes (#)	BAS	118(+1)	0(0)	1(-1)	0(0)	1(+1)
	DAB	2(0)	415(0)	13(-4)	13(-2)	1(+2)
	GAS	9(-5)	7(+2)	127(+3)	32(-11)	0(0))
	MAT	0(0)	0(-2)	3(-1)	301(+2)	0(0)
	DWA	1(+2)	9(-2)	3(-2)	2(0)	62(0)

the fbEM algorithm, the maximum likelihood criterion is used to raise or lower the frequency resolutions at certain frequencies such that more discriminative information for classification can be extracted from spectra. Therefore, a lower classification error rate can be achieved.

The bandwidths of the filters in both filter banks are small at low frequencies, which means more discriminative information for classification resides at low frequencies. The bandwidths of 1st, 2nd, 9th, 10th, and 15th filters in the optimal filter bank are small compared to other adjacent filters. The bandwidths of these filters are also significantly less ($> 25\%$) than their counterparts in the Mel-scaled filter bank. Thus, more discriminative information for classification may reside between 360 - 532, 1176 - 1458, and 2227 - 2552 Hz compared to other frequencies in the filter bank.

Table 5.2: Center frequencies (α_l^0 and $\hat{\alpha}_l$) and bandwidths (B_l^0 and $\hat{B}_l$) of the Mel-scaled and optimized filter bank, where $l = 1 \cdots L$. $L = 26$. Δ_l^α and Δ_l^B are the percentage change as defined in Eqs. 5.22 and 5.23. The upper and lower cut-off frequencies of the filter banks are: $\alpha_0^0 = \hat{\alpha}_0 = 360$ Hz, and $\alpha_{L+1}^0 = \hat{\alpha}_{L+1} = 6500$ Hz, respectively.

l	α_l^0 (Hz)	$\hat{\alpha}_l$ (Hz)	Δ_l^α (%)	B_l^0 (Hz)	$\hat{B}_l$ (Hz)	Δ_l^B (%)
1	438	415	-5.3	162	112	-30.4
2	522	472	-9.5	174	118	-31.8
3	611	532	-12.8	186	177	-4.8
4	708	650	-8.2	200	270	35.2
5	811	803	-1.0	215	250	16.3
6	923	899	-2.5	230	213	-7.6
7	1042	1016	-2.5	247	277	11.9
8	1170	1176	0.5	266	257	-3.2
9	1307	1273	-2.6	285	187	-34.5
10	1455	1363	-6.3	306	185	-39.6
11	1614	1458	-9.6	329	286	-12.9
12	1784	1649	-7.5	353	315	-10.8
13	1966	1772	-9.9	379	577	52.4
14	2162	2227	-3.0	407	595	46.4
15	2373	2368	-0.2	436	325	-25.5
16	2599	2552	-1.8	469	399	-14.9
17	2841	2767	-2.6	503	607	20.7
18	3102	3159	1.8	540	639	18.4
19	3381	3406	0.7	580	508	-12.4
20	3682	3667	-0.4	622	676	8.7
21	4004	4083	2.0	668	700	4.8
22	4350	4367	0.4	717	667	-7.0
23	4721	4750	0.6	770	829	7.6
24	5120	5196	1.5	827	788	-4.7
25	5547	5537	-0.2	887	886	-0.1
26	6007	6082	1.3	953	963	1.1

5.3 Conclusions

The fbEM algorithm offers an approach to jointly estimate filter bank parameters in feature extraction, and model parameters. Using the fbEM algorithm, the bird species classification accuracy on a large noisy corpus is increased by optimizing the center frequencies and bandwidths of the filter bank used in cepstral feature extraction. In the future, we will attempt to expand the work to speech recognition.

CHAPTER 6

Syllable Pattern-Based Bird Song Detection

The motivation of this study is to automatically detect Robin songs from continuous recordings.

An HMM-based detector with a general model trained from all the syllables is designed as a baseline system. In an improved system, syllable patterns are first inferred from similar syllables observed in the recordings; HMMs of the inferred syllable patterns are then trained to allow finer acoustic modelling of the syllables. According to our experimental results, the proposed syllable pattern-based detector is promising in terms of the hit rate and false alarm rate.

6.1 Robin Syllable and Song

The time waveform and spectrogram of a typical Robin song is shown in Fig. 6.1. It can be seen that the song is composed of several different syllables. Note that these units are sometimes referred to as phrases or song types [29]. Although these syllables have similar harmonic structures as voiced speech of humans, there are three main differences. The first is that the pitch of the Robin is higher than that of humans with fundamental frequencies ranging between 1500 and 4500 Hz. The second is that Robins can only intermittently vocalize syllables, but not continuously as can humans. The third is that Robins may produce two pitch frequencies simultaneously during vocalization as shown in the regions labeled as

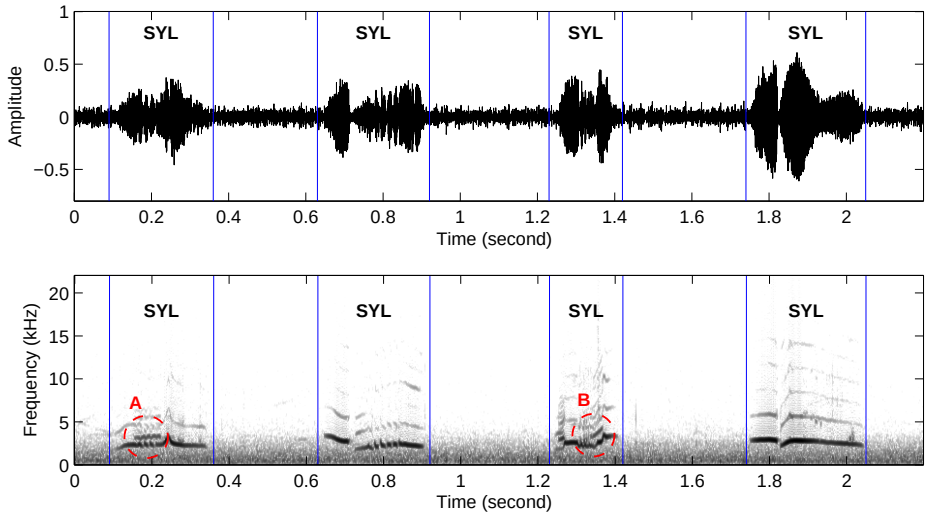

Figure 6.1: Time waveform and spectrogram of a typical Robin song. **SYL** refers to the syllable units.

A and B in Fig. 6.1. This phenomenon can be attributed to how birds produce songs [69]. During Robin vocalizations, air flows from two different syringes are controlled by the lateral labium and the medial tympaniform membranes. These membranes are located on the medial walls of the bronchus, and these morphological structures enable Robins to have two voicing sources. When the controllers of the two sources are vibrating at different speeds, two different fundamental frequencies are produced simultaneously.

6.2 RMBL-Robin Database

The RMBL-Robin database used in this study was collected by using a close-field song meter (www.wildlifeacoustics.com) at the Rocky Mountain Biological Laboratory near Crested Butte, Colorado. The sampling rate is 44.1 kHz. The recorded Robin songs are naturally corrupted by different kinds of background

Table 6.1: The details of the RMBL-Robin database

	Length (minutes)	Syllable #	Song #
Training Set	45.5	1644	457
Test Set	32.8	970	277

noises, such as wind, water and other vocal bird species. Non-target songs may overlap with target songs. Each song usually consists of 2-10 syllables. The dataset is 78.3 minutes long and divided into two sets for training and testing purposes. The details of the database are shown in Table 6.1. Note that all analyses are conducted on the training set.

6.3 Inference of Syllable Patterns

Objectively inferring syllable patterns is not only important in studying the singing behaviour of Robins, but also necessary to improve Robin song detection in the audio stream.

6.3.1 Distance Measure Between Syllables

A distance measure which was originally used for isolated word recognition is adopted. The distance between two syllables is defined as the minimum accumulative frame-level difference obtained in a dynamic time warping scheme [70]. The difference between two frames is based on the log likelihood ratio of the minimum prediction error [26].

The details of the distance measure is described in the following.

The likelihood ratio of the prediction error from frame y to frame x, $D(y||x)$,

is defined as

$$D(y||x) = \log \frac{E_{yx}}{E_{xx}} = \log \frac{\mathbf{a}_y^T \mathbf{R}_x \mathbf{a}_y}{\mathbf{a}_x^T \mathbf{R}_x \mathbf{a}_x} \tag{6.1}$$

where E_{yx} denotes the error obtained by feeding frame y into the inverse LPC filter inferred from frame x, E_{xx} is the minimum prediction error for the LPC system inferred from frame x; $\mathbf{a}_x$ and $\mathbf{R}_x$ denote the LPC coefficients and auto-correlation coefficient matrix of frame x, $\mathbf{R}_x$. Here, we use a symmetric difference measure, $D_f(x, y)$, defined as

$$D_f(x, y) = \frac{1}{2}\big[(D(x||y) + D(y||x)\big] \tag{6.2}$$

In this study, a fixed frame rate LPC analysis is first conducted on the training set to acquire the distribution of the difference $D_f(x, y)$ between two adjacent frames. There are some frames between which the distances are small. Downsampling of the LPC analysis over these frames is essential to removing redundant information. When the distances are large between other frames, an upsampling of the LPC analysis is also necessary to capture the rapidly changing pitch information. In essence, a frame dropping LPC analysis is then applied on each syllable.

We also use a symmetric distance measure between every two syllables $\mathbf{X}$ and $\mathbf{Y}$. It is denoted by $D_s(\mathbf{X}, \mathbf{Y})$, which is defined as

$$D_s(\mathbf{X}, \mathbf{Y}) = \frac{1}{2}\big[D_s(\mathbf{Y}||\mathbf{X}) + D_s(\mathbf{X}||\mathbf{Y}))\big] \tag{6.3}$$

where $D_s(\mathbf{Y}||\mathbf{X})$ denotes the distance from syllable $\mathbf{Y}$ to syllable $\mathbf{X}$. It is obtained through dynamic time warping (DTW) [71], i.e. minimizing the accumulative aligned frame-level differences defined in Eq. 6.2.

Although the defined distance $D_s(\mathbf{X}, \mathbf{Y})$ does not satisfy the triangular inequality, it was used as a distance measure for isolated word recognition [70],

and can be used as the distance measure for the Robin syllable clustering in the following section.

6.3.2 Hierarchical Clustering Analysis

The objective of clustering analysis in this section is to search common patterns which allow fine acoustic modelling of the Robin syllables compared to only using one single general pattern for all the syllables. Training different models or templates for different keywords has proved to be effective for keyword spotting [72] in which phoneme level transcription is available. However, for the training set of Robin songs, only boundary information of the syllables is annotated. Thus, it is necessary to infer the number of common syllable patterns from the training set, and then train acoustic models for those patterns.

Providing the distance measure between two syllables defined in the previous section, it is possible to conduct a distance measure-based hierarchical clustering analysis. In this study, a modified average-linkage hierarchical clustering is used to reliably cluster syllables into patterns. Before introducing the algorithm, the inter-cluster distance of cluster C, $D_c(C)$, is defined as

$$D_c(C) = \frac{1}{N_C(N_C-1)} \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} D_s(\mathbf{X}_i, \mathbf{X}_j) \tag{6.4}$$

where N_C denotes the number of syllables in the cluster, $\mathbf{X}_i$ denotes the i_{th} syllable in the cluster. The intra-cluster distance between cluster C_a and C_b denoted by $D_c(C_a, C_b)$ is defined as

$$D_c(C_a, C_b) = \frac{1}{N_{C_a} N_{C_b}} \sum_{i=1}^{N_{C_a}} \sum_{j=1}^{N_{C_b}} D_s(\mathbf{X}_i^{C_a}, \mathbf{X}_j^{C_b}) \tag{6.5}$$

where N_{C_a} denotes the number of syllables in the cluster C_a, and $\mathbf{X}_i^{C_a}$ denotes the i_{th} syllable in the cluster C_a.

The pseudocode of the modified average-linkage hierarchical clustering algorithm is expressed in the following:

Algorithm 6.3.1: A MODIFIED HIERARCHICAL CLUSTERING (C)

Set the stopping distance threshold as $D^C_{\max}$

Each syllable is initiated as a cluster.

do
- Search the closest two clusters, C_{i^*} and C_{j^*}, by comparing $D_c(C_i, C_j)$
- Copy the elements of C_{i^*} and C_{j^*} into a new cluster C^*
- **if** $D_c(C^*) > D^C_{\max}$
  - **then** Remove C^*, **break** ;
- **else**
  - **then** Use C^* to replace C_{i^*} and C_{j^*}

while More than one cluster is left

$D_c(C)$: intra cluster distance of cluster C;

$D_c(C_a, C_b)$: inter cluster distance of cluster C_a and C_b;

---

s In this chapter, only clusters with a number of syllables greater than a threshold, denoted by N^C_{th}, are retained as syllable patterns. The relationship between the number of syllable patterns and the stopping distance threshold $D^C_{\max}$ given different N^C_{th} is shown in Figure 6.2. Under the same clustering stopping threshold, the larger the cluster number threshold, the fewer syllable patterns there are. Under each cluster number threshold, the number of syllable patterns first increases then decreases when the clustering stopping threshold $D^C_{\max}$ increases. The increase/decrease patterns might occur because when $D^C_{\max}$ is small, many small clusters are not regarded as syllable patterns; when $D^C_{\max}$ has a high value, i.e. the allowable maximum intra-cluster distance is high, many syllables are

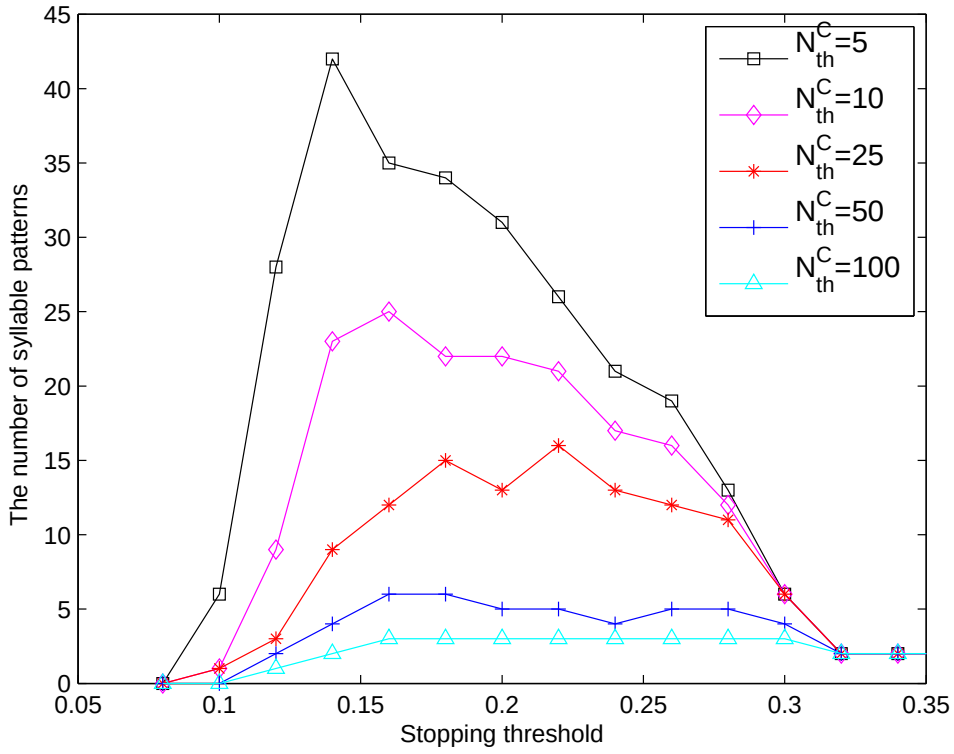

Figure 6.2: The relationship between the number of syllable patterns and stopping distance threshold $D^C_{\max}$ given different cluster number threshold N^C_{th}. Only clusters with numbers of syllables greater than N^C_{th} are regarded as syllable patterns.

clustered together, which causes the number of patterns to be small.

It is still difficult to infer the actual number of syllable patterns from the clustering results, because biologists are not clear about the repertoire size of the syllable patterns in Robin songs. However, clustering results are helpful in the sense of training acoustic models from the syllable patterns that are close in a certain feature space, which may improve detection and classification results.

6.4 Robin Song Detection System

During training, feature segments required by the template-based approach, i.e. DTW, can be obtained by examining the boundary information contained in the transcriptions. However, the boundary information is no longer available in the test set which implies that the template-based method is not suitable for the detection task, and pre-processing is needed to acquire the boundary information. As an HMM-based system with models of the Robin syllables and background sounds is capable of detecting the boundaries and classifying the sounds, by decoding the continuous feature stream, simultaneously, HMMs are used for acoustic modelling in our detection task. A left-to-right HMM with 3 emitting states is adopted for modelling the syllable patterns; an ergodic HMM with 3 emitting states is used for modelling the background sounds.

Two HMM networks A and B are constructed for acoustic model training and audio feature stream decoding purposes. Network A, shown in Figure 6.3, models all syllables as a single general HMM, and all background sounds as another general HMM. The difference between networks A and B, shown in Figure 6.4, is that different syllable patterns are modeled as different HMMs. As mentioned above, not all syllables can be clustered into a syllable pattern. An extra HMM with the same topology as the syllable pattern HMM is used for modelling unclustered syllables. Syllable patterns are inferred by using the clustering-based method mentioned in the previous section.

Bigram models for both HMM networks are learned from the training set such that each arc in the network is assigned a transition probability. The integration of the bigram model into the HMM networks implies the occurrence relationship between syllable and background sounds are taken into consideration.

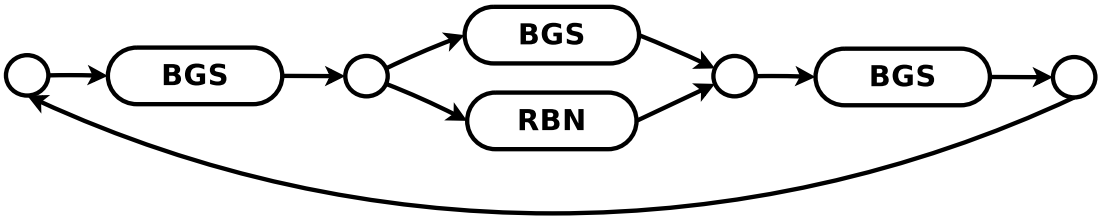

Figure 6.3: HMM network A. **RBN**: the general HMM for all Robin syllables. **BGS**: background sound HMM.

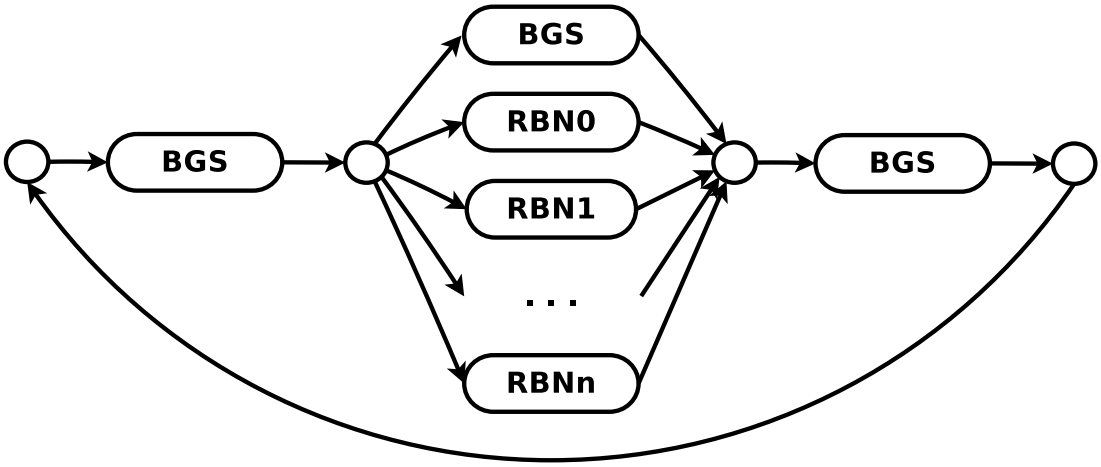

Figure 6.4: HMM network B. **RBNn**: the HMM for the n_{th} Robin syllable pattern. **RBN0**: the HMM for the remaining Robin syllables that do not belong to any syllable pattern. **BGS**: background sound HMM.

Unsupervised Maximum Likelihood Linear Regression (MLLR) adaptation [62] is applied to minimize the mismatch between the trained acoustic models and the test cases.

As we are interested in detecting the existence of the Robin songs, the syllable level decoding results need to be converted to song level results. According to our observation, the duration between syllables in a Robin song is less than 0.5 seconds most of the time. Therefore, detected syllables that are less than 0.5 seconds in distance are grouped into a single song.

6.5 Experimental Results

The performance of the Robin song detection is evaluated in terms of the recall rate and precision rate denoted by R and P which can be expressed as

$$R = \frac{N_h}{N_g} \times 100\%, \quad P = \frac{N_h}{N_d} \times 100\%, \tag{6.6}$$

where N_h is the number of hit songs, N_g is the number of the ground truth songs, and N_d is the number of detected songs. A detected song is regarded as a hit song only if the center of the detected song in time falls into the vicinity (± 0.5 seconds) of the center of a ground truth song.

The objective is to increase the recall rate and precision rate at the same time. Because of the well-known trade-off relationship between the two rates, the F-score, a weighted combination of the two rates denoted by F [73], is defined as:

$$F = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{6.7}$$

where β is a weighting factor. Since the recall rate is more important than the precision rate in this study, β is set to be 1.5.

The sampling rate of the recordings is 44.1 kHz. When the microphone is far from the vocalizers during the recording, the high frequency components (> 5000 Hz) of the songs are sometimes strongly attenuated. As the pitch information of the Robin ranging from 1500 to 4500 Hz are retained most of the time, a band pass filter with cut-off frequencies of 1000 and 5000 Hz is applied to the raw recordings.

For Robin syllables, the magnitude of the first harmonic is usually higher than other harmonics, and hence is less susceptible to background noise. As a pair of conjugate poles of the LPC filter is supposed to match one spectral peak, given the fact that there may exist one or two pitch harmonics in the pass-band, i.e. one or two spectral peaks in the spectrum, the order of LPC has to be 4 to capture the possible pitch frequencies.

In the fixed frame rate LPC analysis, a frame shift of 5 ms is used. In a frame dropping (FD)-based LPC analysis, effective frame shifts of 5, 10, and 20 ms are used. The parameters of the FD scheme are set to make the ratio of the numbers of frames with high, middle, and low frame rates to be 1:1:1. In both analyses, the frame length is 10 ms. A Hamming window is used in the framing processing.

In feature extraction, to be consistent with LPC-based clustering analysis, a 15-dimension feature composed of the 4th-order LPCs plus logarithm energy and first and second derivatives is computed every frame for model training and testing. The frame step size is fixed to 5 ms. The frame length is 10 ms.

In the frame dropping-based clustering analysis, the clustering stopping threshold $D^C_{\max}$ ranges from 0.08 to 0.40, the threshold of the number of syllables in a cluster N^C_{th} is set to be 5, 10, 25, 50, or 100. In acoustic modelling, the number of Gaussian mixtures per state is set to be 1, 2, 4, 8, 16, or 32. For the HMM network B, the highest F-score is achieved when $D^C_{\max} = 0.12$, $N^C_{\text{th}} = 25$, and

Table 6.2: the detection results including the Recall Rate (R), Precision Rate (P), and F-score (F) using HMM networks A and B. **wo VFR**: uses a fixed frame rate in syllable pattern clustering. **+ adapt**: unsupervised MLLR adaptation.

	R (%)	P (%)	F
Network A	74.2	71.8	0.734
Network B wo VFR	75.5	73.3	0.748
Network B	76.0	73.6	0.753
Network B + adapt	76.0	75.2	0.758

the number of Gaussian mixtures per state is 8. Changing the number of the states in the HMMs to other than 3 can not improve the F-score. Under this configuration, there are 3 HMMs for syllable patterns and 1 HMM for the background sound. The details of the detection results using HMM networks A and B are shown in Table 6.2. When replacing the simple HMM network A with the advanced network B, the recall and precision rate are both improved by 1.8% . When the network B is followed by an unsupervised MLLR adaptation module, the precision rate has a gain of 1.6% while the recall rate is unchanged. We also found that using a fixed frame rate in the syllable pattern clustering can result in a lower recall and precision rate.

6.6 Conclusions

Syllable patterns of Robin songs can be objectively inferred by performing a hierarchical clustering analysis in which the distance measure is calculated by aligning the LPC-based frame level differences. This HMM-based Robin song detection system with models trained for the syllable patterns has a higher hit rate under the same false alarm rate compared with a system with models trained

from all syllables.

CHAPTER 7

Summary and Future Work

7.1 Summary and Discussion

In this dissertation, we investigate noise-robust F0 tracking methods to reduce F0 estimation and voicing decision errors, and analyze bird song properties to improve bird song classification and detection accuracy.

In Chapter 2, we show that Prominent Signal-to-Noise Ratio (SNR) peaks constitute a simple and effective information source for F0 inference under both clean and noisy conditions. We model the effect of additive noise on clean speech spectra, given F0, in a statistical framework. We find that middle and high frequency bands (1-3 kHz) provide supplemental useful information for F0 inference in addition to low frequencies. We show that the proposed SAFE algorithm is more effective in reducing the Gross Pitch Errors (GPE) compared to other F0 estimators especially at low SNRs, and is robust in maintaining low Mean and Standard Deviation of the Fine Pitch Errors.

In Chapter 3, we show that the model-based U/V classifier can output robust U/V masks for F0 trackers under both white and babble noise conditions which is helpful for reducing the overall F0 Frame Errors (FFE), which is combination of GPE and Voicing Decision Errors (VDE). We also show that minimizing the FFE is more effective than minimizing the VDE alone. We have shown that the SAFE algorithm using masks generated from the GMM-based unvoiced/voiced

classifier has lower FFEs compared with prevailing F0 tracking algorithms under both clean and noisy conditions.

In Chapter 4, we propose a Correlation-Maximization denoising filter which is effective in enhancing target bird calls with a quasi-periodic structure in the time domain and suppressing non-target bird calls and other non-stationary noises, which results in a reduction in classification error rate. We show that compared to the Wiener filter, the Correlation-Maximization filter avoids estimating the SNR by using the periodicity of the target bird call. The advantage of the Correlation-Maximization filter over the Wiener filter is the ability of handling non-stationary noise.

In Chapter 5, we propose the fbEM algorithm which is an approach to jointly estimate filter bank parameters in feature extraction, and model parameters. We use the fbEM algorithm to increase the bird species classification accuracy on a large noisy corpus by optimizing the center frequencies and bandwidths of the filter bank used in cepstral feature extraction.

In Chapter 6, we show that syllable patterns of Robin songs can be objectively inferred by performing a hierarchical clustering analysis in which the distance measure is calculated by aligning the LPC-based frame level differences. The HMM-based Robin song detection system with models trained with information about 'syllable patterns' achieves a higher hit rate under the same false alarm rate compared with a system with models trained from all syllables.

7.2 Future work

Since the SAFE algorithm is a data-driven method, it is worthwhile to train it on a larger F0 database to obtain more robust models for F0 estimation and

unvoiced/voiced classification. However, recording a database with simultaneous speech and laryngograph signals is less easy than recording a database with only audio speech signals. For example, both KEELE and CSTR corpora used in this work are less than 6 minutes. Therefore, obtaining large databases for training F0 estimation models is an issue to be solved.

Currently, the SAFE algorithm assumes that the noise type and level are known in F0 estimation. However, it is necessary for a real world application to estimate the noise type and level. Therefore, a noise type and level identifier is needed.

For bird call classification, we currently denoise acoustic signals by running the proposed Correlation-maximization filter before feature extraction. We also used the proposed fbEM algorithm to improve the discriminability of the extracted features. In the future, long-term features should be explored and combined with MFCC features. Acoustic modeling techniques other than GMM and HMM can also be explored.

For bird song detection, we currently improve the detection accuracy by training finer acoustic models based on the inferred syllable patterns. In the future, it should be worthwhile to explore better acoustic modeling techniques.

References

[1] Gunnar Fant, *Acoustic Theory of Speech Production*, Mouton, 1960.

[2] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.

[3] W. Hess, *Pitch Determination of Speech Signals*, Springer-Verlag, Berlin, 1983.

[4] J.D. Markel, "The SIFT algorithm for fundamental frequency estimation," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 5, pp. 367–377, 1972.

[5] A.M. Noll, "Cepstrum pitch determination," *The Journal of the Acoustical Society of America*, vol. 41, no. 2, pp. 293–309, 1967.

[6] D. Talkin, "Robust algorithm for pitch tracking," *Speech Coding and Synthesis*, pp. 497–518, 1995.

[7] K. Kasi and S. Zahorian, "Yet another algorithm for pitch tracking," in *ICASSP*, 2002, vol. 1, pp. 361–364.

[8] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[9] H. Kawahara, H. Katayose, A de Chevigne, and R. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity," in *EUROSPEECH*, 1999, vol. 6, pp. 2781–2784.

[10] A. de Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *JASA*, vol. 111, no. 4, pp. 1917–1930, 2002.

[11] B. Yegnanarayana and K.S.R. Murty, "Event-based instantaneous fundamental frequency estimation from speech signals," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, no. 4, pp. 614–624, 2009.

[12] J. Le Roux, H. Kameoka, N. Ono; A. de Cheveigne, and S. Sagayama, "Single and multiple F0 contour estimation through parametric spectrogram modeling of speech in noisy environments," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1135–1145, 2007.

[13] B. Gold and L.R. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time domain," *The Journal of the Acoustical Society of America*, vol. 46, no. 2B, pp. 442–448, 1969.

[14] M. Lahat, R. Niederjohn, and D. Krusback, "A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, no. 6, pp. 741–750, 1987.

[15] F. Sha, J. Burgoyne, and L. Saul, "Multiband statistical learning for F0 estimation in speech," in *ICASSP*, 2004, vol. 5, pp. 661–664.

[16] J.C.R. Licklider, "A duplex theory of pitch perception," *Experientia*, vol. 23, no. 4, pp. 128–134, 1951.

[17] P. Hedelin and D. Huber, "Pitch period determination of aperiodic speech signals," in *ICASSP*, 1990, vol. 1, pp. 361–364.

[18] A. de Cheveigne, "Speech F0 extraction based on Licklider's pitch perception model," in *ICPhS*, 1991, pp. 218–221.

[19] M. Wu, D. Wang, and G.J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.

[20] D. Krusback and R. Niederjohn, "An autocorrelation pitch detector and voicing decision with confidence measures developed for noise-corrupted speech," *IEEE Trans. on Signal Processing*, vol. 39, no. 2, pp. 319–329, 1991.

[21] T. Shimamura and H. Kobayashi, "Weighted autocorrelation for pitch extraction of noisy speech," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 7, pp. 727–730, 2001.

[22] T. Abe, T. Kobayashi, and S. Imai, "Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency," in *ICSLP*, 1996, pp. 1277–1280.

[23] D.-J. Liu and C.-T. Lin, "Fundamental frequency estimation based on the joint time-frequency analysis of harmonic spectral structure," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 6, pp. 609–621, 2001.

[24] T. Nakatani and T. Irino, "Robust and accurate fundamental frequency estimation based on dominant harmonic components," *JASA*, vol. 116, no. 6, pp. 3690–3700, 2004.

[25] O. Deshmukh, C.Y. Espy-Wilson, A. Salomon, and J. Singh, "Use of temporal information: Detection of periodicity, aperiodicity, and pitch in speech," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 776–786, 2005.

[26] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[27] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo, "A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments," *Speech Communication*, vol. 50, no. 3, pp. 203–214, 2008.

[28] P. Marler, "A comparative approach to vocal learning: song development in white-crowned sparrows," *J Comp Physiol Psychol*, vol. 71, pp. 1–25, 1970.

[29] C. K. Catchpole and P. J. B. Slater, *Bird Song: Biological Themes and Variations*, Cambridge University Press, New York, 1995.

[30] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, Wiley Press, New York, 1949.

[31] C. Rogers, "High resolution analysis of bird sounds," in *Proc. of ICASSP*, 1995, vol. 5, pp. 3011–3014.

[32] P. Somervuo, A. Harma, and S. Fagerlund, "Parametric representations of bird sounds for automatic species recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 14, no. 6, pp. 2252–2263, 2006.

[33] J. Mallett and I. Pepperberg, "Identifying bird species from bird song using frequency component analysis," *JASA*, vol. 111, no. 5, pp. 2391–2392, 2002.

[34] A.L. McIlraith and H.C. Card, "Bird song recognition using back propagation and multivariate statistics," *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2740–2748, 1997.

[35] A.L. McIlraith and H.C. Card, "Bird song identification using artificial neural networks and statistical analysis," in *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering*, 1997, vol. 1, pp. 25–28.

[36] L. Ranjard and H.A. Ross, "Unsupervised bird song syllable classification using evolving neural networks," *JASA*, vol. 123, no. 6, pp. 4358–4368, 2008.

[37] J.A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study," *JASA*, vol. 103, no. 4, pp. 2185–2196, 1998.

[38] V. Trifa, A. Kirschel, and C. E. Taylor, "Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models," *The Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 2424–2431, 2008.

[39] A. Harma, "Automatic identification of bird species based on sinusoidal modeling of syllables," in *Proc. of ICASSP*, 2003, vol. 5, pp. 545–548.

[40] P. Somervuo and A. Harma, "Bird song recognition based on syllable pair histograms," in *Proc. of ICASSP*, 2004, vol. 5, pp. 825–828.

[41] S.E. Anderson, "Speech recognition meets bird song: A comparison of statistics-based and template-based techniques," *The Journal of the Acoustical Society of America*, vol. 106, no. 4, pp. 2130–2130, 1999.

[42] C. Kwan, G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, and K.C. Ho, "Bird classification algorithms: theory and experimental results," in *Proc. of ICASSP*, 2004, vol. 5, pp. 289–292.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[44] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annuals of Eugenis*, vol. 7, pp. 179–188, 1936.

[45] N. Kumar, "Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition," in *PhD thesis*, 1997.

[46] M.J.F. Gales, "Maximum likelihood multiple subspace projections for hidden Markov models," *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 37–47, 2002.

[47] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *ICASSP*, 2005, vol. 1, pp. 961–964.

[48] S.S. Stevens, J. Volkman, and E.B. Newman, "A scale for the measurement of the psychological magnitude of pitch," *JASA*, vol. 8, no. 3, pp. 185–190, 1937.

[49] M. Graciarena, M. Delplanche, E. Shriberg, A. Stolcke, and L. Ferrer, "Acoustic front-end optimization for bird species recognition," in *ICASSP*, 2010, pp. 293–296.

[50] G.S. Ying, L.H. Jamieson, and C.D. Michell, "A probabilistic approach to AMDF pitch detection," in *ICSLP*, 1996, vol. 2, pp. 1201–1204.

[51] Y.R. Wang, I.J. Wong, and T.C. Tsao, "A statistical pitch detection algorithm," in *ICASSP*, 2002, vol. 1, pp. 357–360.

[52] J.G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," pp. 347–354, 1997.

[53] F. Plante, G. Meyer, and W. A. Ainsworth, "A pitch extraction reference database," in *EUROSPEECH*, 1995, pp. 837–840.

[54] W. Chu and A. Alwan, "Reducing F0 frame error of F0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *ICASSP*, 2009, pp. 3969–3972.

[55] P.C. Bagshaw, S.M. Hiller, and M.A. Jack, "Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching," in *EUROSPEECH*, 1993, vol. 2, pp. 1003–1006.

[56] H. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000*, 2000, pp. 181–188.

[57] A.P. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, "The NOISEX-92 study on the effect of additive noise on automatic speech recognition," in *Technical report, DRA Speech Research Unit*, 1992.

[58] J. Sohn, N. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[59] Y. D. Cho and A. Kondoz, "Analysis and improvement of a statistical model-based voice activity detector," *IEEE Signal Processing Letters*, vol. 8, no. 10, pp. 276–278, 2001.

[60] J.-H. Chang, N. S. Kim, and S. K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Trans. on Signal Processing*, vol. 54, no. 6, pp. 1965–1976, 2006.

[61] ETSI ES 202 050 recommendation, "Speech processing, transmission and quality aspects (stq); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," 2007.

[62] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[63] E. Vilches, I.A. Escobar, E.E. Vallejo, and C.E. Taylor, "Data mining applied to acoustic bird species recognition," in *Proc. of ICPR*, 2006, vol. 3, pp. 400–403.

[64] G. Turin, "An introduction to matched filters," *IEEE Trans. on Information Theory*, vol. 6, no. 3, pp. 311–329, 1960.

[65] C. H. Greenewalt, *Bird Song: Acoustics and Physiology*, Smithsonian Institution Press, 1968.

[66] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[67] A. Satt and D. Malah, "Design of uniform DFT filter banks optimized for subband coding of speech," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1672–1679, 1989.

[68] W. Chu and A. Alwan, "A correlation-maximization denoising filter used as an enhancement frontend for noise robust bird call classification," in *Proc. of Interspeech*, 2009, pp. 2831–2834.

[69] R.A. Suthers, F. Goller, and C. Pytte, "The neuromuscular control of birdsong," *Philos Trans R Soc Lond B Biol Sci*, vol. 354, no. 1385, pp. 927–939, 1999.

[70] L. Rabiner, A. Rosenberg, and S. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 575–582, 1978.

[71] C. Myers, L. Rabiner, and A. Rosenberg, "An investigation of the use of dynamic time warping for word spotting and connected speech recognition," in *Proc. of ICASSP*, 1980, vol. 5, pp. 173–177.

[72] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. of ICASSP*, 1989, pp. 627–630.

[73] C. J. van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, 1979.