Noise Robust Pitch Tracking by Subband Autocorrelation Classification

Byung Suk Lee

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2012

Abstract

Noise Robust Pitch Tracking by Subband Autocorrelation Classification

Byung Suk Lee

Speech pitch tracking is one of the elementary tasks of the Computational Auditory Scene Analysis (CASA). While a human can easily listen to the voiced pitch in highly noisy recordings, the performance of automatic speech pitch tracking degrades in unknown noisy audio conditions. Traditional pitch trackers use either autocorrelation or the Fourier transform to calculate periodicity, which works well for clean recordings. For noisy recordings, however, the accuracy of these pitch trackers degrades in general. For example, the information in parts of the frequency spectrum may be lost due to analog radio band transmission and/or contain additive noise of various kinds.

Instead of explicitly using the most obvious features of autocorrelation, we propose a trained classifier-based approach, which we call Subband Autocorrelation Classification (SAcC). A multi-layer perceptron (MLP) classifier is trained on the principal components of the autocorrelations of subbands from an auditory filterbank. The output of the MLP classifier is temporally smoothed to produce the pitch track by finding the Viterbi path of a Hidden Markov Model (HMM). Training on various types of noisy speech recordings leads to a great increase in performance over state-of-the-art algorithms, according to both the traditional Gross Pitch Error (GPE) measure, and a proposed novel Pitch Tracking Error (PTE) which more fully reflects the accuracy of both pitch estimation/extraction and voicing detection in a single measure.

To verify the generalization and specificity of SAcC, we test SAcC on a real world

problem that has a large-scale noisy speech corpus. The data is from the DARPA Robust Automatic Transcription of Speech (RATS) program. The experiments on the performance evaluation of SAcC pitch tracking confirm the generalization power of SAcC across various unknown noise conditions and distinct speech corpora. We also report the use of SAcC output adds a significant improvement to a Speaker Identification (SID) system for RATS as well, suggesting the potential contribution of SAcC pitch tracking in the higher-level tasks.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SAcC	Subband Autocorrelation Classification.
SubSel	Subband Selection.
GPE	Gross Pitch Error.
HMM	Hidden Markov Model.
MLP	Multi-Layer Perceptron.
PCA	Principal Components Analysis.
PTE	Pitch Tracking Error.
SAD	Speech Activity Detection.
SNR	Signal-to-Noise Ratio.
UE	Unvoiced Error.
VAD	Voice Activity Detection.
VDE	Voicing Decision Error.
VE	Voiced Error.

Acknowledgments

As I finish writing this thesis, I feel so humbled. I was able to finish this thesis only with the help from the following people.

First, I would like to thank my advisor Dan Ellis. Without his exceptional trust, this unpredictable journey of the dissertation work and writing this thesis could not have been even undertaken. He showed an unwavering support in times when the progress was sluggish. I am very much fortunate to be able to work with him.

I also thank my thesis committee, Michael Collins, Julia Hirschberg, Xiaodong Wang, and John Wright, for their insightful discussions that refined my thesis.

I would like to express my gratitude to Chris Wiggins, who rescued me from the first plight during my graduate study and taught me how to be a graduate student.

I thank all the members of LabROSA for sharing thoughts and friendship.

Finally, I am so much grateful to my parents, Hyun Sang Lee and Jee Ho Kim, for their unconditional support: you waited so long for me to finish. I would like to thank my wife, Na Young Lee, for being with me through the difficult times. I am thankful to all the people who prayed for me when I encountered various obstacles.

To my family

Chapter 1

Introduction

As the number of portable devices with recording capabilities and large storage capacities increases, people have more and more unanalyzed and unlabeled audio data in uncontrolled acoustic conditions, mostly noisier than studio recordings. Often, the speech portion of the audio is of interest. Pitch is a recognizable trait that can be used to track or locate voiced speech in audio. In this thesis, we develop methods that identify voiced speech and estimate the corresponding fundamental frequency for various types of noisy recordings, i.e. noise robust pitch tracking methods.

Pitch is an important characteristic of speech, and determining pitch is important in analyzing speech signals. The definition of pitch can vary depending on the context. In this work, pitch refers to the fundamental frequency f0 of the voiced speech.

Figure 1.1 shows an example audio signal of a clean speech at a sampling rate of 16 kHz: (a) waveform, (b) autocorrelation, and (c) spectrogram (pitch in green dots). The waveform shows repeated patterns over time; the period of these repeated pattern corresponds to the fundamental period. The period of the pattern is 7.7 ms (130 Hz) in the beginning of speech and is 6.7 ms (150 Hz) in the end. As we will discuss in more detail in the next chapter, autocorrelation and spectrogram corresponds to two major ways to find the pitch, the time-domain and the frequency-domain methods. Both autocorrelation and the Fourier transform are calculated every 10 ms. The

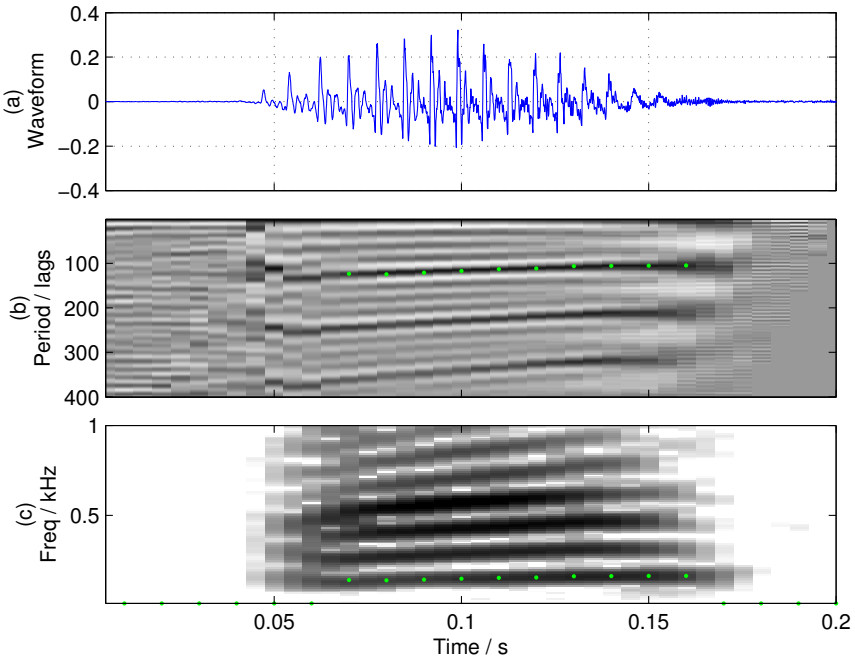


Figure 1.1: An example audio signal of a clean speech at a sampling rate of 16 kHz: (a) waveform, (b) autocorrelation, and (c) spectrogram (pitch in green dots).

ground truth pitches are also shown every 10 ms. In the spectrogram, green dots at 0 Hz indicate the non-pitch frames. On the other hand, in autocorrelation, the largest lag, 400, corresponds to 40 Hz, not 0 Hz; hence, there is no green dots for unvoiced frames in Figure 1.1 (b). The 10 ground truth pitch values, highlighted with green dots, start around 130 Hz and end around at 150 Hz, both in the autocorrelation and in the spectrogram.

The autocorrelation shows clear peaks (the darker color is, the higher the autocorrelation value is.) at pitches. But, there are peaks at multiple lags of the pitch, which corresponds to the submultiples of f0. We can also observe weaker peaks at sub-multiples of pitch period, which corresponds to the harmonics of f0. When these non-f0 peaks are stronger than f0, the "octave error" occurs. The "octave error" refers to the incorrect pitch recognition at multiples or submultiples of the true fundamental frequency/period. The spectrogram also shows repeated peaks at the multiples of f0,

the harmonics. It is not intuitively clear why the harmonics exist. When the repeated pattern is not a pure (sine or cosine) tone, the signal is represented by combination of f0 and the multiples of f0 (harmonics) by the Fourier analysis. This effect is shown as the harmonics in the spectrogram. It is another source of "octave error" at the multiples of f0.

There are several computational tasks related to pitch. The tasks that will be studied in this work are Voice Activity Detection (VAD), pitch estimation, and pitch tracking. In this thesis, we use the following definitions of these pitch-related tasks.

- Voice Activity Detection (VAD) refers to determining whether the audio contains voiced speech or not.
- Pitch estimation refers to estimating the f0 of voiced speech in audio. A pitch estimation algorithm will typically report some estimate of pitch even when no voicing is present. In addition to the pitch estimation, the algorithm may report the pitch strength, indicating how certain the pitch estimation is. Usually, the pitch strength output is not the best single value to perform VAD, hence the VAD performance based on this pitch strength is worse than specifically designed VAD algorithms.
- Pitch tracking refers to performing both pitch estimation and VAD at the same time. The point of distinguishing pitch tracking and pitch estimation is that pitch tracking focuses estimating f0 only on the voiced speech, in contrast, pitch estimation focuses more on estimating the pitch for the entire input audio. This thesis focuses on this pitch tracking aspect.

1.1 Motivation

Among many applications of pitch tracking, our direct application was to obtain robust speech pitch tracks automatically within large-scale unlabeled recordings such

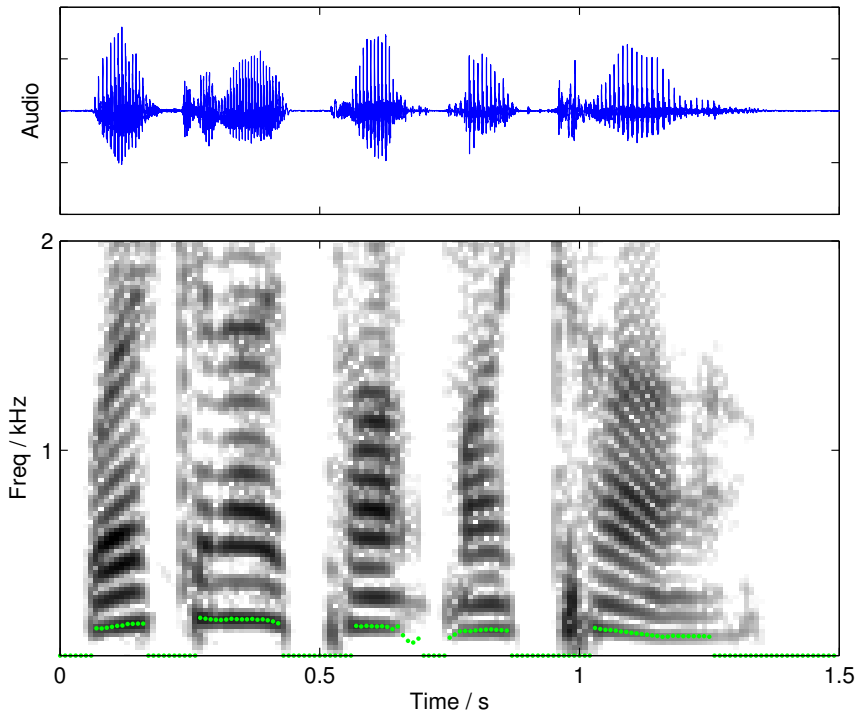


Figure 1.2: The audio signal and the spectrogram (pitch in green dots) of a clean speech example.

as user-generated audio content and communications intercepts. In such recordings, the audio is often corrupted by additive noise and/or communication channel degradation.

Traditional pitch trackers use either autocorrelation or the Fourier transform to calculate periodicity, which works well for clean recordings. For noisy recordings, however, the accuracy of pitch trackers degrades in general. For example, some of the frequency regions could be lost due to radio band transmission[1] and/or contain additive noise of various kinds. While autocorrelation is a useful technique for detecting periodicity, autocorrelation peaks suffer ambiguity, leading to the classic "octave error" in pitch tracking. Moreover, additive noise can affect autocorrelation in ways that are difficult to model.

[1]Radio band transmission in this thesis refers to analog modulation, not digital wireless.

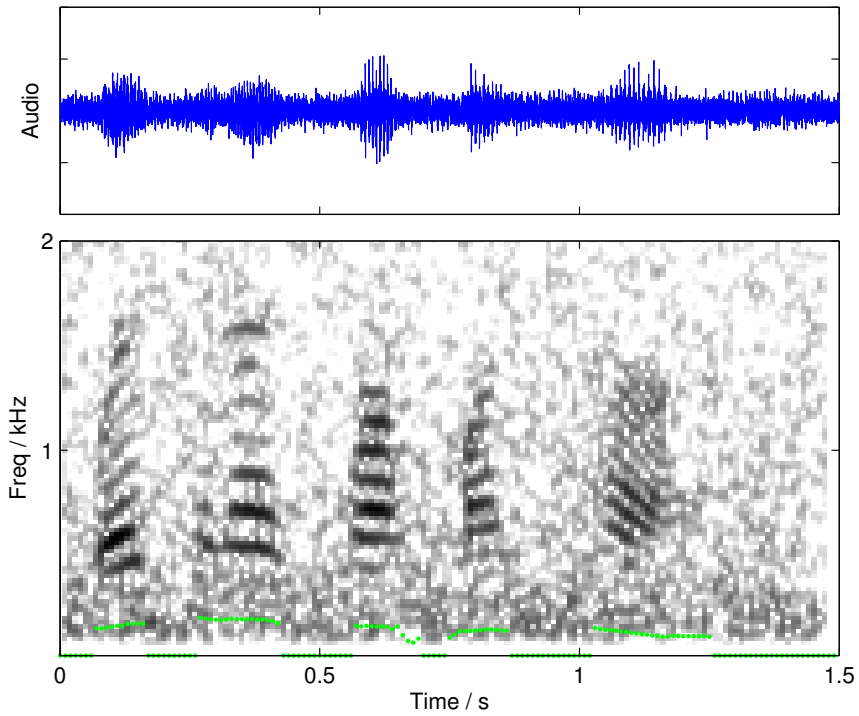


Figure 1.3: The audio signal and the spectrogram (pitch in green dots) of a noisy speech example.

Figure 1.2 shows the waveform, spectrogram, and ground-truth pitch for a clean speech example. Figure 1.3 shows the same example corrupted by a simulated radio channel. The audio in Figure 1.3 is both bandlimited (processed to simulate a narrowband radio channel) and mixed with additive pink noise. Comparing Figure 1.2 and 1.3, the spectrograms show the fundamental frequency f0 and a few harmonics are lost, making the pitch tracking a challenge.

The first idea for a noise robust pitch tracker pursued in this work was to build pitch tracking systems based on trained classifiers, rather than using heuristics to identify specific peaks in the spectrum or autocorrelation. Our approach, based on the trained classifiers, is expected to adapt to the specific noisy conditions and to generalize for unseen noise conditions as well. The second idea is using subband information selectively. Pitch tracking based on subband autocorrelation is likely to

be relatively robust to various noisy audio (including the radio transmission degradation) when compared to a full-band (single band) algorithm. These ideas resulted two pitch tracking systems.

In developing the noise robust pitch tracker, we found that principal components of subband autocorrelation are a reliable feature for various pitch-related classification tasks. In evaluating the pitch tracker, we proposed a novel performance measure for pitch tracking.

1.2 Contribution

The key contributions of this thesis are the following.

- Subband Autocorrelation PCA Feature

  Instead of using the autocorrelation of the full-band signal, subband (multi-band) autocorrelation is able to find periodicity in noisy conditions. Subband autocorrelation, however, is very high dimensional and probably is redundant. To overcome these drawbacks, we propose to use a low-dimensional principal component analysis of subband autocorrelation as features for classification-based pitch tracking.

- Pitch Tracking Algorithms

  We propose two systems based on two ideas (1) subband autocorrelation PCA features and (2) classification. The first system, Subband Selection (SubSel), selectively aggregates the subband autocorrelation trained on pitch information in the subbands. The second system, Subband Autocorrelation Classification (SAcC), calculates pitch observations directly by classifiers on subband autocorrelation PCA features. Because our proposed SAcC algorithm involves a simple, trained classification stage, it can be optimized for particular speech conditions.

- Performance Measure

  Gross Pitch Error (GPE) is the standard measure reported in previous works. To properly represent the performance of pitch tracking according to its definition, we propose a new measure, Pitch Tracking Error (PTE), which reflects errors both during the voiced portion and the unvoiced portion.

1.3 Organization

In Chapter 2, the background of the thesis is given. The related algorithms, YIN and the Wu algorithm, are described in detail.

In Chapter 3, the first approach toward the classification-driven pitch tracker using subband autocorrelation PCA features, the **Sub**band **Sel**ection (SubSel) pitch tracking is presented.

In Chapter 4, a more radical approach, the pitch tracking by **S**ubband **A**uto**c**orrelation **C**lassification (SAcC), is presented.

In Chapter 5, applications of SAcC pitch tracker on a real world problem with a large-scale speech corpus with various unknown noise conditions are described.

In Chapter 6, we draw the conclusions and outline the future work.

Chapter 2

Background

This chapter provides the background for the topics covered in this thesis. Section 2.1 give introduction to the pitch tracking. Section 2.2 describes two related works on pitch tracking in detail.

2.1 Introduction

The pitch of speech is an important characteristic and has a very long history of study in literature, hence we do not attempt to provide a comprehensive review. For a more complete description of computational methods to find speech pitch, please see to the following references [Gold *et al.*, 2011; de Cheveign, 2005].

2.1.1 What is pitch?

Human perception of pitch is a psychoacoustic phenomenon. Understanding of the pitch perception process is yet incomplete: this psychoacoustic definition of pitch of speech is subjective. For computational pitch analysis, pitch refers to the fundamental frequency f0 of harmonic structure in the spectrum, so that it can be measured objectively.

	Transform-domain	Time-domain
Full-band (Single-band)	SWIPE′ [Camacho and Harris, 2008], Klapuri 06 [Klapuri, 2006], Chu [Chu and Alwan, 2012]	Talkin [Talkin, 1995], YIN [de Cheveigne and Kawahara, 2002]
Subband (Multi-band)	Klapuri 08 [Klapuri, 2008], Tan [Tan and Alwan, 2011], Sha [Sha *et al.*, 2004]	Wu [Wu *et al.*, 2003]

Table 2.1: Representative pitch estimators.

2.1.2 Computational approaches to finding pitch

Because speech is so important to humans, the development of computer algorithms imitating the human auditory system started with the advent of the programmable modern computer era [Licklider, 1951; Slaney and Lyon, 1990; Meddis and Hewitt, 1991]. Pitch estimation algorithms have a long history in various applications such as speech coding and extracting information, as well as other domains such as bioacoustics and music signal processing.

Pitch detection was loosely defined: it means finding the pitch in speech. Pitch estimation, on the other hand, focuses on estimating the pitch as close to the ground truth as possible. Pitch tracking performs pitch estimation and voicing detection (VAD) at the same time. The precise definitions of Voice Activity Detection (VAD), pitch estimation, and pitch tracking used in this thesis are given in chapter 1.

Computational approaches to finding the pitch of speech have been studied extensively. In finding periodicity, there are two basic approaches—time-domain methods which utilize autocorrelation-like operations [Wu *et al.*, 2003; de Cheveigne and Kawahara, 2002]; and frequency-domain methods that rely on Fourier transform-like operations [Klapuri, 2008; Tolonen and Karjalainen, 2000]. The use of an auditory filterbank inspired by the human auditory physiology led to harnessing of subband (multi-band) information [Licklider, 1951; Meddis and Hewitt, 1991]. Some representative pitch estimators are shown in Table 2.1.

The input of the pitch tracking/detection algorithm is sampled digital audio. The pitch output, usually in the fundamental frequency f0 (Hz) or in the period (ms), is generated typically every 10 ms. For pitch tracking algorithms, zero output values typically indicate unvoiced audio. The ground truth pitch is also generated every 10 ms and uses zero value for the unvoiced portion.

2.1.3 Time-domain approach

Time-domain approaches use a self-similarity measure to find pitch in the signal such as autocorrelation

$$r_t(\tau) = \sum_{t'=t}^{t+W} a[t']a[t'+\tau] \tag{2.1}$$

or the squared difference function

$$d_t(\tau) = \sum_{t'=t}^{t+W} \left(a[t'] - a[t'+\tau]\right)^2 \tag{2.2}$$

where $a[n]$ is a sampled digital audio signal with the time index (integer) n at a specific sampling rate (SR). For example, at a 16 kHz sampling rate (SR = 16 kHz), each index increment in n corresponds to 0.0625 ms (milli-second) in physical time.

Autocorrelation has been a successful basis both for predicting human pitch perception [Licklider, 1951; Slaney and Lyon, 1990; Meddis and Hewitt, 1991], and for machine pitch tracking. The "robust algorithm for pitch tracking" (RAPT) algorithm is based on normalized cross-correlation [Talkin, 1995]. RAPT is a basis for the `get_f0` pitch tracker software that is very popular in speech processing.

YIN is a very efficient and effective pitch detection algorithm that operates on full-band (as a single-band) as opposed to subbands (or multi-band) [de Cheveigne and Kawahara, 2002]. Nakatani and Irino proposed a f0 estimator based on instantaneous frequencies [Nakatani and Irino, 2004].

Subband autocorrelation, obtained with multiple bandpass filters and autocorrelation, resulted more robust pitch estimation. Tolonen and Karjalainen proposed

a real-time time-domain pitch analysis using two channel filterbank [Tolonen and Karjalainen, 2000]. Wu, Wang, and Brown proposed a robust multi-pitch tracking algorithm (referred to as the Wu algorithm in this paper) that integrates subband autocorrelation information [Wu *et al.*, 2003].

2.1.4 Frequency-domain approach

Frequency-domain approaches use variants of the Fourier transform to analyze the frequency components of audio with a goal of finding the fundamental frequency, f0.

For the fullband algorithms, Joho *et al.* proposed a three-stage pitch tracking system by connecting the partial pitch contours obtained by HMM tracking of the Non-Negative Factorization (NMF) pitch estimation [Joho *et al.*, 2007]. The Sawtooth Waveform Inspired Pitch Estimator (SWIPE) finds F0 that gives most significant harmonics peak-to-valley ratio by calculating the normalized inner product of the cosine kernel with the frequency-warped square-root of the hann-windowed spectrum with Harmonic weighting [Camacho and Harris, 2008; Camacho, 2007]. SWIPE′ (SWIPE-PRIME) is an improvement over SWIPE that uses the cosine the cosine kernel harmonics corresponding to the prime numbers only.

For the subband algorithms, Sha *et al.* proposed a pitch tracking choosing from multi-band pitch estimations based on classification on by coarser subband features [Sha *et al.*, 2004]. Tan and Alwan proposed a robust pitch estimation based on summary autocorrelogram integrating the subband autocorrelation weighted by the subband SNRs inferred using comb filters [Tan and Alwan, 2011]. These techniques have even been successfully applied to mixtures containing multiple pitches [Klapuri, 2008; Wu *et al.*, 2003; Tolonen and Karjalainen, 2000].

2.1.5 Subband Autocorrelation

Subband autocorrelation is one of the major components for our approaches, we present the detailed description here.

The input audio signal $a[n]$, where n is the sample index at a sampling rate (SR), is expanded into s subband signals $x_l[n]$, $l = 1 \dots s$, using an auditory filterbank. The auditory filterbank is implemented with a bank of 4th order Infinite Impulse Response (IIR) gammatone bandpass Equivalent Rectangular Bandwidth (ERB) filters with the center frequencies uniformly spaced in log scale. The configuration for the auditory filterbank can be controlled by (1) the lowest center frequency f_1, (2) the number of bands per octave (BPO), and (3) the number of the subbands s. We used $f_1 = 100$ Hz, BPO $= 16$, and $s = 48$, which gives the f_l range from $100 \dots 800$ Hz.

The normalized subband autocorrelation A_l (hereafter, subband autocorrelation) is calculated for each subband every 10 ms (at $t_b = (b-1) \times 0.001$ s $\times$ SR for frame $b \in \{1, \dots, N_F\}$ where N_F is the number of the analysis frames.) and τ is the autocorrelation lag:

$$A_l(t_b, \tau) = \frac{r_l(t_b, \tau)}{\sqrt{r_l(t_b, 0)}\sqrt{r_l(t_b + \tau, 0)}} \tag{2.3}$$

where

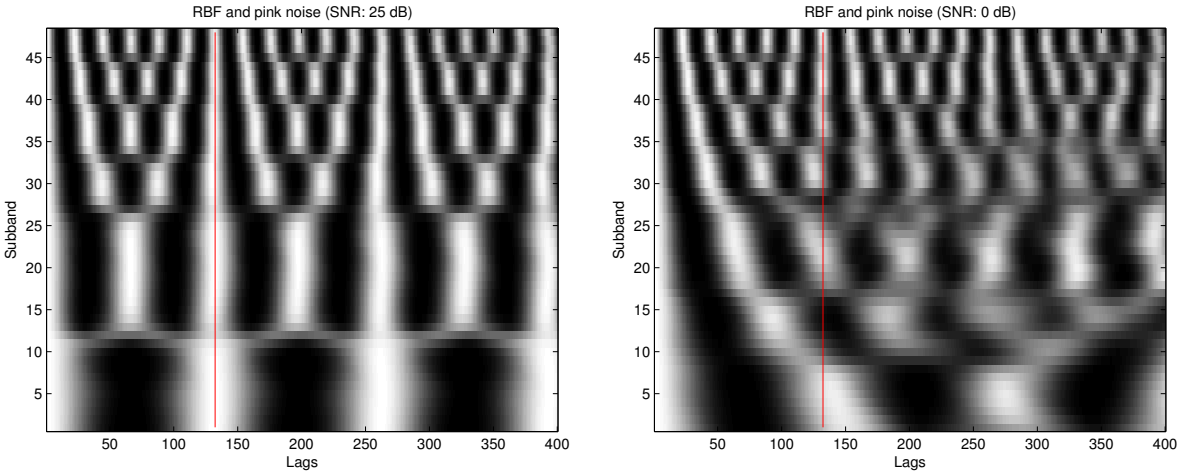
$$r_l(t_b, \tau) = \sum_{n=-N/2}^{N/2} x_l[t_b + n] x_l[t_b + n + \tau] \tag{2.4}$$

and the window length $N = 400$, corresponding to 25 ms at SR $= 16000$. The largest lag is also $\tau = 400$, i.e., down to 40 Hz fundamental at 16 kHz sampling rate. Note that this is an unwindowed autocorrelation.

2.1.6 Post-processing

Pitch detection algorithms are frequently enhanced by various post-processing methods, from a simple median filter to a complex statistical model. A local estimate finds the most likely periodicity around a short interval [de Cheveigne and Kawahara, 2002]. Hidden Markov models (HMMs) have been used to refine the pitch tracking results by imposing sequential consistency [Wu *et al.*, 2003; Lee and Ellis, 2006; Tan and Alwan, 2011].

2.1.7 Challenges



(a) 25 dB SNR

(b) 0 dB SNR

Figure 2.1: The subband autocorrelation of voiced speech in a bandlimited audio corrupted by pink noise at (a) 25 dB SNR and (b) 0 dB SNR. The f0 lag (red line) is clearly marked by a common ridge across all subbands in (a); this common pitch ridge is degraded in (b).

Determining the fundamental period of voiced speech signals (hereafter, "pitch tracking") is important in a range of applications from speech coding through to speech and prosody recognition and speaker identification. However, high-accuracy pitch tracking is difficult because of the wide variability of periodic speech signals [Talkin, 1995]. There are many speech phenomena that can make the true pitch hard to identify or even define.

For example, pitch tracking algorithms can be used to find speech automatically within large-scale unlabeled recordings such as user-generated audio content and surveillance recordings. In such recordings, the audio is often corrupted by additive noise and/or communication channel degradation. This thesis is motivated by the problem of identifying and recognizing speech signals in such low-quality radio transmissions.

While periodic signals have obvious features in these domains, they also exhibit some ambiguity, leading to the well-known "octave errors" and other phenomena. Moreover, additive noise can affect autocorrelation in ways that are difficult to model. When the background noise contains strong periodic components (such as air-conditioning), pitch tracking result in many false detections of pitch [Lee and Ellis, 2006].

Figure 2.1 show examples of the subband autocorrelation of voiced speech in a bandlimited audio corrupted by pink noise at (a) 25 dB SNR and (b) 0 dB SNR. The subband autocorrelation of the cleaner 25 dB example has strong white ridge at the fundamental lag position (the vertical red line) across all subbands. In the subband autocorrelation of the 0 dB SNR example, information relating to the pitch after the severe degradation can be seen at the f0 lag position in a majority of the subbands. Hence, we decided to build classification-based noise robust pitch tracker using subband autocorrelation as a feature.

In many cases, the previous pitch trackers report single digit error values in Gross Pitch Error (GPE) under various additive noise conditions. Rather than using GPE as a golden measure to optimize, we try to identify the performance of pitch tracking to fit its objective, obtaining both f0 estimation and voicing decision.

Although we generally believe the correctness of the hand-labeled ground truth pitch provided in standard pitch corpora [Bagshaw *et al.*, 1993; Plante *et al.*, 1995], the ambiguity of start and end points of pitch makes the ground truth unreliable at the boundary of the voiced speech. In turn, the effort to reduce the small error made by pitch trackers at the boundaries is marginally rewarded.

Rather than improving the accuracy in the boundary of voiced speech, we focus on improving pitch tracking for the case of more disruptive degradation, such as low-quality band-limited audio.

2.2 Related works

In this section, we describe two related algorithms in detail: YIN, a pitch estimator, and, the Wu algorithm, a pitch tracker.

2.2.1 YIN Algorithm

YIN is a very efficient and effective full-band pitch estimation algorithm [de Cheveigne and Kawahara, 2002]. YIN operates in 6 steps: (1) autocorrelation; (2) difference function; (3) cumulative mean normalized difference function calculation; (4) absolute thresholding; (5) parabolic interpolation; and (6) best local estimation. The power of YIN comes from the robust difference function.

The objective of the first three steps is to get an initial measure of periodicity using autocorrelation-derived methods. The autocorrelation $r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau}$ which is used to calculate the difference function $d_t(\tau) = \sum_{j=1}^{W}(x_j - x_{j+\tau})^2 = r_t(0) + r_{t+\tau}(0) - 2r_t(\tau)$ in the first and the second steps leads to the third step to improve the error rate. In the third step, the cumulative mean normalized difference function is calculated as follows:

$$d'_t(\tau) = \begin{cases} 1, & \text{if } \tau = 0 \\ d_t(\tau)/c_t(\tau), & \text{otherwise} \end{cases} \tag{2.5}$$

where

$$c_t(\tau) = \frac{\sum_{j=1}^{\tau} d_t(j)}{\tau}.$$

The difference function starts with 0 at $\tau = 0$ and may stay close to 0 for small τ. To avoid being misled by these small dissimilarity values for very small periods, the cumulative mean normalized difference function starts with a fixed value, 1, by definition and stays above 1 near $\tau = 0$. As a result, (2.5) avoids the obvious minimum and near minimum values near zero-lag and focuses on distinguishing the true minimum at the fundamental period from other smaller minimums at the harmonic lags.

In the fourth step, absolute thresholding chooses the pitch period estimate $\hat{\tau}$ by finding the smallest value of τ that gives a minimum of d' smaller than the threshold θ_y. If there is no such τ, $\hat{\tau} = \arg_\tau \min d'(\tau)$. The absolute thresholding further reduces the error rate by decreasing period-multiple (suboctave) errors. The fifth step is the parabolic interpolation which improves accuracy of the pitch period estimate, especially in high f_0. In the sixth step, the best local estimate is proposed for stability. At t, the initial estimate of period is $\bar{\tau} = \hat{\tau}_{\hat{\theta}}$ where $\hat{\theta} = \arg_\theta \min d'_\theta(\hat{\tau}_\theta)$ where $t - T_{max}/2 \leq \theta \leq t + T_{max}/2$ and the interval T_{max}=25 ms. Then, the best local estimate τ^* is $\arg_\tau \min d'_t(\tau)$ where $0.8\bar{\tau} \leq \tau \leq 1.2\bar{\tau}$.

To use pitch estimators as pitch trackers, voicing decision is needed. YIN provides aperiodicity output which is inversely proportional to the voicing. Simple thresholding of this aperiodicity can be used as voicing decision. But, thesis single voicing index tend to change rapidly and might cause un wanted jumps between voiced and unvoiced outputs. These unwanted voiced/unvoiced transitions can be reduced by post-processing.

Although YIN is a very successful and stable time-domain pitch estimation algorithm, it does not have a post-processing stage for pitch tracking. To compare the performance of YIN against the Wu algorithm, which does include post-processing, we added a HMM pitch tracking back-end to YIN, giving YIN-HMM. The largest gross error improvement came from introduction of the (cumulative) difference function; thus, we built the YIN-HMM tracking based on the cumulative mean normalized difference function d' (2.5). According to our experiments, however, the performance improvement of YIN-HMM from YIN was marginal.

2.2.2 Wu Algorithm

Wu, Wang, and Brown proposed a robust multi-pitch tracking algorithm (henceforth, the Wu algorithm) [Wu *et al.*, 2003] that combines pitch peaks identified in per-subband autocorrelations, followed by HMM pitch tracking. By separately searching

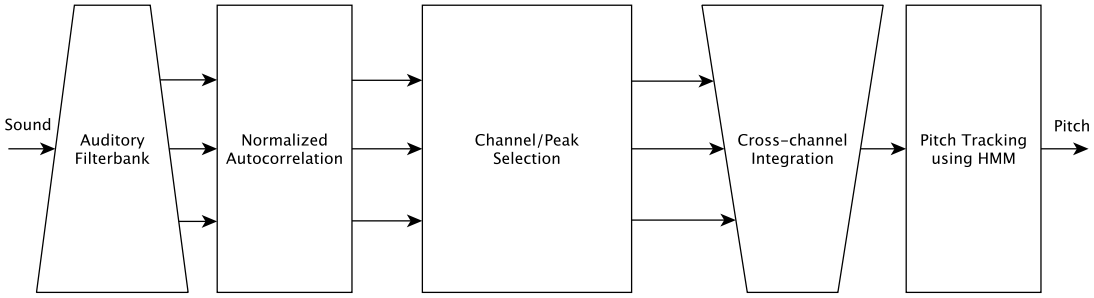


Figure 2.2: The diagram of the Wu pitch tracking system.

for pitch periodicity in multiple frequency regions, this algorithm achieves robustness against noise that might corrupt the information in some subbands, while leaving others relatively unaffected. Since this is the basis of our system, we now describe it in more detail.

The Wu algorithm performs the following steps as in Figure 2.2: (1) cochlear filtering; (2) normalized correlogram; (3) peak selection; (4) subband selection; (5) cross subband integration; and (6) pitch tracking with hidden Markov model (HMM). The details for each step are given in the below.

The first two steps are used to calculate the subband autocorrelation (2.3).

In the peak selection step, the Wu algorithm picks the candidate lags for pitch period in the following manner. As a first attempt to get the lag corresponding to the pitch period, the lags with the local maxima of autocorrelation are inspected for each subband. For the initial candidate lags, the algorithm picks only the lags that give the maximum autocorrelation using a wider window for search.

A period is selected if the normalized autocorrelation maximum is greater than $\theta = 0.945$. (The original paper used a different criteria for high-frequency subbands, but in our implementation we used this single criteria throughout without apparent impact.) Selected maxima from different subbands are combined into a single score by spreading each peak according to an empirical Laplacian fit, then averaging across all subbands. The result can be interpreted as the likelihood of the observations O_t at

time t given a period hypothesis τ, i.e., $P(O_t|\tau)$. One interpretation of this combination is that it allows any true periodicity to emerge as the consensus of channels that are minimally affected by the interference. Channels that are corrupted by noise will either fail to reach the normalized autocorrelation threshold, or contribute random period estimates which will be washed out in the combination process.

The Viterbi path through a Hidden Markov Model (HMM) is used to smooth the pitch track, and to differentiate no-pitch and one-pitch states. (The original implementation also accommodated two-pitch states to track mixture signals.) The HMM finds the period sequence that maximizes the likelihood of the autocorrelation observations O_t by optimizing the sum across time of

$$P(O_t|\tau_t, \tau_{t-1}) = P(O_t|\tau_t)P(\tau_t|\tau_{t-1}) \tag{2.6}$$

where τ_t and τ_{t-1} are the pitch periods at frames t and $t-1$, and the transition probabilities $P(\tau_t|\tau_{t-1})$ are optimized empirically. $\tau_t = 0$ is a special case meaning no-pitch, whose probability is set to a fixed percentile of the probabilities estimated for actual pitches. This indirect way to predict no-pitch frames is addressed in Chapter4.

Although our implementation of the Wu system differs from the original, it has performance essentially equivalent to the c-code released by the original authors for single-pitch conditions according to our tests on the sample audios provided by the original authors[1].

[1] http://www.ee.columbia.edu/~bsl/projects/wu/

Chapter 3

Subband Selection (SubSel) Pitch Tracker

This chapter presents our first approach toward the classification-driven pitch tracking system using subband feature, the Subband Selection (SubSel) pitch tracker.

The proposed SubSel system is presented in section 3.1. The dataset and the experimental setup are explained in section 3.2. Section 3.3 discusses the results. Finally, a summary is drawn in section 3.4.

3.1 The SubSel Pitch Tracker

Figure 3.1 shows the diagram of the proposed Subband Selection (SubSel) pitch tracking system. This approach is closely based on the Wu algorithm (Figure 2.2), but with the idea of using a more complex, trained classifier to decide when a channel is providing useful pitch information, rather than the simple normalized autocorrelation threshold. A filterbank is used to generate subband audio channels as described in section 3.1.1. To avoid overfitting and reduce complexity, the autocorrelation is compressed by Principal Components Analysis (PCA) dimensionality reduction (section 3.1.2). A simple linear classifier per channel/subband is trained on the reduced

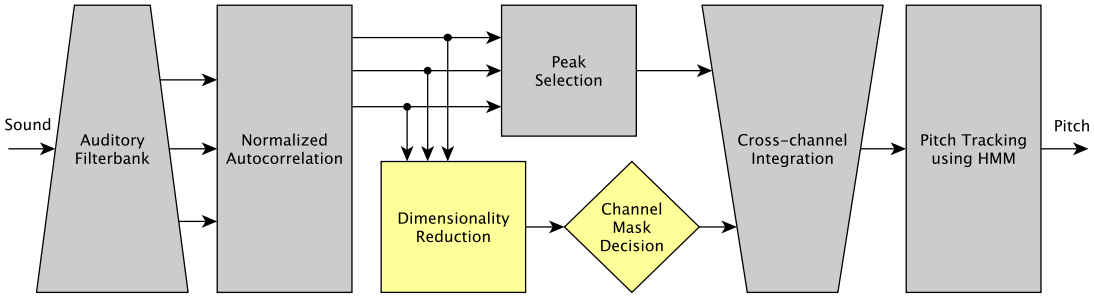


Figure 3.1: The diagram of the proposed Subband Selection (SubSel) pitch tracking system.

k-dimensional feature space (section 3.1.3). To train the classifiers, two subband mask labels for training classifiers are considered (section 3.1.4). To capture the pitch information shared across the adjacent subbands, adjacent subband features are concatenated (section 3.1.5).

3.1.1 Subband Filtering

The subband filtering is used to selectively process different frequency ranges according to their local Signal-to-Noise Ratio (SNR). Frequency ranges with relatively higher SNRs will contain more reliable information than those with lower SNRs. For instance, when a signal has been through a bandlimiting channel (such as radio transmission), the range of SNRs across different frequency bands can be very large.

For subband filtering, a set of 48 bandpass filters (4th order IIR gammatone ERB filters) with center frequencies ranging from 100 Hz to 800 Hz uniformly spaced in log scale with 16 bands per octave density is used. The gain frequency responses of the subband filters are shown in Figure 3.2.

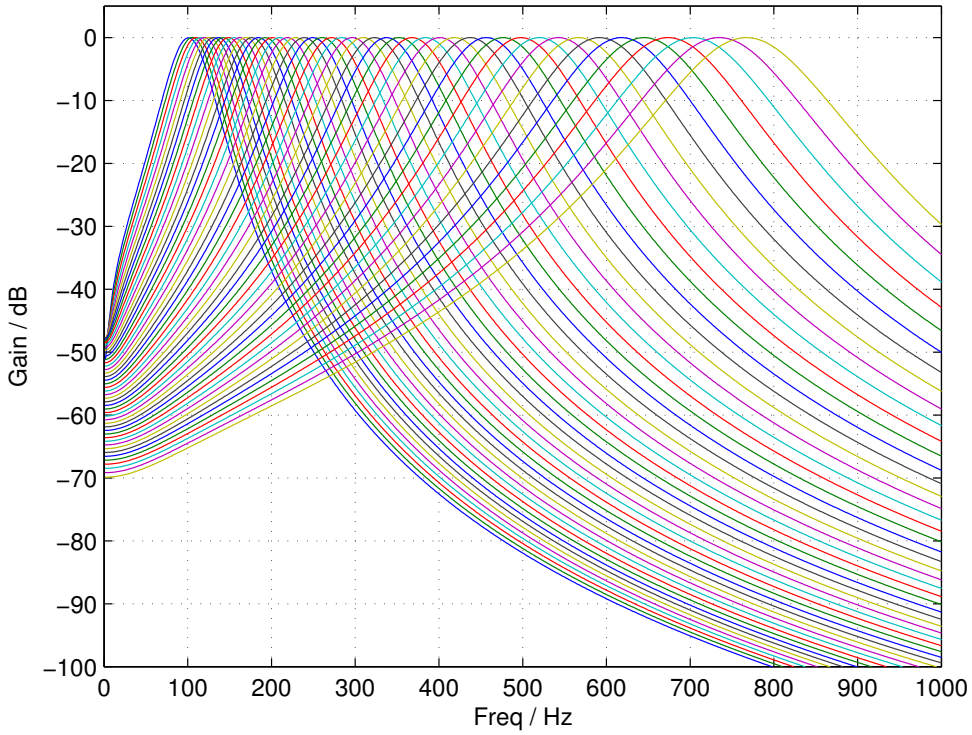


Figure 3.2: The gain frequency response of the 48 subband filters.

3.1.2 Subband PCA Dimensionality Reduction

Subband autocorrelation (2.3) has 400 dimensions, which corresponds to the largest lag τ_{max} for the autocorrelation, for each subband for each 10 ms frame. Subband PCA is used to reduce dimensionality. PCA is performed over a large amount of data drawn from the training set. The principal components are sorted in the decreasing order of the corresponding eigenvalues. The first k principal components are chosen to form the k-dim PCA. The eigenvalues of PCA components decrease very fast as the number of principal component increases in the sorted PCA as in Figure 3.5.

The first five principal components of PCA for the subband 1, 21, and 41 (left, middle, and right columns) learned from our dataset (described in section 3.2.1) are shown in in Figure 3.3. The legend "All" indicates that the PCA is learned from the all utterances. The legend 'Pink' indicates that the dataset is corrupted by additive pinknoise only. The legend "RBF" indicates that the dataset is corrupted by Radio

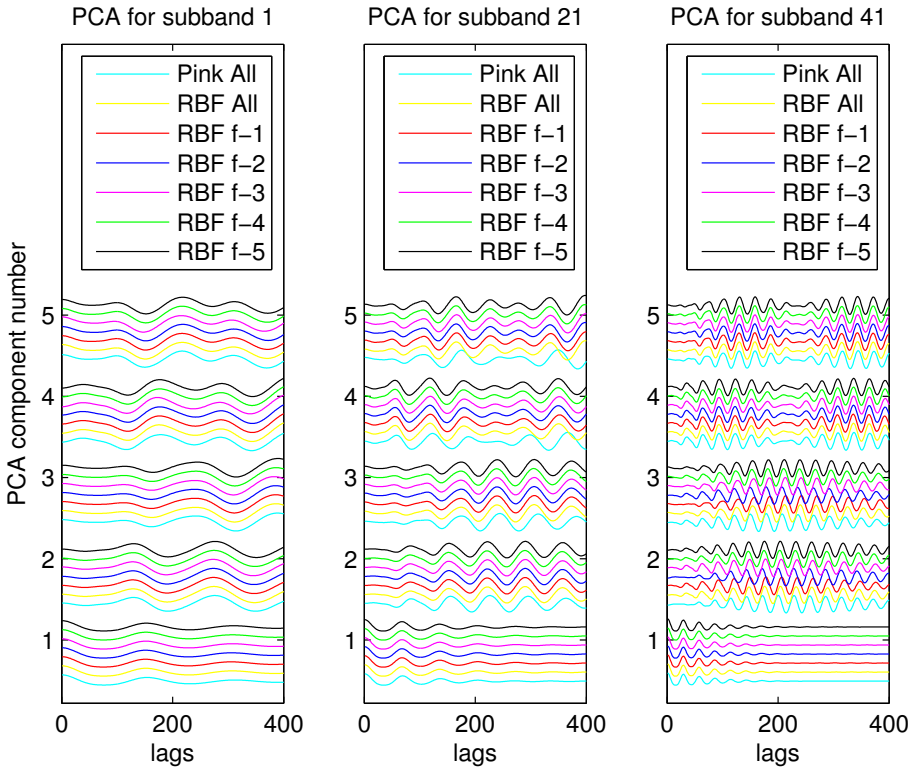


Figure 3.3: The first five principal components (eigennumbers in y-axis) of subbands $l = 1, 21, 41$ (left, middle, and right columns) learned from KEELE dataset. The legends indicate the PCA conditions.

Band Filtering and additive pinknoise. The legend 'f-i' where $i = 1, \ldots, 5$ indicates that the PCA is learned from the i-th fold training dataset of the 5-fold CV. The principal components learned from the dataset with various conditions are basically the same. In the experiments, we used the PCA learned from all the data with degradations corresponding to the test conditions.

The individual subband principal components show interesting patterns. The first principal component shows a decaying sinusoidal shape with frequency proportional to the corresponding subband frequency. The subsequent principal components show in-phase and out-of-phase sinusoidal shapes with frequencies proportional to the corresponding subband frequency modulated by a fundamental and harmonic frequencies

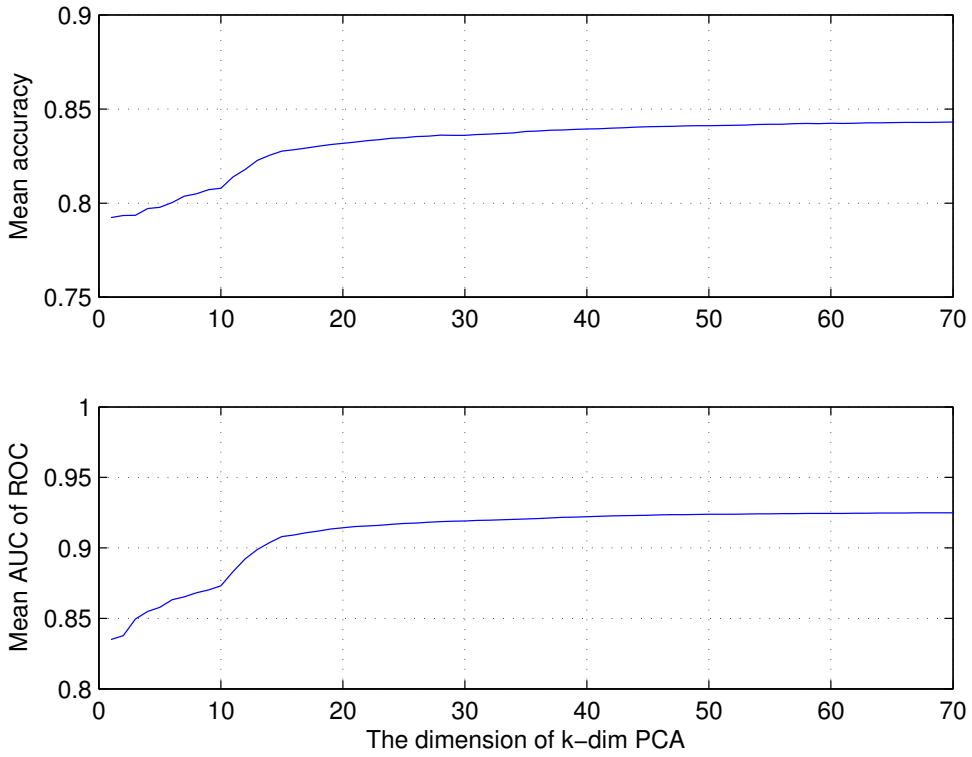


Figure 3.4: The mean accuracy of k-dim PCA subband selection (top panel) and the AUC of ROC curve of the subband selection (bottom panel).

of the analysis window length.

In the next section, we describe how the PCA features are used to classify subbands as noisy or clean. To illustrate the impact of different configurations, Figure 3.4 shows the performance of subband linear classifiers with k-dim PCA for k from 3 to 70. In particular, the plot of the subband mean accuracy (top panel) of k-dim PCA linear classifiers (predicting the Wu criterion) demonstrate that increasing k does not linearly improve the performance. The mean accuracy is already 79 % at $k = 3$ and does not increase very much as k increases. The Area Under the Curve (AUC) of Recriver Operator Characteristics (ROC) curve of the subband selection classifier (bottom panel) in Figure 3.4 shows AUC reaches a plateau using the few principal components with the largest eigenvalues.

Figure 3.5 shows that most energy is concentrated in the few largest principal

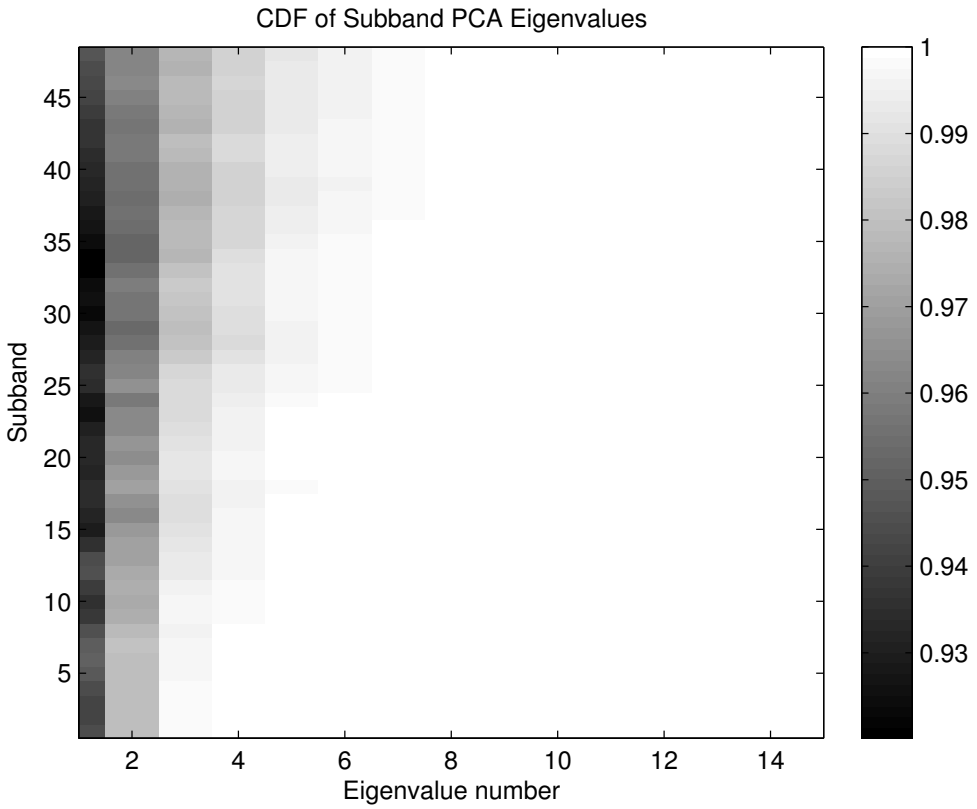


Figure 3.5: The Cumulative Density Function (CDF) of Subband PCA eigenvalues.

components. Since the objective of the proposed algorithm is pitch tracking, not the reconstruction of some intermediate values, the optimal value of k for PCA should be determined empirically as the one that gives the best pitch tracking. We will discuss this k decision in section 3.3.

3.1.3 Linear Classifier for Subband Selection

In the original Wu algorithm, individual subbands are included or rejected as too noisy based on a simple threshold (0.945) applied to the largest peak in their normalized autocorrelation. From the subband PCA output, the following classification is used for subband selection instead of the Wu criterion. We used only a linear classifier because we believed it would be sufficient for replacing a simple threshold with a decision that

incorporated a wider basis of information. The following is done for each subband $l = 1 \dots s$; the subscript l is dropped in the derivation for the simplicity.

With k-dimension PCA dimension reduction, the data of size $n \times m$ becomes a $n \times (k+1)$ matrix $\mathbf{X}$, which is augmented by 1's for a constant term; and the label $\mathbf{y}$ of a vector of length n with $y_i \in \{-1, 1\}$ for all $i = 1, \dots, n$. We model the label $\mathbf{y}$ with a linear classifier $\mathbf{X}\theta$, where θ is a $(k+1)$-dimensional coefficient vector.

Using minimization of the mean squared error as an optimization objective, the loss function $L(\theta)$ is

$$L(\theta) = |\mathbf{X}\theta - \mathbf{y}|^2 = (\mathbf{X}\theta - \mathbf{y})^{\mathrm{T}} (\mathbf{X}\theta - \mathbf{y}). \tag{3.1}$$

Using matrix algebra, the gradient of (3.1) can be expanded to

$$\nabla_\theta L(\theta) = 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\theta - 2\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{3.2}$$

By solving $\nabla_\theta L(\theta) = 0$, the optimal $\hat{\theta} = \operatorname{argmin}_\theta L(\theta)$ can be found as

$$\hat{\theta} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{3.3}$$

The subband classification is given by $f(\mathbf{X}\hat{\theta})$ where

$$f(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

3.1.4 Labels used to train classifier

Unlike pitch, for which we have hand-labeled ground truth [Plante *et al.*, 1995], the labels for subband selection have no ground truth. To train the classifiers, the ground truth subband selection labels are required. Two approaches to generate the ground truth subband labels are proposed: (1) Wu criterion and (2) ground truth pitch criterion.

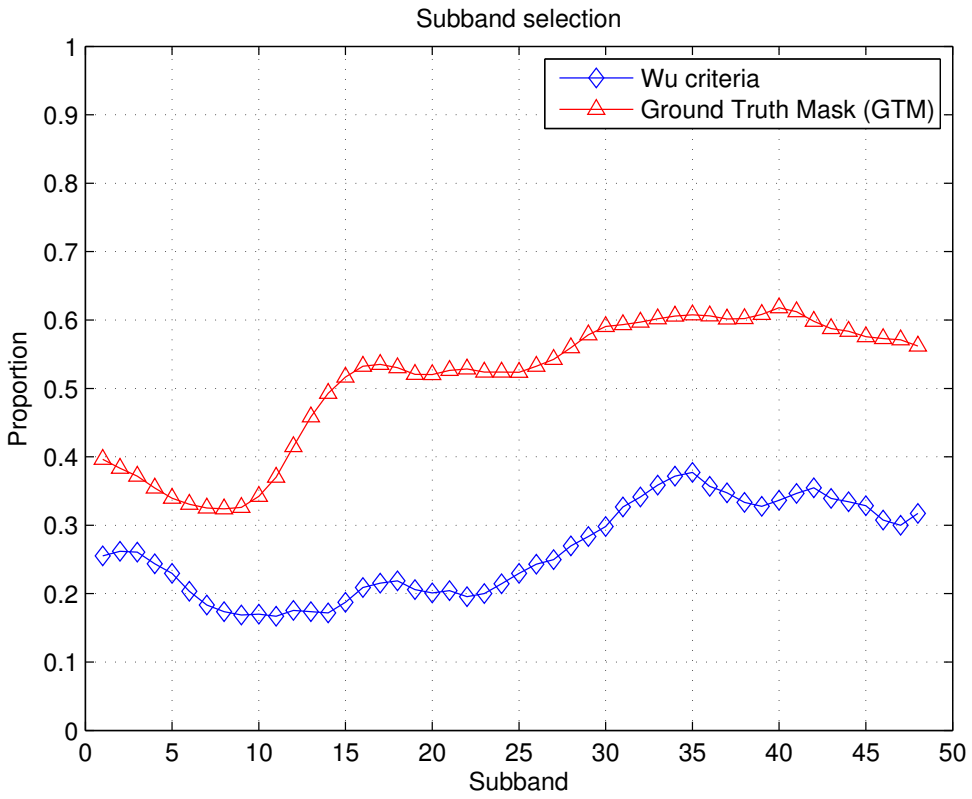


Figure 3.6: The distribution of subband selection by the Wu criterion and the Ground Truth Mask (GTM) criterion.

The Wu algorithm selects subbands by looking at the maximum value of the normalized subband autocorrelation function $A_l(t_b, \tau)$ (2.3). As a starting point, the subband classifiers were trained to learn this simple criterion. If the proposed approach can properly learn this criterion, the performance of the Wu algorithm and the proposed algorithm should converge as the behavior of two algorithms converge.

Another subband selection label criterion, which we call the Ground Truth Mask (GTM), is based on the manually-labeled ground truth pitch. The GTM criterion labels that the subband is positive when the ground truth pitch is consistent with one of the prominent autocorrelation peaks in that subband. Since the center frequencies of the subbands are different, the tolerance γ_l of ground truth pitch to the selected pitches for the subband l can be increased as the center frequency of the subband

f_l decreases with a parameter ν: $\gamma_l = 2\nu\,(f_{max}/f_l - 1) + 2$ (lags). By increasing ν, subbands with lower center frequencies can allow more tolerance between the truth pitch and the selected pitches.

Figure 3.6 shows the proportion of positive labels in all subbands of the Wu criterion (blue diamond) and the GTM criterion with pitch tolerance parameter ν =1 (red triangle). The GTM criterion had more positive labels than the Wu criterion across all subbands.

3.1.5 Adjacent Subbands

The information of subband autocorrelation is centered around the center frequency of the subband. Accordingly, it was observed that adjacent subbands contain similar autocorrelation characteristics. More specifically, adjacent subbands show the periodicity of similar harmonics; hence, they share pitch candidates. Subband integration will find the pitch period candidate that is the most preserved across all subbands, rejecting the higher order harmonics that are detected in small portions of subbands.

To take the adjacent subbands information into account, the reduced k-dim features of adjacent subbands were concatenated to form $k \times q$ dimension features, and classifiers were trained on these concatenated feature sets instead of single subbands. The best value of q is found empirically found as q=3 using candidates q=3, 5, 7, and 9. For subbands at the top and bottom of the frequency axis where adjacent subbands of equal distances are not available, the q closest subbands of unequal distances are used instead.

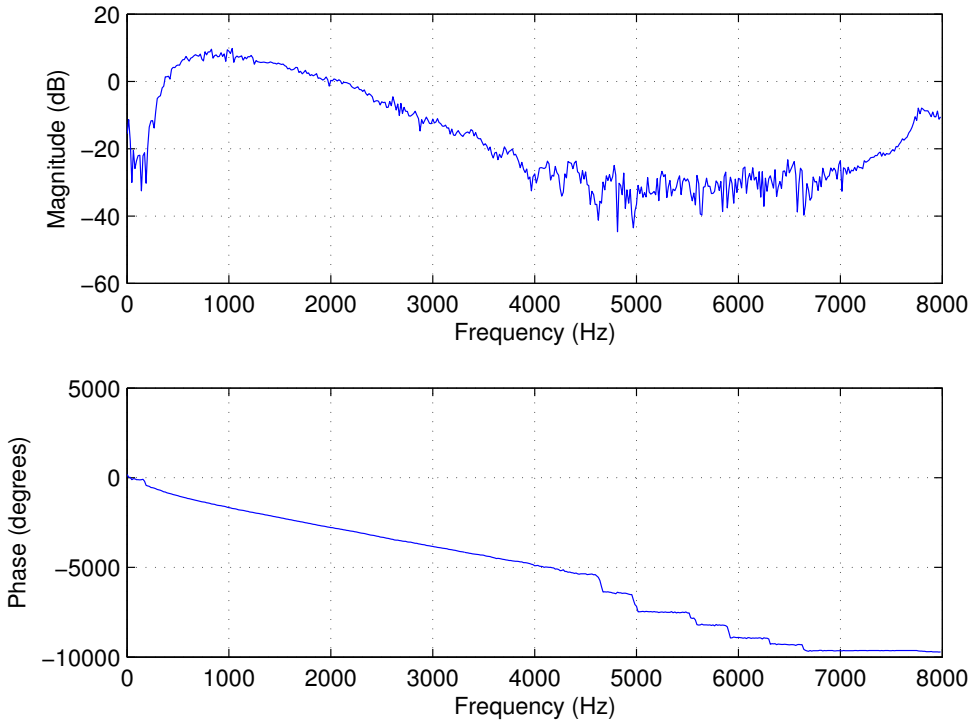


Figure 3.7: The magnitude and phase frequency response of a typical radioband filter (RBF) estimated from the radioband-channel-transmitted speech.

3.2 Experiments

3.2.1 Experimental Setup

While development was performed on a small set of speech examples collected and pitch-marked in-house, we used the KEELE [Plante *et al.*, 1995] and FDA [Bagshaw *et al.*, 1993] corpora for evaluation. These datasets used electroglottograph (EGG) sensors to directly measure vocal fold closure events as an independent basis for the ground truth. As we discussed in chapter 1, we are specifically interested in speech whose spectrum has been shaped by radio transmission. The reason for interest is two-fold: (1) there are large-scale speech corpora generated under this type of noise condition; (2) the performance of the traditional pitch trackers degrades severely for the radio transmitted speech. To simulate such a channel, we used a filter estimated

from actual speech transmitted across a handheld narrow-FM walkie-talkie[1]. The radioband filter (RBF) is modeled as an FIR filter and has a bandpass characteristic spanning approximately 500 Hz ... 2 kHz. The frequency response of our typical RBF is shown in Figure 3.7.

The performance of pitch tracking was measured by the Voiced Error (VE). The VE is defined as the proportion of correctly predicted non-zero pitches. (System outputs during periods that were marked unvoiced by the ground-truth have no effect on Voiced Error (VE). This limitation is addressed later in chapter 4.) A predicted pitch value is considered correct if it falls within 20 percent of the true value. The ground truth provided values every 10 ms.

For each dataset, we employed 5-fold cross-validation (CV), in which the dataset was divided with 80% used for training and the remainder for test, and the training repeated five times until all the dataset had been used for test. The overall performance is the average of these five test results. Since we focus on pitch tracking, not on predicting no-pitch regions, the probability of no-pitch state in the HMM pitch tracking was set to a very small value 10^{-10} to discourage the system from reporting no-pitch.

3.3 Discussion

The pitch tracking results of the proposed SubSel algorithm and the Wu algorithm on a speech sample corrupted by the RBF filtering and the additive pinknoise at 5 dB SNR are shown in Figures 3.8 and 3.9. For the proposed algorithm, the five adjacent subband features were concatenated to learn the ground truth pitch mask with pitch tolerance parameter ν=1.

The subband selection masks in Figure 3.8 (b) and Figure 3.9 (b) show that the proposed algorithm selects more subbands than the Wu algorithm on the pitched

[1] http://labrosa.ee.columbia.edu/projects/renoiser/

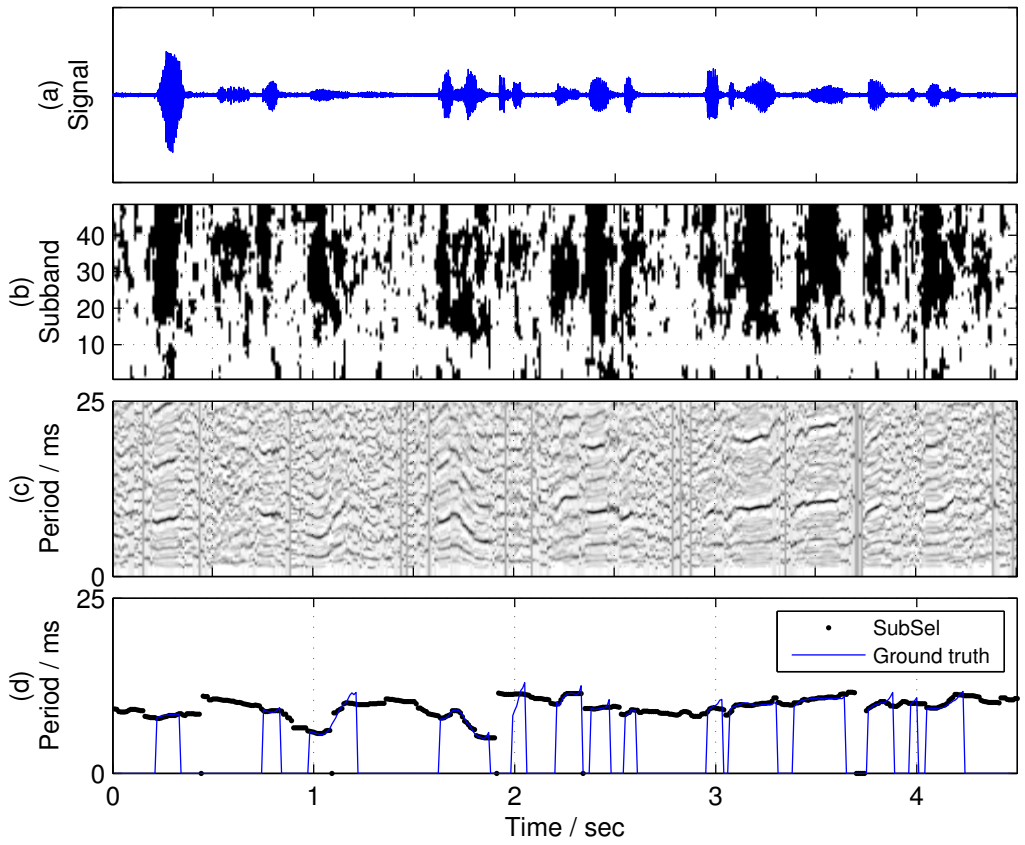


Figure 3.8: The audio (a), the subband selection (b), the pitch likelihood (c), and the pitch tracking (d) of the SubSel algorithm on an example speech corrupted by the RBF filtering and the additive pinknoise at 5 dB SNR.

speech portion.

In Figure 3.8 (d) and Figure 3.9 (d), the blue line is the ground truth pitch; and the black dots are the pitch tracking output. The vertical blue lines indicate onset/offset of VAD based on the ground truth pitch.

In the third pitched speech region (around 1-1.2 s), SubSel makes fewer harmonic errors than the Wu algorithm. In the fifth and the sixth pitch regions (around 2-2.3 s), SubSel does not make the same harmonic/sub-harmonic errors that the Wu algorithm makes.

The improvement comes from the pitch probability in the non-pitch regions. In

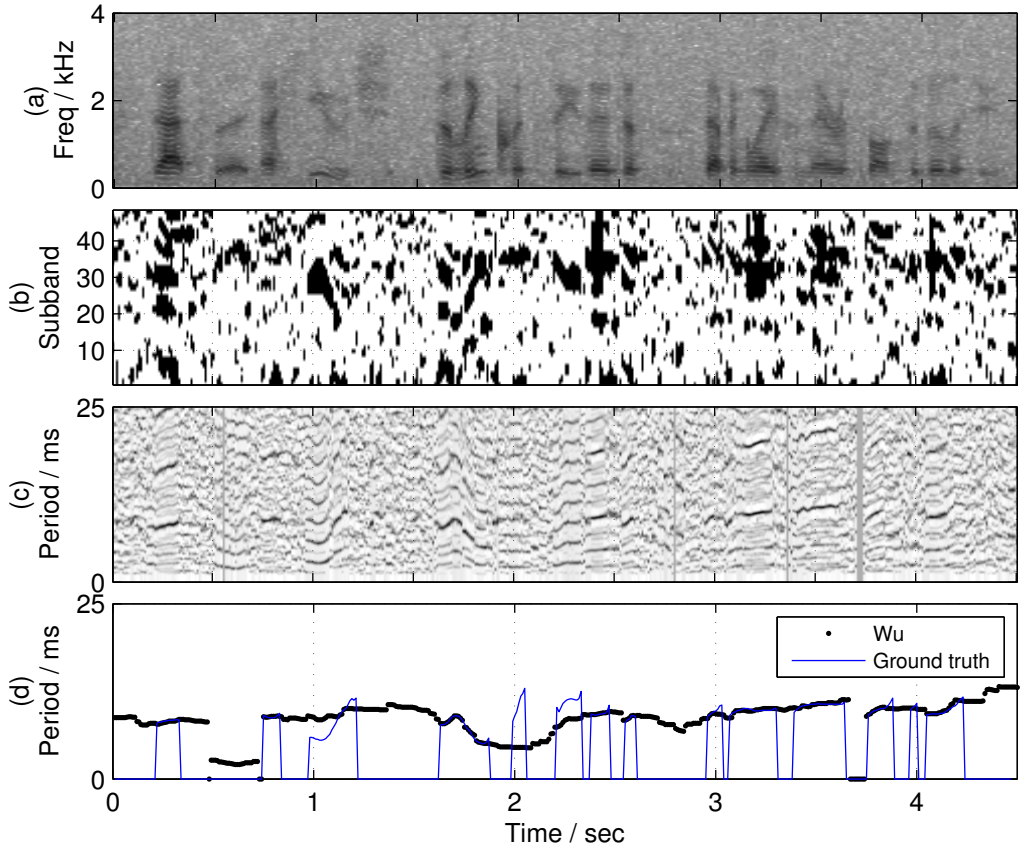


Figure 3.9: The spectrogram (a), the subband selection (b), the pitch likelihood (c), and the pitch tracking (d) of the Wu algorithm on an example speech corrupted by the RBF filtering and the additive pinknoise at 5 dB SNR.

other words, SubSel correctly disconnects the pitch tracks in the non-pitch regions. On the other hand, the Wu algorithm connects the pitches incorrectly across the non-pitch region.

Figure 3.10 shows the overall pitch tracking performance of Wu, YIN, and the proposed SubSel algorithms trained on the ground-truth pitch mask using 5-fold CV on the FDA dataset corrupted with additive pink noise, with and without RBF filtering, as a function of the noise level (SNR). The mean VE in log-scale vs SNR of Wu, YIN, and SubSel (GTPvarAC5) algorithms are shown. For PCA, $k = 30$ is used. The legend 'var' indicates that the subband classifier is trained using the ground

	SNR (dB)							
	25	20	15	10	5	0	-5	-10
	KEELE (pink noise)							
Wu	4.1	3.98	4.33	4.43	**5.74**	**9.97**	**36.53**	**68.44**
YIN	**3.83**	**3.83**	**3.89**	**4.11**	6.42	20.67	55.11	73.95
SubSel	5.14	5.27	5.79	6.5	8.54	17.35	50.56	76.68
	KEELE (RBF pink noise)							
Wu	**4.51**	**4.69**	**5.27**	**8.1**	**14.89**	38.23	63.45	74.77
YIN	18.76	18.75	19.37	21.8	31.64	50.99	70.16	76.07
SubSel	4.96	5.37	6.42	9.47	15.1	**35.38**	**62.82**	**69.28**
	FDA (pink noise)							
Wu	**2.33**	**2.45**	3.28	3.33	7.15	12.51	34.62	**65.79**
YIN	2.61	2.6	**2.6**	**2.76**	**3.11**	**7.84**	**33.61**	72.01
SubSel	3.52	3.85	3.69	5.6	8.14	19.31	34.76	79.81
	FDA (RBF pink noise)							
Wu	**2.33**	**3.09**	5.54	7.49	17	39.54	**57.37**	**75.68**
YIN	11.94	11.94	12.11	13.67	20.89	34.85	59.67	76.84
SubSel	4.78	4.41	**5.4**	**6.28**	**9.12**	**21.83**	57.47	76.65

Table 3.1: Mean VE (%) of Wu, YIN, and SubSel on KEELE and FDA corpora

truth pitch mask labels with variable pitch tolerance proportional to the subband center frequency. The legend 'AC5' indicates that features of five adjacent subbands are concatenated to form the features of one subband. The errorbars indicate the standard deviation over 5-fold CV.

The VE results on KEELE and FDA corpora (cross-dataset train/test result) are shown in Table 3.1. In fact, the last two rows of Table 3.1 are the same result as Figure 3.10, showing the numbers as opposed to the log-scale plot. For the RBF and pink noise conditions, SubSel performs the best for mid-to-high (15, 10, 5, and 0 dB)

SNR conditions, which is our main target since there is more room for performance improvement. The radioband filtering degrades the VE for all algorithms, with the full-band YIN algorithm showing a drastic degradation. The proposed algorithm had the least variation in VE under the radioband filtering. In general, subband algorithms (Wu and SubSel) performed more robustly than the single-band (full-band) algorithm (YIN) under RBF conditions.

Since the SubSel results are cross-corpora, the lower performance in KEELE corpus re-confirms that KEELE corpus, which contains speech of ten speakers, is more general (diverse) than FDA corpus, which contains speech of two speakers. In other words, the SubSel trained on FDA corpus seems to be too specific that it cannot generalize on KEELE corpus, which is more diverse.

The small performance improvement at high SNRs comes mainly due to ambiguous frames at the edge of speech utterances. Hence, we focus on performance improvement at higher noise conditions. SubSel focuses on these mid-to-high (15, 10, 5, and 0 dB) SNR conditions, where speech is still intelligible to human listeners, but common automatic methods tend to fail. The performance improvement of SubSel over the Wu system makes sense since SubSel combines information from more sources (subbands) as in Figure 3.8 (b) and 3.9 (b).

The performance of the proposed algorithm trained on the Wu criterion was not included in the result because it performed very similarly to the Wu algorithm. We also tried to find the best threshold for subband classifiers. But changing the threshold for the linear classifier such that the classifier gives the maximum accuracy on the training data gave a similar VE result as using a fixed threshold of zero for the subband classifiers.

Although YIN is a very successful and stable time-domain pitch detection algorithm, it does not have a HMM post-processing tracking stage. To make the performance comparison of YIN against the Wu algorithm, we introduced pitch tracking to YIN using an HMM. The largest gross error improvement in YIN comes from

introduction of the (cumulative) difference function; we built a YIN-HMM tracking system based on the cumulative mean normalized difference function. We observed that YIN-HMM improves VE for SNRs in the range 0 to 10 dB, but actually harms performance in the high SNR region (20 and 25 dB). Contrary to intuition, the best VE of the YIN-HMM system occurs for the relatively noisy 5 dB SNR condition. We found that this phenomenon was caused by two sources. First was the radioband filtering. After the radioband filtering, YIN detected spurious high pitches, especially in the non-pitched portions of the audio, since the sharp high-pass edge of the filter results in pitch-like features at a multiple of the true f0 in the full-band autocorrelation used by YIN.

The second cause was the HMM tracking. The HMM connects the pitch trace in voiced regions with spurious pitch periods detected in the adjacent non-pitched portions of the audio. In particular, the HMM pitch track would settle on rather high pitch estimates in the unvoiced regions, making it more likely to get stuck on higher harmonics in the subsequent pitched region. However, at lower SNRs, the pitch track in unvoiced regions was more uniformly spread, and this bias was eliminated.

3.4 Summary

In this chapter, we presented a noise robust pitch tracking system based on based on subband selection by classification (SubSel). The proposed SubSel algorithm incorporates the neighboring subband information by forming adjacent-subbands features from k-dim PCA of subband autocorrelations. For the performance evaluation, we reported VE as the performance metric on the KEELE and FDA corpora that provide ground truth pitch labels. To simulate the target noise conditions, a radioband filter was learned from real recorded samples and used in combination with additive pink noise. The subband selection was performed by a set of linear classifiers trained on subband mask labels. Two labeling methods were proposed: (1) Wu criterion and (2)

the ground truth pitch mask with increasing tolerance proportional to the subband center frequency.

In the experiment on the KEELE and FDA corpora corrupted by the radioband filtering plus additive pink noise, the two subband algorithms – the Wu algorithm and the proposed variant – produce relatively robust pitch estimations compared to YIN, a full-band (single-band) algorithm.

The classification approach naturally extends to VAD, where the ground-truth labels are now derived from labeled voicing. Since we observed that most of the pitch tracking errors arise from tracking the irrelevant periodicity in the non-pitch regions, a joint solution of pitch tracking combined with the VAD was expected to improve results, and led to the revised evaluation described in the next chapter.

The SubSel algorithm modifies the way that subbands are selected, but otherwise retains the Wu algorithm's approach of estimating pitch likelihood by picking autocorrelation peaks in the selected channels and combining them across frequency. Instead, the pitch probability can be directly estimated by classifiers operating on local subband information, eliminating the ad-hoc summary autocorrelation stage. This idea led the topic of the chapter 4.

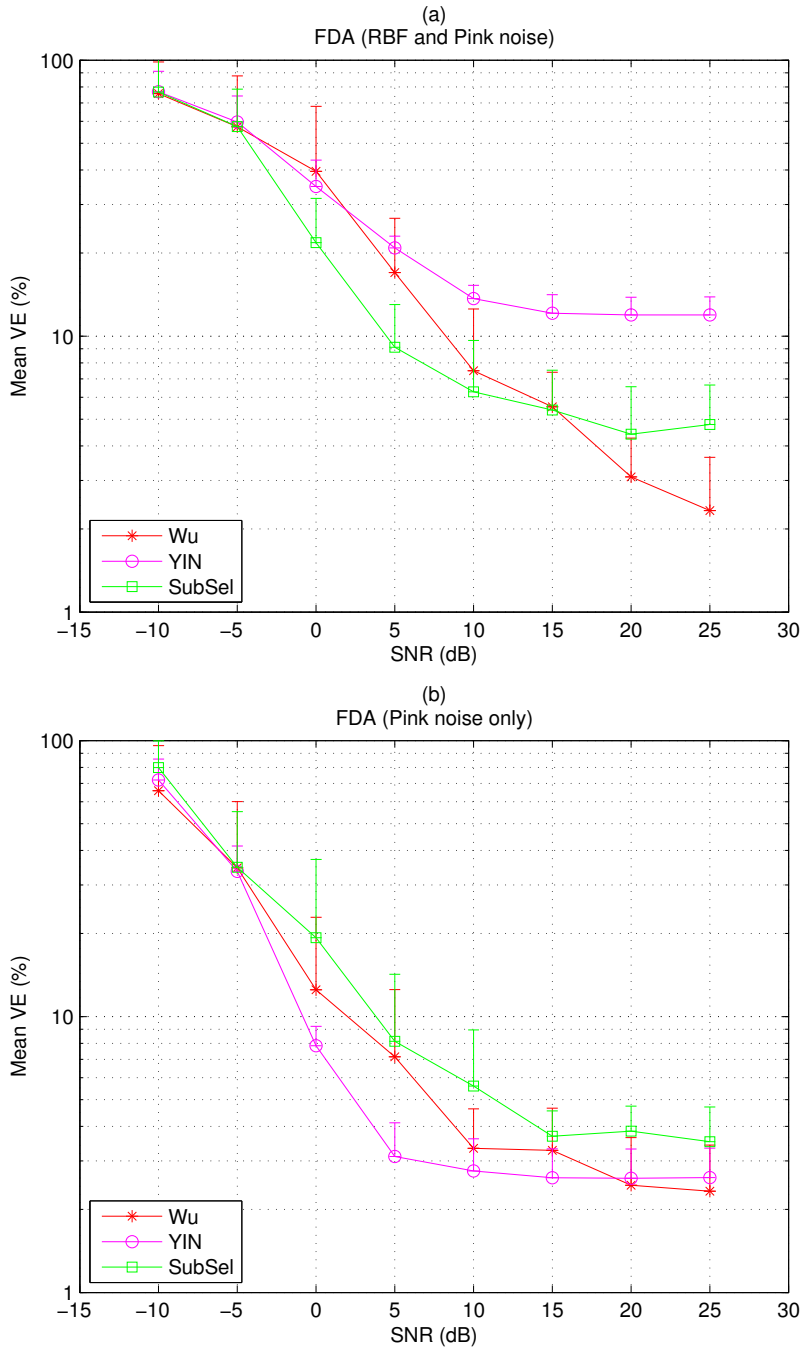


Figure 3.10: The mean VE in log-scale vs SNR of Wu, YIN, SubSel (GTPvarAC5) algorithms on FDA corpus under (a) RBF and pink noise and (b) Pink noise conditions.

Chapter 4

Subband Autocorrelation Classification (SAcC) Pitch Tracker

In this chapter, we extend a pitch tracking system based on the autocorrelation of multiple subbands coming out of an auditory filterbank. However, rather than attempting to explicitly detect the peaks that indicate particular pitches, we train a classifier on the full autocorrelation pattern corresponding to a corpus of labeled training examples. Since these training examples can be processed to include noise and channel characteristics specific to particular conditions, it can be made much more accurate in difficult conditions than "generic" pitch tracking. We also propose a new metric that gives a balanced evaluation of both pitch estimation accuracy and voicing detection. The contents of this chapter is based on [Lee and Ellis, 2012]. An implementation of the SAcC algorithm, with configurations trained on several different conditions, is available both as source and as a Matlab compiled binary[1].

The proposed pitch tracking system is described in section 4.1. The proposed performance metric for pitch tracking is described in section 4.2. The experimental setup

[1] http://labrosa.ee.columbia.edu/projects/SAcC/

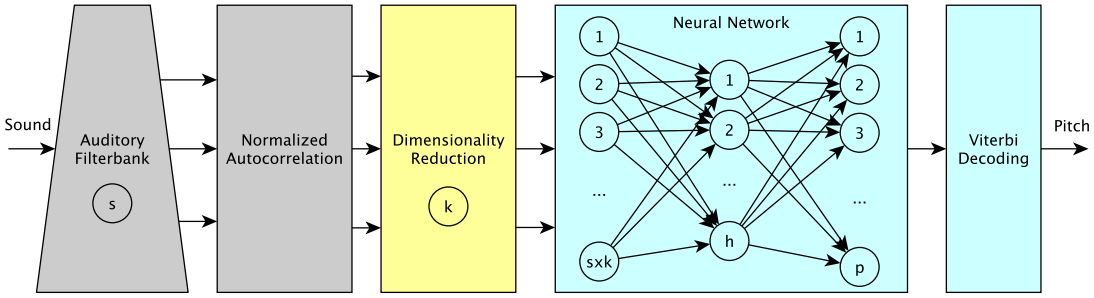


Figure 4.1: The diagram of the proposed Subband Autocorrelation Classification (SAcC) pitch tracking system.

and the results are described in section 4.3. Section 4.4 makes various observations. Section 4.5 provides a summary.

4.1 The SAcC Pitch Tracker

The diagram of the proposed Subband Autocorrelation Classification (SAcC) pitch tracking system is shown in Figure 4.1. The key change from the Wu algorithm is that the pitch period probability is calculated by a single classifier working on the autocorrelations from all subbands, rather than explicit peak picking and cross-band integration. The modified stages are now described in more detail:

4.1.1 Subband PCA Dimensionality Reduction

To avoid overfitting and reduce complexity, the autocorrelation is compressed by k-dim PCA dimensionality reduction as discussed in section 3.1.2. Each subband autocorrelation $A_l(t, \cdot)$ is 400 points long; combining these across the $s = 48$ subbands would give an extremely large feature space. In fact, the normalized autocorrelation of each band-pass filtered signal $x_l[n]$ is highly constrained, leading to large redundancy. To simplify the classification problem, we reduce the dimensionality within each subband by applying PCA.

The principal components corresponding to the k largest eigenvalues were used to produce the subband k-dim PCA features $F_l(t, m)$ for each subband where $l = 1, \cdots, s$ is the subband index, and $m = 1, \cdots, k$ is the principal component index. We tried values for k in the range 5 to 20. The sorted eigenvalues of the PCA components decreased very fast, reflecting the redundancy in the autocorrelations.

4.1.2 MLP Classifier

As shown in Figure 4.1, the SAcC system uses a Multi-Layer Perceptron (MLP) neural network classifier to predict the posterior probabilities across a set of discrete, log-spaced pitch candidates from the subband PCA features. The MLP is trained using QuickNet[2]. The number of inputs to the MLP is $s \times k$. We used a single hidden layer with h hidden units, where h was varied between 50 and 1600.

The MLP had separate outputs for different pitch (period) values over a range which quantized 60 to 404 Hz using 24 bins per octave (in a logarithmic scale), a total of 67 bins. Each ground-truth pitch value in the training data was mapped to the nearest quantized pitch target. Any pitches outside this range were mapped to special "too low" and "too high" bins. Finally, an additional "no-pitch" target output accounted for unvoiced frames, giving $p = 70$ output units in total. To increase the range and volume of training data, each soundfile example was resampled at 8 rates from 0.6 to 1.6 and added to the training pool with a correspondingly-shifted ground truth pitch label.

The output of the MLP gives the observation probability $P(\tau_t|O_t)$ of a pitch candidate τ_t given input observations O_t. Dividing by the pitch prior $P(\tau)$ gives a value proportional to $P(O_t|\tau)$. To smooth the temporal progression of pitch dynamics, the transition probability of the ground truth pitch is modeled as $P(\tau_t|\tau_{t-1})$ empirically. The Viterbi path through a HMM is used to smooth the pitch track, and to differentiate no-pitch and one-pitch states. More specifically, the HMM finds the pitch

[2] http://www.icsi.berkeley.edu/Speech/qn.html

sequence that maximizes the likelihood of the observations O_t by optimizing the sum across time of

$$P(O_t|\tau_t, \tau_{t-1}) = P(O_t|\tau_t)P(\tau_t|\tau_{t-1}) \tag{4.1}$$

where τ_t and τ_{t-1} are the pitches at frames t and $t-1$. The transition probabilities are set parametrically (pitch-invariant) and tuned empirically.

4.2 Performance Metrics

The standard error measures for pitch tracking are Gross Pitch Error (GPE) and Voicing Decision Error (VDE) [Chu and Alwan, 2009]:

$$\text{GPE} = \frac{E_{f_0}}{N_{vv}} \qquad \text{VDE} = \frac{E_{v\to u} + E_{u\to v}}{N} \tag{4.2}$$

where N is the total number of frames, N_{vv} is the count of frames in which both the pitch tracker and the ground truth reported a pitch, E_{f_0} counts the frames in which these pitches differ by some factor (typically 20%), $E_{v\to u}$ is the count of voiced frames misclassified as unvoiced, and $E_{u\to v}$ is the number of misclassified unvoiced frames. The problem with this measure is that GPE can be improved by labeling voiced frames whose period is ambiguous as unvoiced, thereby reducing the N_{vv} denominator. This will increase VDE, but it is difficult to compare overall performance with this pair of numbers, and the temptation is to optimize GPE as a primary objective.

We therefore propose a modified metric to evaluate pitch trackers which we call the Pitch Tracking Error (PTE). It is a simple average of VE and Unvoiced Error (UE):

$$\text{PTE} = \frac{\text{VE} + \text{UE}}{2} \tag{4.3}$$

The Voiced Error (VE) and the Unvoiced Error (UE) are given as:

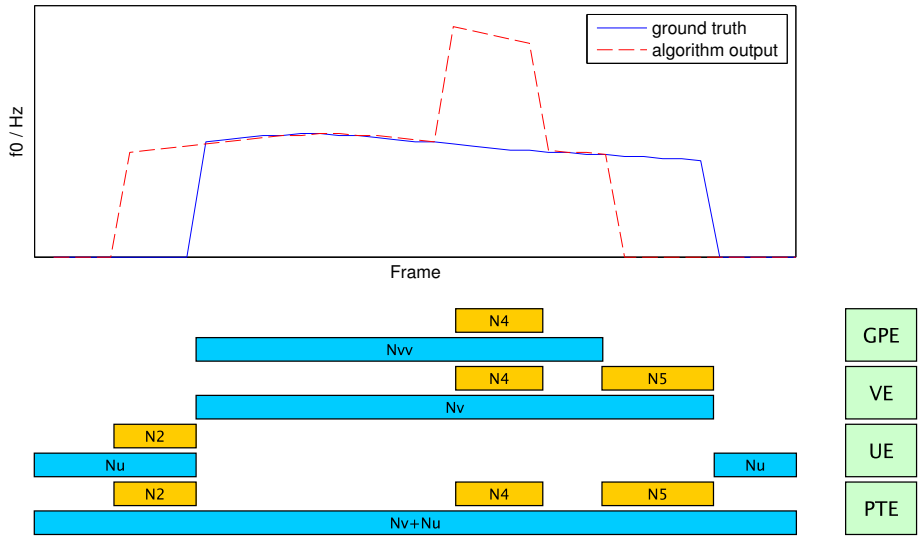


Figure 4.2: An illustration of pitch tracking and the corresponding GPE, VE, UE and PTE.

$$\text{VE} = \frac{E_{f_0} + E_{v \to u}}{N_v} \qquad \text{UE} = \frac{E_{u \to v}}{N_u} \tag{4.4}$$

where N_v is the number of frames for which a pitch is reported in the ground truth, and $N_u = N - N_v$ is the remaining (unvoiced) frame count.

An illustration of pitch tracking and the corresponding error metrics are shown in Figure 4.2. For each error metric, the numerator portion is in yellow and the denominator portion is in blue. The denominator portion of GPE can vary according to the pitch tracker outputs. In contrast, the denominators of VE, UE, and PTE are fixed by the ground truth. On the denominator side, PTE takes all frames into account. On the numerator side, only PTE covers all types pitch tracking errors, namely $E_{u \to v} = \text{N2}$, $E_{f_0} = \text{N4}$, and $E_{v \to u} = \text{N6}$.

It is more transparent to compare VE, UE, and PTE between different pitch trackers because the denominators N_v and N_u do not vary depending on the system output. If we consider pitch tracking as detection, VE resembles miss rate, and UE is similar to false alarm rate. Another advantage of PTE is that it can balance the

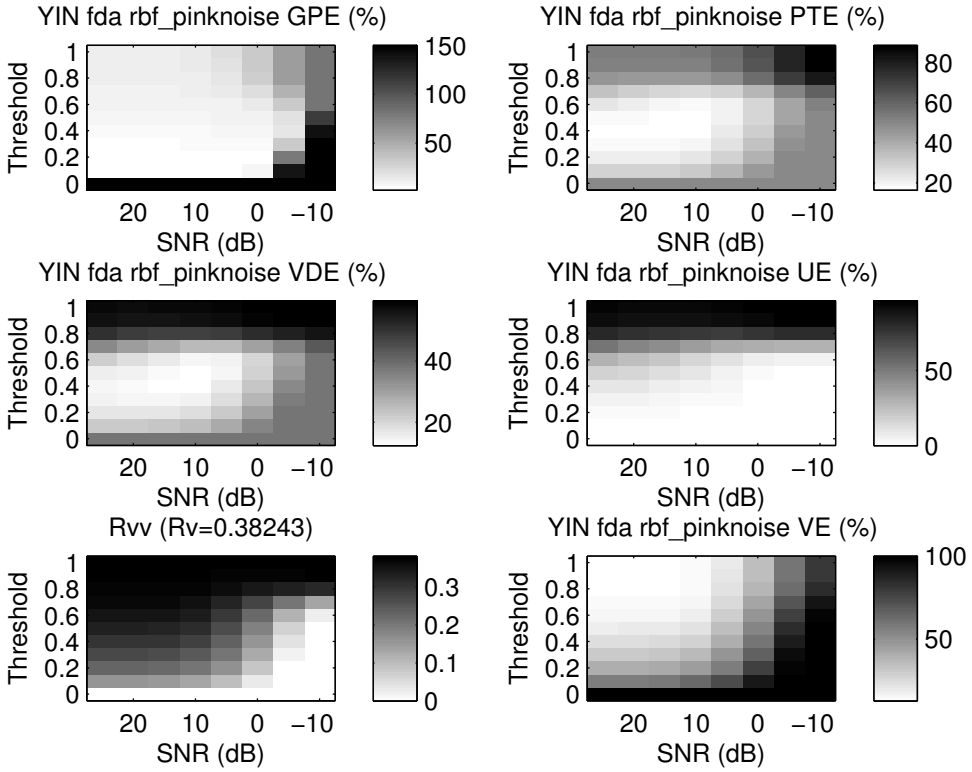


Figure 4.3: The GPE, PTE, VDE, UE, R_{vv}, and VE for YIN at various threshold and SNR points on FDA under RBF and pink noise condition.

contribution of errors on voiced and unvoiced frames regardless of their proportion in the actual evaluation material, making the results more comparable between different test sets.

Different weights for VE and UE could be considered for tasks where one kind of error was more important. In this case, PTE can more generally be defined:

$$\mathrm{PTE}_\gamma = \gamma \mathrm{VE} + (1-\gamma)\,\mathrm{UE} \tag{4.5}$$

where $\gamma \in [0, 1]$ is a weight of voiced error.

If there is any need to emphasize VE or UE in reporting or measuring error, PTE_γ can be used. In this thesis, however, $\gamma = 0.5$ is always used to fairly account for VE and UE of pitch trackers.

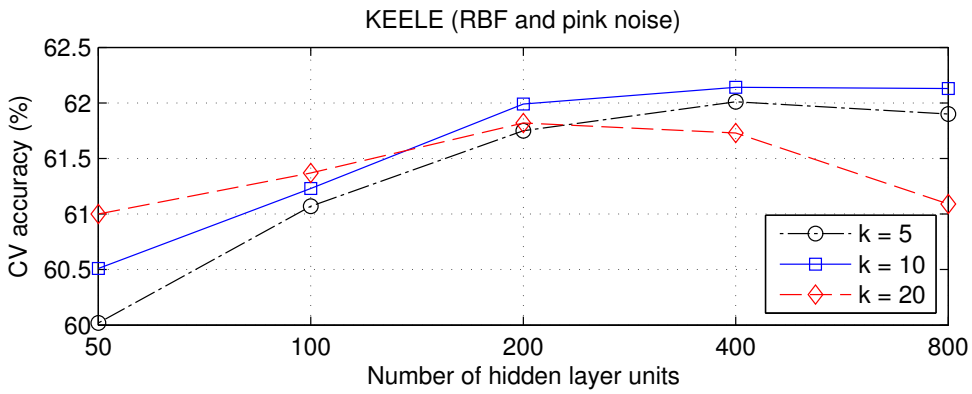


Figure 4.4: The Cross Validation (CV) accuracy of the MLP using the k-dimensional PCA feature.

4.3 Experiments

4.3.1 Data

We used the KEELE [Plante *et al.*, 1995] and FDA [Bagshaw *et al.*, 1993] corpora for evaluation. The lengths of the datasets are 337 s and 332 s respectively. KEELE consists of 10 speakers each reading the same story for about 30 s; FDA has two speakers reading the same 50 short sentences of around 3 s each. Since KEELE includes greater variation, and to illustrate generalization, we chose to train on KEELE and report results on FDA. Since our interest is in pitch tracking that can be used on low-quality radio transmissions, our main experiment applied to both training and test material a simulated radio-band filter (RBF) modeled from a real recording made across a narrow-FM channel[3] (shown in Figure 3.7), to a bandpass spanning around 500 Hz to 2 kHz, along with additive pink noise at various levels.

[3] http://labrosa.ee.columbia.edu/projects/renoiser/

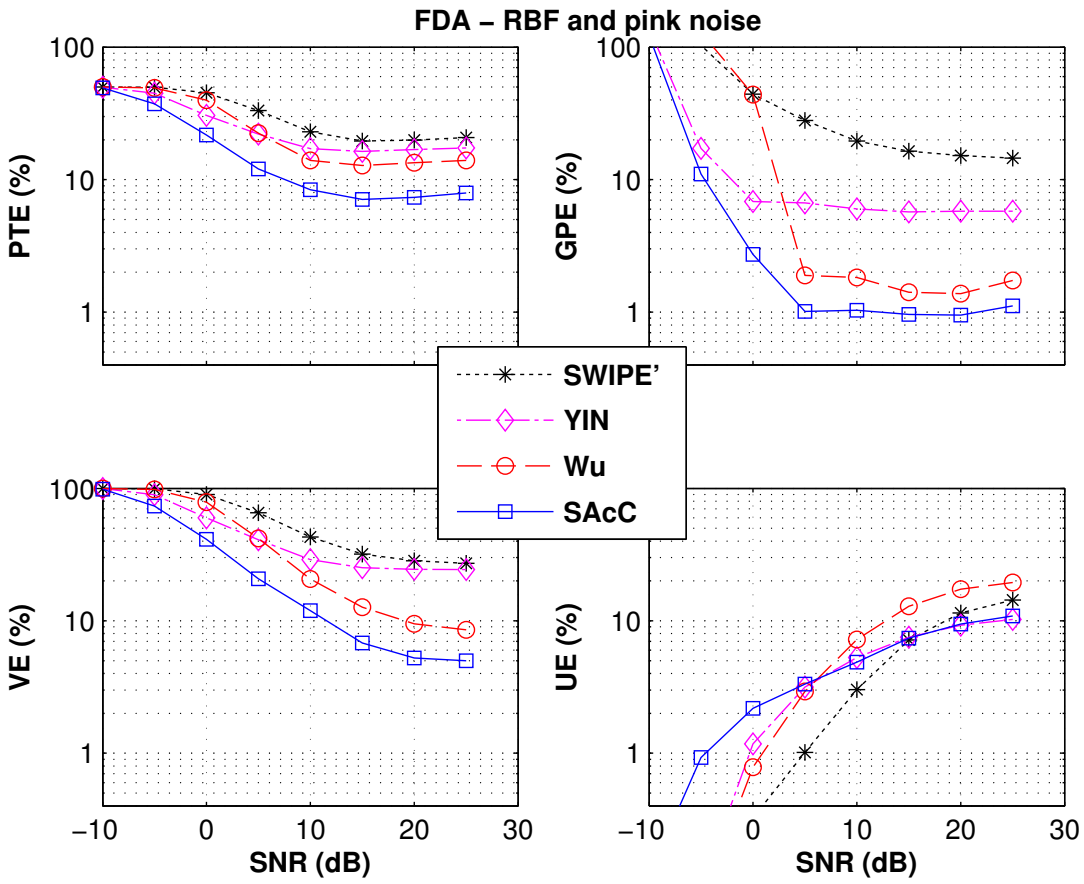


Figure 4.5: The PTE, GPE, VE, and UE for SAcC, Wu, YIN, and SWIPE′ on FDA under RBF and pink noise condition.

4.3.2 Experiment Setup

YIN [de Cheveigne and Kawahara, 2002], Wu [Wu *et al.*, 2003], and SWIPE′ [Camacho and Harris, 2008] algorithms are used for performance comparison. Both the ground truth and the pitch trackers gave pitch values for every 10 ms.

To use YIN and SWIPE′ as pitch trackers, the pitch strength outputs (aperiodicity for YIN and pitch strength for SWIPE′) are thresholded to provide voiced/unvoiced decisions. Figure 4.3 shows the GPE, PTE, VDE, UE, R_{vv}, and VE of YIN versus SNR for various thresholds for speech with RBF and pink noise, where $R_{vv} = N_{vv}/N$ and $R_v = N_v/N$. The threshold giving the best PTE was used in evaluation. For the

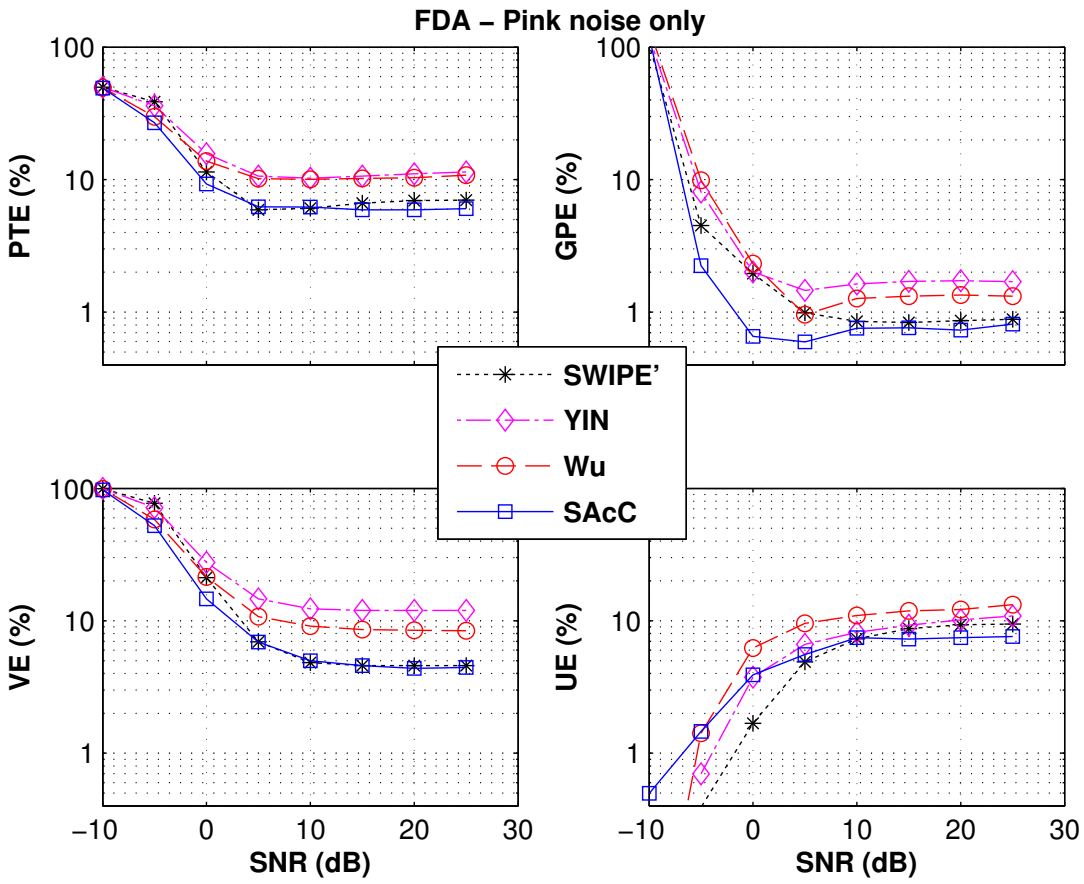


Figure 4.6: The PTE, GPE, VE, and UE for SAcC, Wu, YIN, and SWIPE′ on FDA under pink noise condition.

Wu algorithm, the probability of no-pitch is searched over the 1^{st} to 90^{th} percentiles of the remaining pitch likelihoods to find the value that optimized PTE.

For the SAcC MLP, 66.7% of the data was used for training, with the rest used for cross validation (CV). Figure 4.4 shows the CV accuracy as a function of k, the number of principal components retained, and h, the hidden layer size on the most challenging RBF case. From these results, we chose $k = 10$ and $h = 800$.

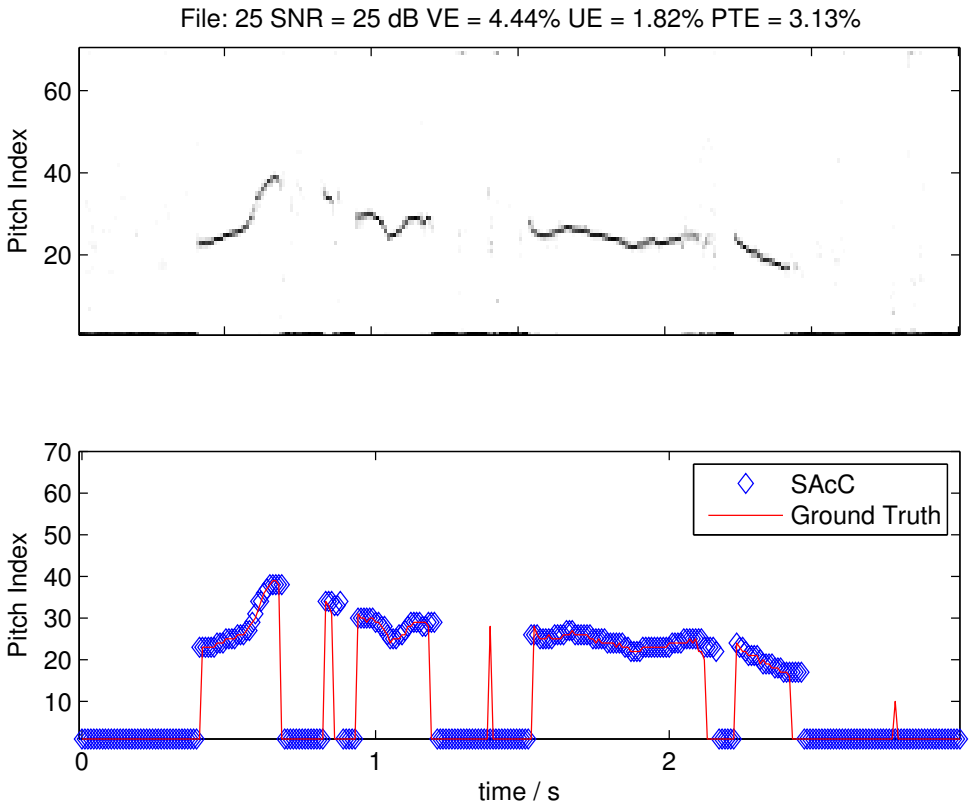


Figure 4.7: The MLP outputs $P(\tau_t|O_t)$ (top panel); and Viterbi tracking output of SAcC (blue diamond) and the ground truth (red line) on a sample speech corrupted with RBF and pink noise at 25dB SNR. (bottom panel)

4.3.3 Results

Looking at the right column of Figure 4.3, we see that lowering the threshold and thus reducing the proportion of voiced frames lowers UE (as all frames, including the unvoiced ones, are labeled unvoiced) while increasing VE. As the sum of these competing trends, PTE shows a clear optimum for a threshold around 0.4. In the left column, GPE appears to improve as the threshold decreases, but this hides the disappearing proportion of frames, N_{vv} (bottom pane), over which this measure is calculated. When $N_{vv} = 0$, an arbitrary high value (150%) is assigned to GPE to reflect that it is based on zero frames. Note that optimizing GPE at higher SNRs

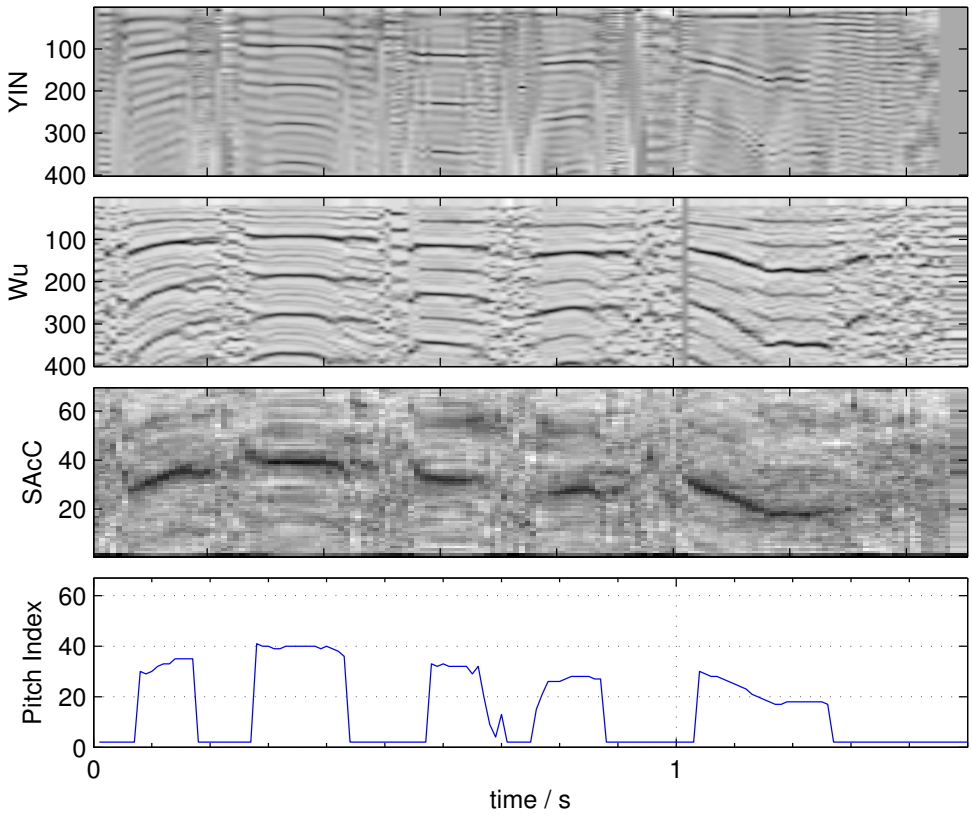


Figure 4.8: The observation pitch likelihood of YIN, Wu, and SAcC on a speech sample corrupted with RBF and pink noise at 25dB SNR. Note that the grayscale for SAcC likelihood is log-scaled to reveal more detail at very small probabilities.

would give a threshold closer to 0.2, quite different from the optimum for PTE. VDE reveals an optimal threshold similar to PTE, but ignores actual pitch estimation errors.

The performance comparisons of SAcC, YIN, Wu, and SWIPE′ on FDA dataset under the RBF plus pink noise condition and the pink noise condition are shown in Figure 4.5 and 4.6. SAcC is shown to outperform all the other algorithms by a substantial margin for all positive SNRs. To be fair, this should be interpreted in light of the fact that SAcC has been trained specifically for these conditions, whereas the other algorithms attempt to address any possible condition.

For SAcC, PTE is dominated by UE in the high SNR and VE in the low SNR.

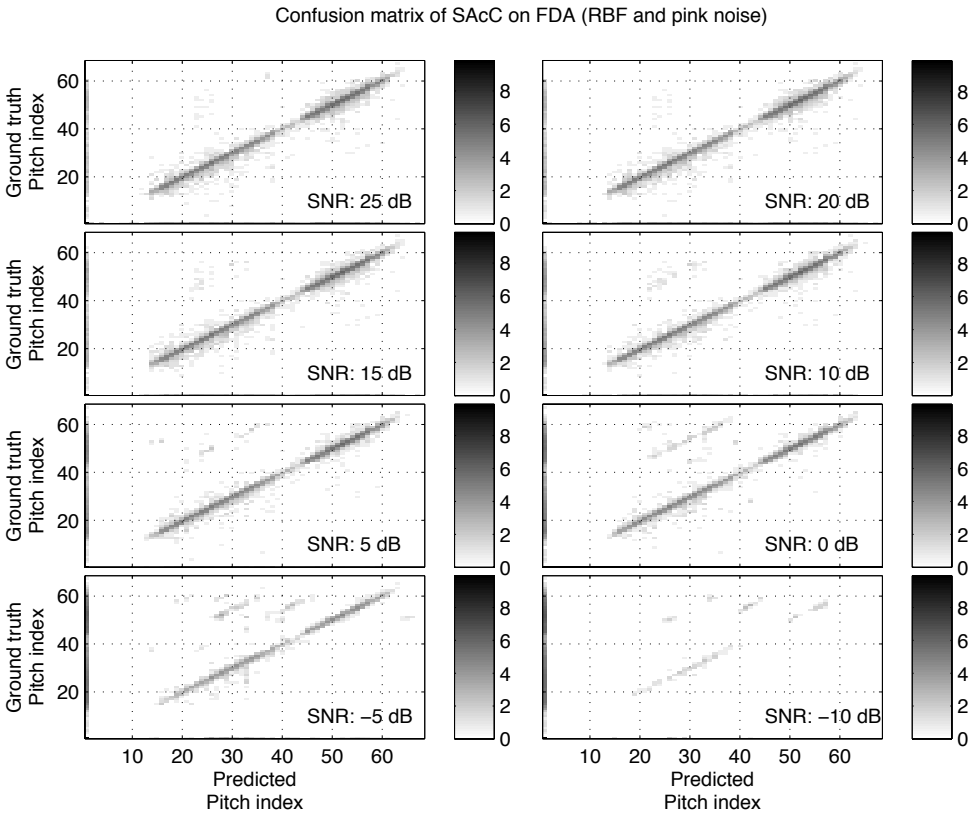


Figure 4.9: The log-scale confusion matrix of SAcC on FDA corrupted with RBF and pink noise.

In low SNRs, pitch tracker outputs are mostly no-pitch, lowering UE and increasing VE. Note that PTE gives higher absolute values than GPE since it reflects both difficult voiced frames and voicing errors; we consider performance in these areas to be critical. Also note that PTE asymptotes at 50% for high-noise conditions, since typically systems will label all frames as unvoiced in this case, making UE = 0% and VE = 100%.

4.4 Discussion

The output of the SAcC MLP $P(\tau|O_t)$ on a sample speech is shown on the top pane of Figure 4.7. The most likely pitch candidate for each frame has a significantly stronger

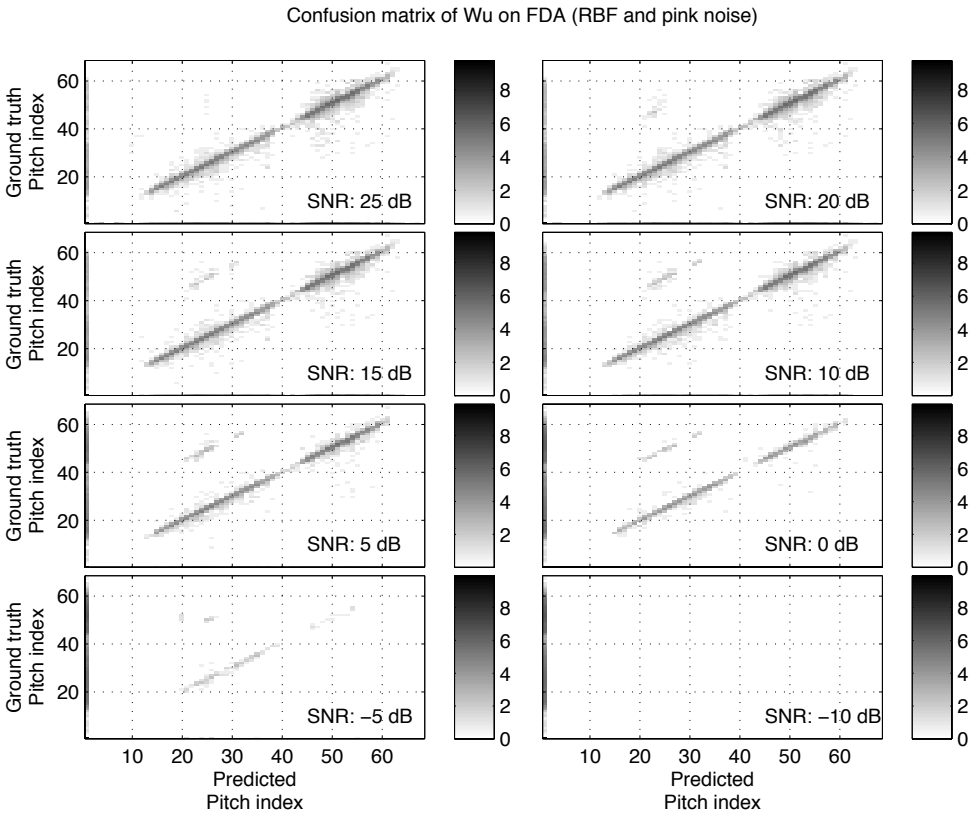


Figure 4.10: The log-scale confusion matrix of the Wu system on FDA corrupted with RBF and pink noise.

value than the others. The HMM tracking result of SAcC on the same example is shown on the bottom panel in Figure 4.7 along with the ground truth pitch. The HMM tracking promotes continuous pitch tracks and discourages voicing transitions, which sometimes causes the extension of pitch tracks into unvoiced regions.

The observed pitch likelihood of YIN, Wu, and SAcC on another speech sample corrupted with RBF and pink noise at 25dB SNR is shown in Figure 4.8. The vertical axis is lag in samples (increasing downwards) for YIN and Wu, but quantized (log-frequency) pitch for SAcC. For SAcC, the log of the MLP output, $\log(P(\tau|O_t))$, is shown to reveal details in the non-favorite candidates. Both YIN and Wu are based on autocorrelation operations, and have harmonic and subharmonic structures. Due to the absent fundamental and low harmonics, YIN is confused with more pro-

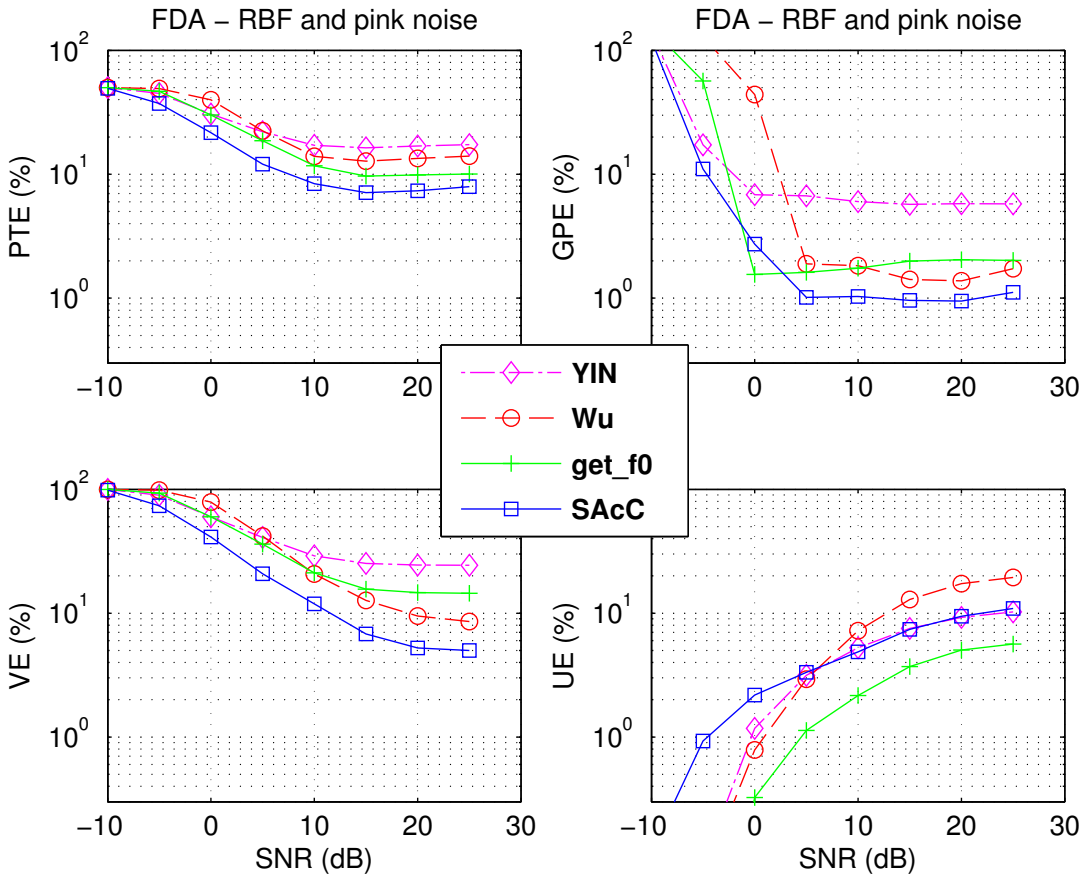


Figure 4.11: The PTE, GPE, VE, and UE for YIN, Wu, `get_f0` and SAcC on FDA under RBF and pink noise condition.

nounced sub-period errors. For the Wu system, the multiple-period errors are more pronounced.

Since SAcC is trained to discriminate between these ambiguous cases with harmonic relationships, it has one strong peak in most frames, reducing the confusion of octave errors. Figure 4.9 and 4.10 show the log-scale confusion matrix figures on the FDA corpus under the RBF and pink noise condition. The counts in the confusion matrix is in log-scale to make the small differences in off-diagonal visible. As a result of discriminative pitch classification of SAcC, it makes less octave/sub-octave errors than the Wu system.

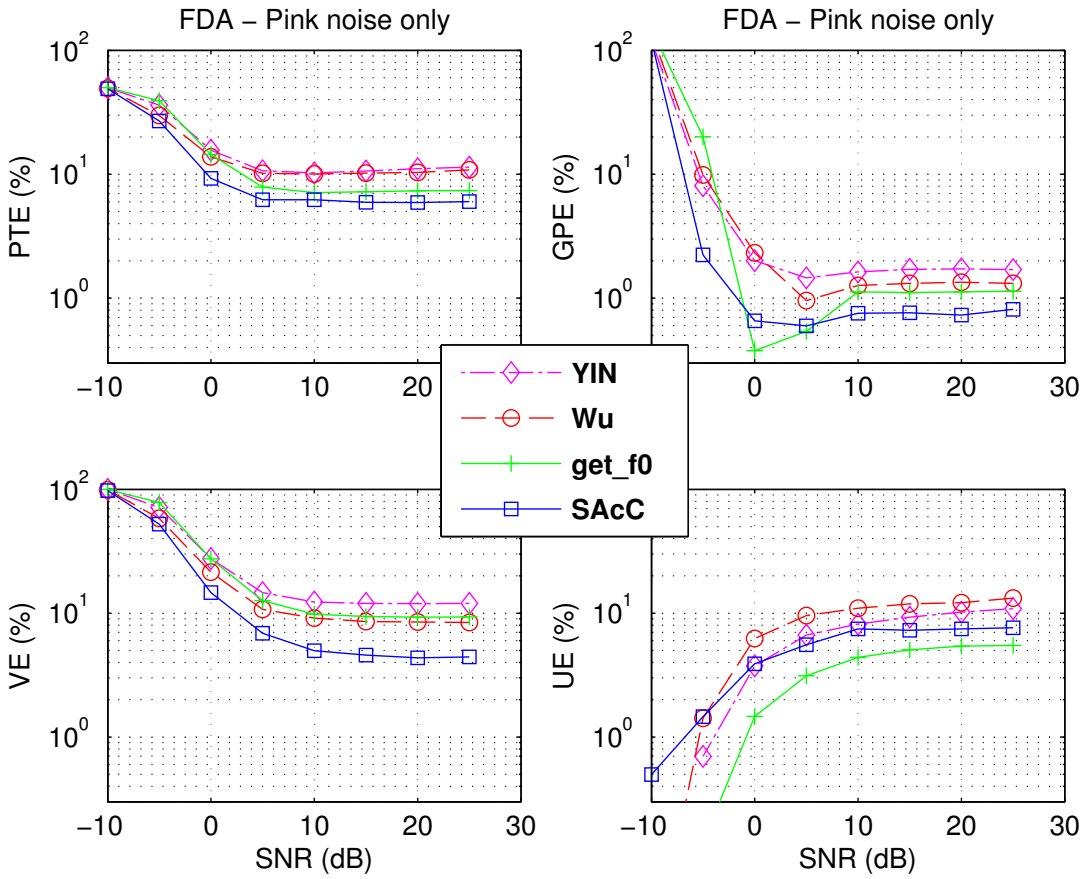


Figure 4.12: The PTE, GPE, VE, and UE for YIN, Wu, `get_f0` and SAcC on FDA under pink noise condition.

For unvoiced likelihood, YIN uses the aperiodicity output which can jump between voiced to unvoiced. The Wu algorithm assigns indirect measure inferred from the summary autocorrelation which performs badly in noisy conditions. In contrast, SAcC trains a classifier for no-pitch discrimination for all noise conditions and shows better performance.

Figure 4.11 and 4.12 show the performance comparisons including the popular `get_f0` in replace of SWIPE′ . The performance advantage of the Wu system in GPE comes at the cost of UE. For pitch tracking, UE is also an important performance requirement, and cannot be sacrificed to improve another, GPE As we saw in this

case. It is dangerous to optimize in terms of GPE only.

The performance improvement of SAcC is due to two major sources. First, SAcC is trained specific to the target noise conditions. We observed that SAcC generalizes well, but there exists performance degradation when SAcC is tested on unmatched noise conditions. Second, the trained classifier of SAcC can discriminate octave/sub-octave error cases better than other pitch trackers.

4.5 Summary

We have proposed a noise robust pitch tracking system, SAcC, based on subband autocorrelation classification. The proposed algorithm incorporates the learning power of a MLP classifier, the smooth tracking of a HMM, and the low dimensional representation of the k-dimensional subband PCA feature. We have also proposed a performance metric, PTE, to give a balanced measure of performance in both voiced and unvoiced regions.

To simulate the target noise condition, a radioband filter was learned from real recorded samples and used in combination with additive pink noise to make a useful simulation of poor quality radio reception, the particular focus of our study. We believe, however, that the subband classification structure should be advantageous in many challenging acoustic conditions, particularly when matched training data is available. We will discuss this issue in the next chapter.

The performance evaluation on KEELE and FDA corpora showed that SAcC improves the state-of-the-art for pitch tracking on this kind of data, as measured both by the conventional GPE metric and by our PTE metric. Because our proposed algorithm involves a simple, trained classification stage, it can be optimized for particular speech conditions and datasets.

Chapter 5

Application of Subband Autocorrelation Classification (SAcC)

In this chapter, we present a set of experiments that demonstrate how SAcC generalizes for a large dataset of various noise conditions.

Section 5.1 introduces the RATS dataset. Section 5.2 explains the experimental setup. Section 5.3 discusses the experiment results. Section 5.4 discusses the application of SAcC for the RATS Speaker Identification (SID) task. Section 5.5 discusses the application of SAcC for relatively low-noise telephone speech (as encountered in the Babel program). Section 5.6 provides a summary.

5.1 Dataset

To verify the generalization performance of SAcC pitch tracker, SAcC was tested on real-world problems with a large-scale speech recording dataset.

The Robust Automatic Transcription of Speech (RATS) is a Defense Advanced Research Projects Agency (DARPA) research program focusing on extracting in-

formation from highly distorted audio signals that have been transmitted across a low-quality radio channel, in a variety of languages.

There are four objectives in RATS program: (1) Speech Activity Detection (SAD): The system need to able to determine whether whether or not the audio contains speech. (2) Language Identification (LID): For the speech portion, the system can identify the language spoken. (3) Speaker Identification (SID): For the speech portion, the system can determine whether the speaker is one of the wanted speakers in the list. (4) Key Word Spotting (KWS): For the speech portion, the system can identify specific words from the list of keywords for the corresponding language.

Pitch tracking is directly directly relevant to SAD and potentially helpful in SID. Although SID is a separate topic, our collaborator at SRI International observed a performance improvement in SID using SAcC pitch and voicing probability features trained on a subset of RATS corpus. We describe this in section 5.4.

The challenges of the RATS program include (i) wireless transmitted (bandlimited) audio, (ii) frequency shifted audio (resulting from single-side band (SSB) transmission), and (iii) high additive noise and distortion.

The SAD training material for RATS, released by the Linguistic Data Consortium (LDC)[1], contains 71.5 hours of clean source speech audio and 572 hours of the radio-band transmitted audios over eight distinct actual radio channels (A-H). Table 5.1 shows the length of audio in the RATS corpus.

To compare the lengths of corpora, the lengths of KEELE [Plante *et al.*, 1995] and FDA [Bagshaw *et al.*, 1993] are 5.5 mins long each; the RATS corpus is about 7000 times longer than KEELE or FDA. Even the source (pre-transmission) audio of RATS is about 780 times longer than KEELE or FDA. The RATS corpus contains 5 language conditions.

[1] http://www.ldc.upenn.edu/

Languages	Source channel	A-H channels	Total
fsh-alv	26.0	208.0	234.0
fsh-eng	34.5	276.0	310.5
rats-cts-alv	4.0	32.0	36.0
rats-cts-pus	3.5	28.0	31.5
rats-cts-urd	3.5	28.0	31.5
Total	71.5	572.0	643.5

Table 5.1: The length in hours of the RATS dataset. Note that lengths for all channels will generally be slightly longer due to channel transmission procedures.

5.2 Experiment

The design of the experiments are described in this section. The successful application of SAcC on RATS corpus will demonstrate generalization performance of SAcC across the different noise conditions and different datasets.

We used a subset of the RATS data, which is large enough to show the generalization performance of SAcC. The training subset of the RATS corpus consists of 180 audio files with total length of 37.3 hours, which breaks down as 20 source (pre-transmission) files (4.1 hours), and the 160 radio transmitted audio files (channel A-H). The 20 source audio files of the training subset include 8 `fsh-alv`, 6 `fsh-eng`, 2 `rats-cts-alv`, 2 `rats-cts-pus`, and 2 `rats-cts-urd` to proportionally represent the language distribution of the RATS corpus.

A separate testing subset of RATS corpus consists of 117 audio files with total length of 25.6 hours, 13 distinct source files (2.8-hour long) and corresponding 104 channel transmitted audio files. The 13 source audio files of the test subset include 4 `fsh-alv`, 3 `fsh-eng`, 2 `rats-cts-alv`, 2 `rats-cts-pus`, and 2 `rats-cts-urd`. Since we are interested in pitch tracking for noisy speech, we measure the performance

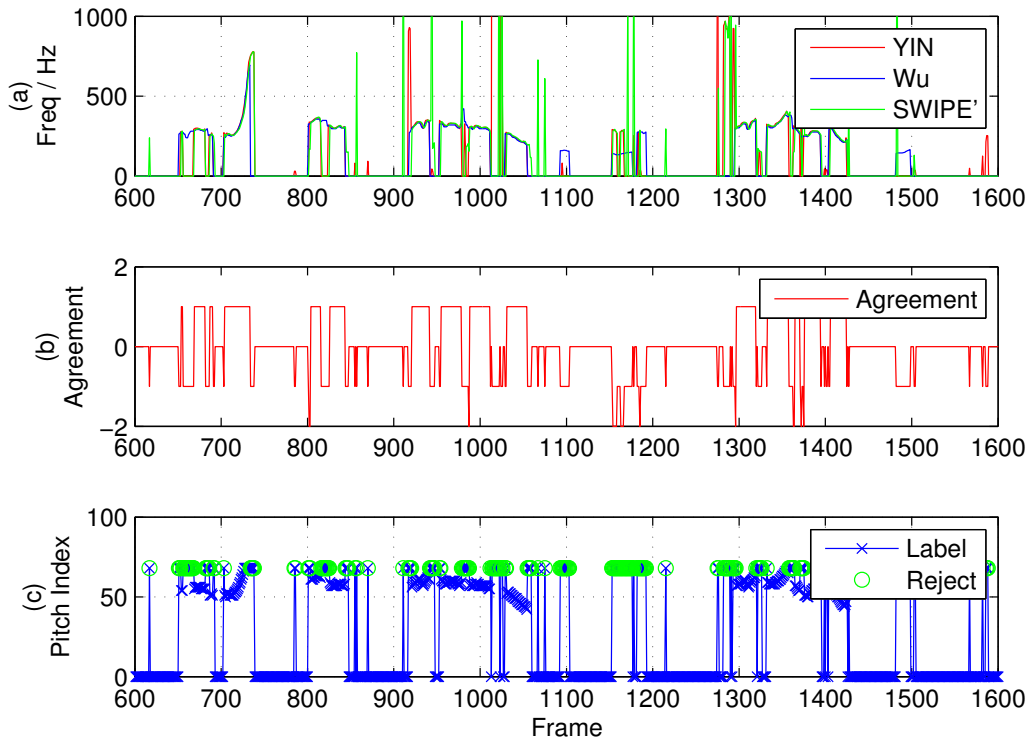


Figure 5.1: An example label generated by agreement of YIN, Wu, and SWIPE′ on RATS corpus.

on the 8 transmitted channels. The source audio is used for the ground truth label generation as explained below.

The RATS corpus contains the audio without frame-wise pitch annotation. Since no pitch label is available, we generated labels by applying three separate pitch trackers, YIN, Wu, and SWIPE′ , to the high-quality source audio using `genPitchLabel`[2]. To use YIN and SWIPE′ for pitch trackers, the aperiodicity of YIN and the pitch strength of SWIPE′ that give the best agreement with the Wu SAD are used to threshold SAD for the corresponding pitch estimators. The three pitch trackers agreed on about 67% of the source frames; the remaining "reject" frames were not used in training. For the test, these "reject" frames were labeled as unvoiced frames. As a result,

[2] http://www.ee.columbia.edu/~bsl/projects/genPitchLabel/

Name	Training condition
$\text{SAcC}_{\text{All CH}}$	RATS all channels (source and channels A-H)
SAcC_{CH}	RATS source and target channel only
SAcC_{nCH}	RATS all channels except target channel
$\text{SAcC}_{\text{ABCEGH}}$	RATS all channels except channels D and F
$\text{SAcC}_{\text{Keele}}$	Keele corpus (RBF and pink noise)

Table 5.2: The SAcCs trained with various conditions.

the generated pitch label is biased toward mark the disagreed frames as unvoiced frames.

An example of the labels generated by the agreement of YIN, Wu, and SWIPE′ using `genPitchLabel` is shown in Figure 5.1. Figure 5.1 (a) shows the individual pitch tracking results of YIN, Wu, and SWIPE′ algorithms. Figure 5.1 (b) shows the SAD decision based on the following criteria. The SAD is 1 when the (voiced) pitches of three algorithms agree, 0 when three algorithms agree on unvoiced, -1 when three algorithms disagree on voiced, and -2 when the (voiced) pitches disagree. Figure 5.1 (c) shows the final pitch label with reject values. Since the negative SAD decision implies the instability of the pitch label, we excluded these reject frames in training.

The labeling generated from the source audio files were used to produce the labels for the channel transmitted audio files. The timing skew between the source and channel files are measured by the `skewview` tool[3] and corrected in the propagated labels.

We trained the SAcC on the five training conditions listed in Table 5.2. Four of them are trained on RATS corpus; and one is trained on Keele corpus.

Among the RATS-trained SAcCs, we expect $\text{SAcC}_{\text{All CH}}$ will be the most general

[3] http://labrosa.ee.columbia.edu/projects/skewview/

SAcC that will perform well for all conditions. SAcC_{CH} is trained to the specific channels, A-H. SAcC_{nCH} is the opposite, oblivious to the specific channels. Comparing $\text{SAcC}_{\text{All CH}}$ and SAcC_{CH} reveals the impact of building a single classifier for all channels versus specializing on each channel individually; Comparing $\text{SAcC}_{\text{All CH}}$ to SAcC_{nCH} gives us a good idea how well SAcC will generalize to unseen channel conditions.

The $\text{SAcC}_{\text{Keele}}$ is trained on Keele corpus with resampling at 9 rates corrupted with RBF and pink noise at the 8 SNR conditions from Chapter 4. The performance of $\text{SAcC}_{\text{Keele}}$ on RATS corpus is included to evaluate SAcC performance on unseen audio with various unseen distortion conditions. We also include the Wu pitch tracker, which emerged as the best competitor in Figure 4.5, as a benchmark reference.

5.3 Discussion

In this section, we discuss the result of the performance evaluation of SAcC on the RATS corpus and FDA corpus. The pitch tracking results on the testing subset of RATS corpus confirms the applicability of SAcC on this real-world dataset.

The performance of each SAcC will show the following effects. $\text{SAcC}_{\text{All CH}}$ will show (1) how the most general (unspecific) pitch tracker performs on the individual channel conditions; and (2) the relative difficulty of each channel condition. SAcC_{CH} shows how well the most specific pitch tracker (specific to a single channel) performs. SAcC_{nCH} will show how specific the test conditions are for SAcC trained with all other channel conditions.

The distortion profiles of channels D and F are different from the other channels in that they include frequency shifting. To test the hypothesis that learning to accommodate frequency-shifted signals might hurt performance on unshifted data, we also trained a system on all channels excluding D and F, $\text{SAcC}_{\text{ABCEGH}}$. $\text{SAcC}_{\text{ABCEGH}}$ on the channels D and F will show how difficult the pitch tracking is by the pitch

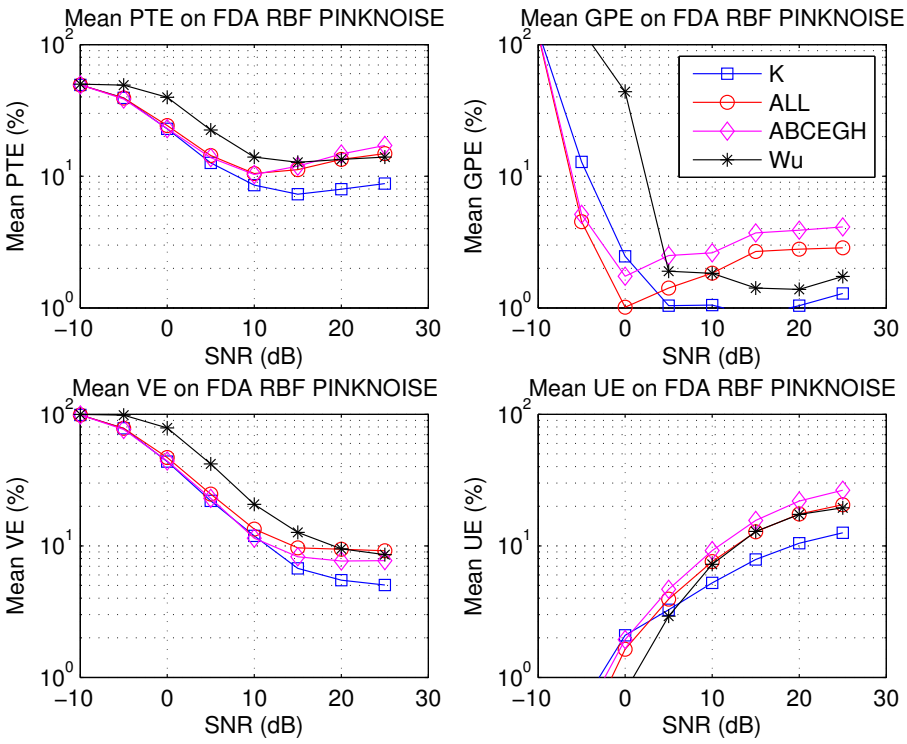


Figure 5.2: The mean PTE, GPE, VE, and UE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, and $\text{SAcC}_{\text{ABCEGH}}$ on FDA dataset.

tracker trained on all data except channels D and F. The performance of $\text{SAcC}_{\text{Keele}}$ will show generalization of both cross-dataset and cross-condition.

The performance of each SAcC on the specific channels of the testing subset of the RATS corpus will show the channel-wise performance of each SAcC. We describe the mean and the sentence-wise PTE, GPE, VE, and UE results on the testing subset of RATS corpus. To examine generalization of RATS-trained SAcC s, we also use FDA corpus.

Figure 5.4 and 5.5 show the mean PTE, GPE, VE, and UE of Wu and the five SAcC pitch tracking systems (trained on the described conditions, namely $\text{SAcC}_{\text{All CH}}$, SAcC_{CH}, SAcC_{nCH}, $\text{SAcC}_{\text{ABCEGH}}$, and $\text{SAcC}_{\text{Keele}}$) on the testing subset of RATS dataset.

As expected, each SAcC_{CH} has the lowest PTE on the corresponding channel

across all channels, demonstrating the specificity of SAcC_{CH}. $\text{SAcC}_{\text{All CH}}$ shows the competitive performance against the specific SAcC_{CH}, demonstrating the generalization of $\text{SAcC}_{\text{All CH}}$. The four pitch trackers other than $\text{SAcC}_{\text{All CH}}$ and SAcC_{CH} perform worse on channels D and F than on the other channels, implying that channels D and F are quite distinct from the others. The Wu system and $\text{SAcC}_{\text{Keele}}$ perform worse than the other SAcC s trained on the RATS dataset, especially in terms of VE. The GPE is relatively lower than PTE, and does not reflect the difficulty of channels such as F as well as PTE. PTE is mostly dominated by VE, except for the Wu algorithm on the channel D. As predicted, $\text{SAcC}_{\text{ABCEGH}}$ performs slightly better than $\text{SAcC}_{\text{All CH}}$ for channels other than D and F, but the margin is very small, indicating that including D and F has not impacted $\text{SAcC}_{\text{All CH}}$ by much.

Figure 5.6 and 5.7 show the sentence-wise PTE, GPE, VE, and UE of Wu and the same five SAcC pitch tracking systems on RATS dataset. The sentence-wise results show how much each performance metric varies for each algorithm on each testing channel. Overall, the sentence-wise variation is most visible when the error is high.

To examine cross-corpus performance of SAcCs trained on RATS corpus, we report results on the FDA corpus. Figure 5.2 shows the mean PTE, GPE, VE, and UE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, and $\text{SAcC}_{\text{ABCEGH}}$ on the FDA corpus corrupted by RBF and pink noise conditions at various SNRs. As a reference, the Wu system and $\text{SAcC}_{\text{Keele}}$, which gives the best performance, were used. The performance of $\text{SAcC}_{\text{All CH}}$, which is the most general among RATS-trained SAcC, is close to that of $\text{SAcC}_{\text{Keele}}$ for lower SNRs ($\leq$ 10 dB) and close to that of the Wu for higher SNRs ($\geq$ 15 dB). In terms of VE, $\text{SAcC}_{\text{All CH}}$ is better than the Wu system. In terms of UE, $\text{SAcC}_{\text{All CH}}$ and the Wu are equivalent. Overall, $\text{SAcC}_{\text{All CH}}$ performs better than the Wu system on the FDA corpus.

	Target True	Target False	Rate
Predict True	Hit (A)	False Alarm (B)	FA = $\frac{B}{A+B}$
Predict False	Miss (C)	Correct Rejection (D)	M = $\frac{C}{C+D}$

Table 5.3: The false alarm (FA) rate and the miss (M) rate

Conditions	`get_f0`		SAcC	
	FA@10m[4]	EER[5]	FA@10m	EER
6Rside-30-30-core-seen	19.82	14.55	16.71	13.23
6Rside-30-30-core-unseen	23.26	16.06	19.07	14.41
6Rside-30-30-core_key	21.47	15.26	17.86	13.86

Table 5.4: The performance of SRI `prospol` SID system using SAcC [L. Ferrer '12]

5.4 Speaker Identification (SID) Task

So far, we have evaluated the performance of SAcC as a pitch tracker. The output of pitch tracking, the pitch value and the voicing probability, can be used as a feature for the Speaker Identification (SID) task. The fundamental frequency has been a major feature for general speech recognition systems [Picone, 1993].

Our collaborators at SRI International[6] conducted a pilot experiment using the SAcC outputs as features for SRI `prospol` Speaker Identification (SID) system in place of their standard features based on the popular `get_f0` software [Talkin, 1995]. The result showed a significant gain in SID performance on the current system at SRI.

Two measures were reported for the SID task. FA@10m measures the false alarm rate at the operating point which gives a miss rate of 10 %. The Equal Error Rate

[6] http://www.sri.com/

(EER) is the error rate when the false alarm and the miss rate are the same. The definitions of false alarm rate and miss rate are given in Table 5.3.

The pilot performance evaluation result of SRI `prospol` SID system on the RATS corpus is summarized in Table 5.4. For various test conditions, both measures, FA@10m and EER, are lower with SAcC features than with `get_f0` features. Considering the large range of other factors involved in SID system performance, the improvement is significant and exciting. These impressive pilot result led to a full run of the SID experiment using the SAcC features.

5.5 Application on the Babel corpus

In this section, we provide another application of SAcC in different kind of large-scale speech corpus.

The Intelligence Advanced Research Projects Activity (IARPA) oversees high risk, high return programs on intelligence. The IARPA Babel program is focused on rapid development of speech recognition for novel languages, where development time and language-specific resources such as transcribed audio may be very limited. The IARPA Babel program develops robust speech recognition on various languages to provide effective search on massive amount of real-world recorded audio.

Thus far, the Babel corpus has been collected via cellphone in relatively quiet conditions, leading to relatively good quality telephone speech recordings. While word transcripts are provided, there are no pitch annotations. We applied the `genPitchLabel` utility that we used to generate the labels on the clean source audio of RATS corpus to generate the pseudo ground truth for the Babel corpus.

Although $\text{SAcC}_{\text{All CH}}$ performs better than $\text{SAcC}_{\text{Keele}}$ on most audio conditions, we found that $\text{SAcC}_{\text{All CH}}$ reported much more incorrect pitches on unvoiced portion of the Babel corpus than $\text{SAcC}_{\text{Keele}}$. To avoid this bias, we trained $\text{SAcC}_{\text{Babelnet}}$ for several epochs on Keele corpus starting with $\text{SAcC}_{\text{All CH}}$. The output of $\text{SAcC}_{\text{Babelnet}}$

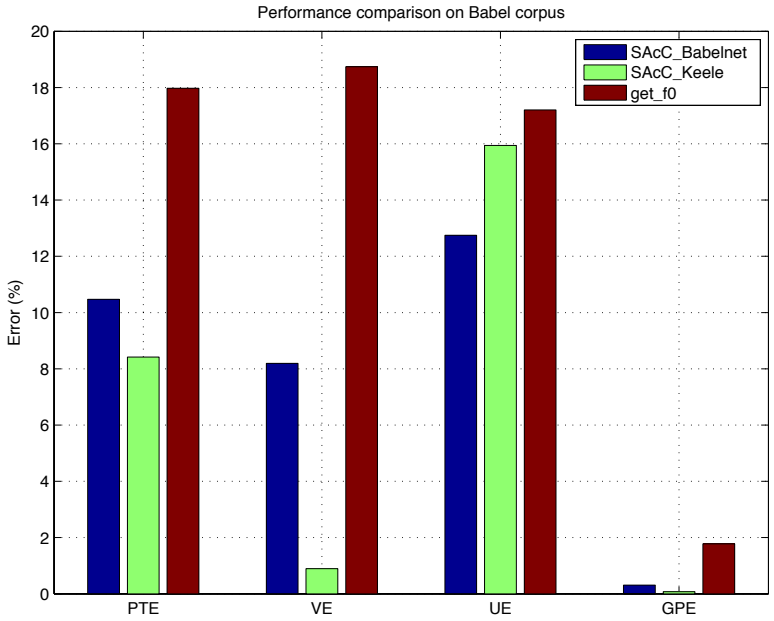


Figure 5.3: The mean PTE, GPE, VE, and UE of $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{Babelnet}}$, and `get_f0` on Babel corpus. The three labels for each algorithms were generated with `genPitchLabel` using leave-one-out strategy.

gave much fewer false pitches. The idea of $\text{SAcC}_{\text{Babelnet}}$ is to further train the general SAcC pitch tracker to adapt on a specific type of noise. $\text{SAcC}_{\text{Babelnet}}$ is obtained by re-training $\text{SAcC}_{\text{All CH}}$ on Keele corpus for several epochs.

The performance evaluation of $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{Babelnet}}$, and `get_f0` is shown in Figure 5.3. To avoid generating labels that agree on all three algorithms, we generated three different labels using `genPitchLabel` with the other two algorithms. When all algorithms are used to generate the `genPitchLabel` labels, they always agree on pitch to label voiced frames, giving zero VE and very small GPE for all algorithms. By generating three different labels, we can measure the performance of one algorithm against the agreement of the others. In PTE, $\text{SAcC}_{\text{Keele}}$ shows the best performance; $\text{SAcC}_{\text{Babelnet}}$ also performs very well compared to `get_f0`. In VE, $\text{SAcC}_{\text{Keele}}$ still performs the best. In UE sense, $\text{SAcC}_{\text{Babelnet}}$ performs the best, showing the superior

ability to avoid the false alarm problem.

5.6 Summary

In this Chapter, we successfully demonstrated the generalization performance of SAcC by showing a set of experiments on a real-world large-scale speech corpus with various unknown audio conditions/distortions.

We propose a method, `genPitchLabel`, to generate pitch label when no ground truth is available. We showed that the output of `genPitchLabel` agrees well with the ground truth pitch labels of the standard pitch corpora.

We showed that SAcC is capable of generalization across different noise conditions and different corpora. The specificity of SAcC, that improves the performance on the specific noise condition, can be obtained by training on the specific noise condition. We effectively showed that SAcC can be used for specificity or for generalization.

We also reported the evidence that SAcC pitch tracking output can be used to improve the performance of the state-of-the-art Speaker Identification (SID) system.

We tried the `genPitchLabel` on Babel corpus that comes with no ground truth pitch labels. Using a leave-one-out strategy to generate labels using `genPitchLabel`, we showed that SAcC can be specifically trained to avoid the common false alarm problem in band-limited corpus.

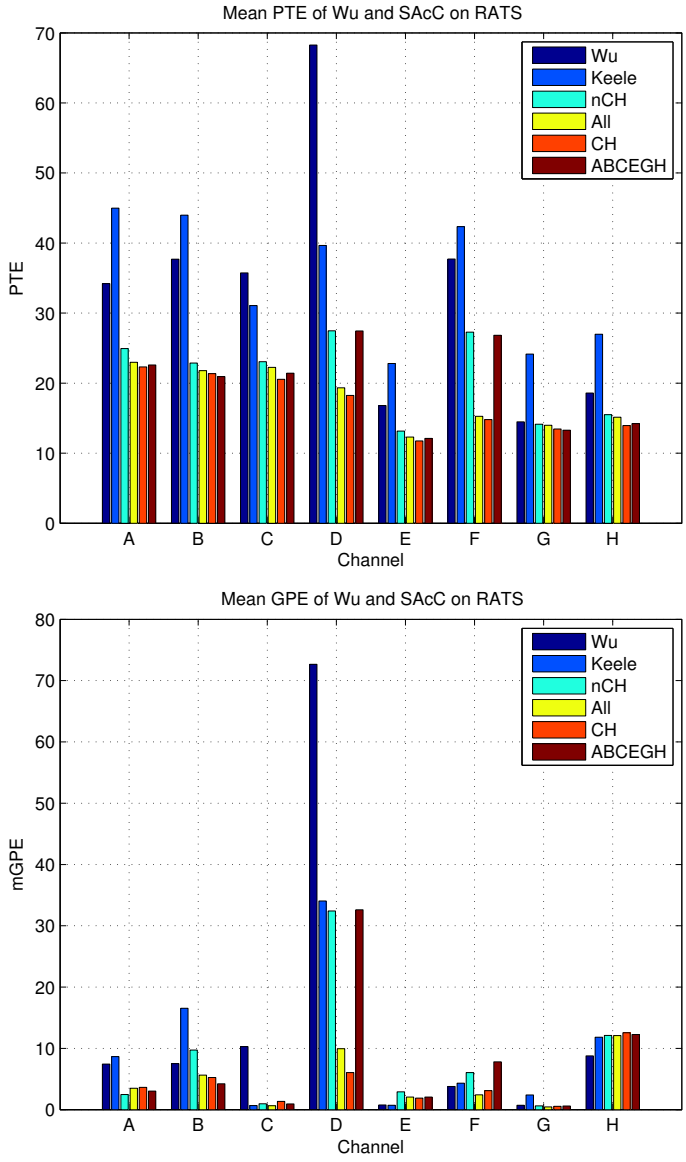


Figure 5.4: The mean (a) PTE and (b) GPE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, SAcC_{CH}, SAcC_{nCH}, and $\text{SAcC}_{\text{ABCEGH}}$ on RATS dataset.

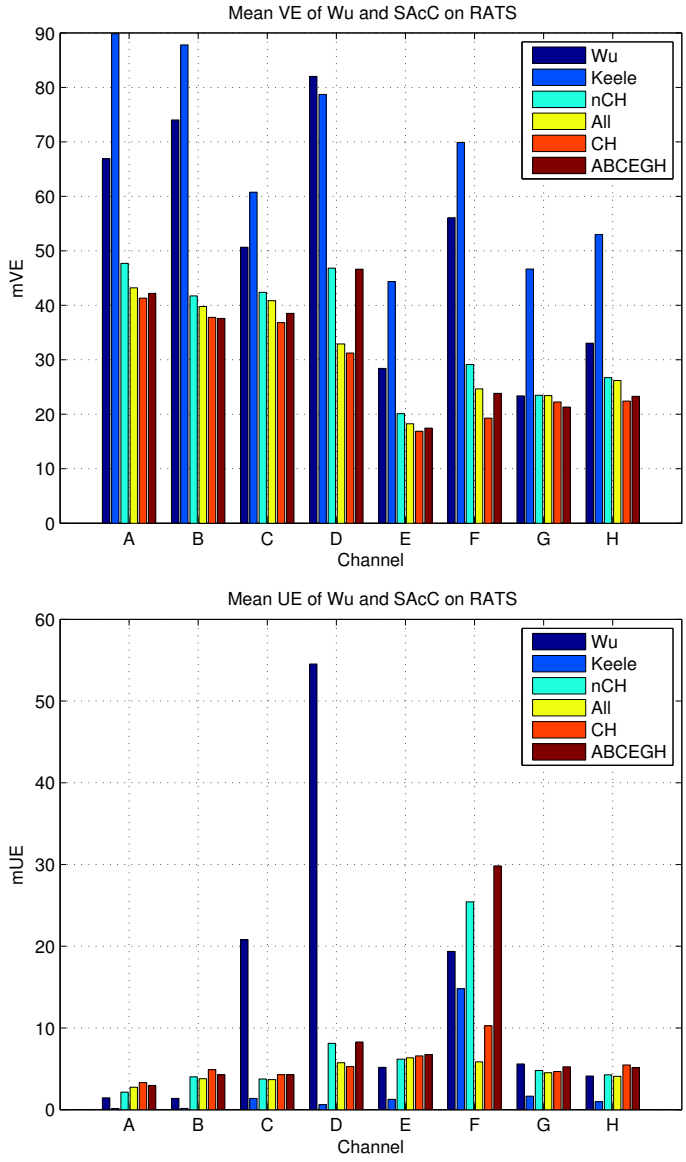


Figure 5.5: The mean (a) VE and (b) UE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, SAcC_{CH}, SAcC_{nCH}, and $\text{SAcC}_{\text{ABCEGH}}$ on RATS dataset.

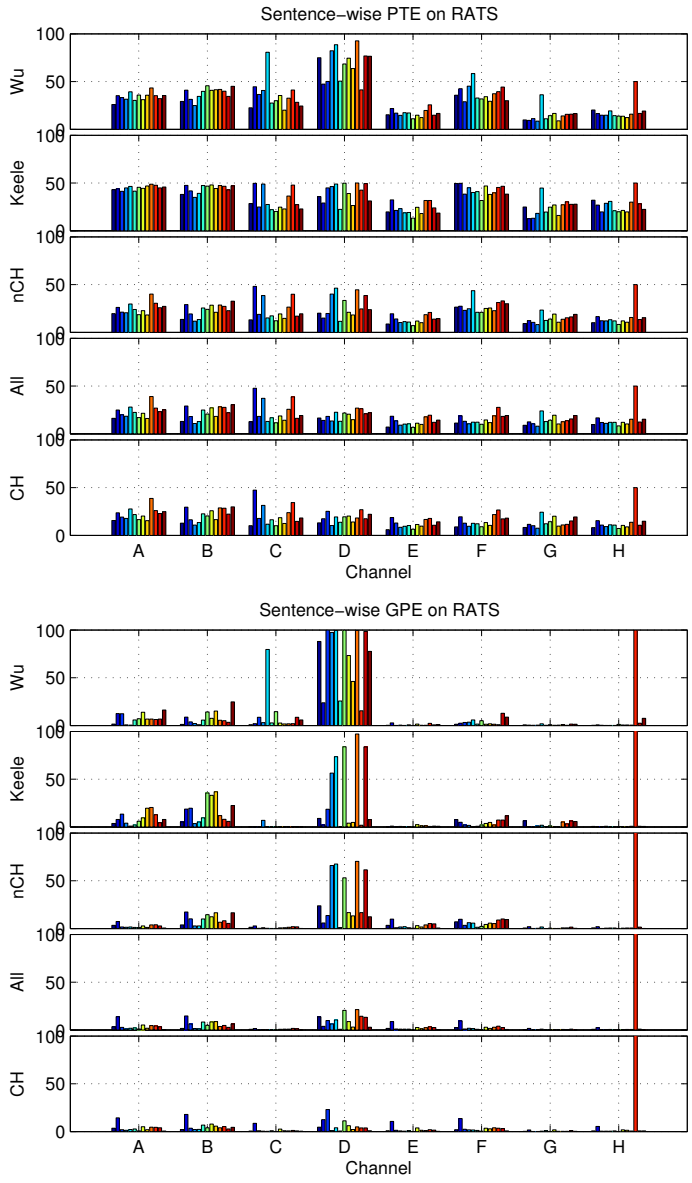


Figure 5.6: The sentence-wise (a) PTE and (b) GPE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, SAcC_{CH}, SAcC_{nCH}, and $\text{SAcC}_{\text{ABCEGH}}$ on RATS dataset.

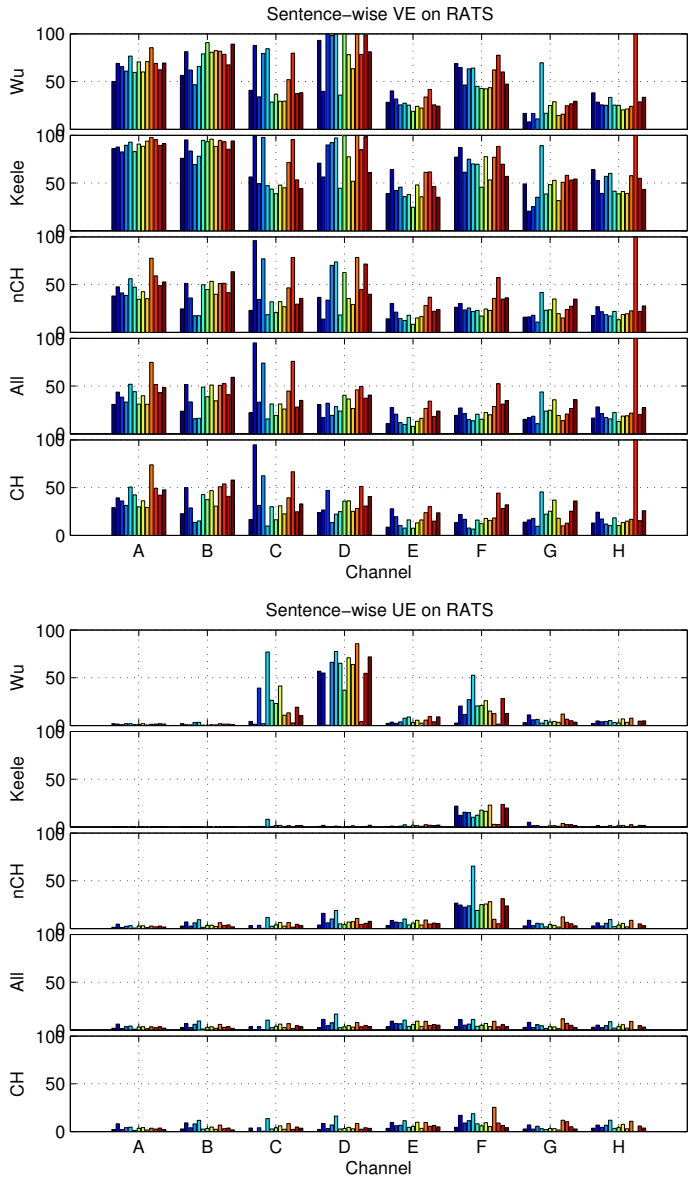


Figure 5.7: The sentence-wise (a) VE and (b) UE of Wu, $\text{SAcC}_{\text{Keele}}$, $\text{SAcC}_{\text{All CH}}$, SAcC_{CH}, SAcC_{nCH}, and $\text{SAcC}_{\text{ABCEGH}}$ on RATS dataset.

Chapter 6

Conclusions

in this thesis, we successfully verified the idea of fully classification-based pitch tracking on low-dimensional subband autocorrelation features. The initial idea was to develop a classification-based approach for pitch tracking. The subband autocorrelation was considered rich in information, but redundant for classification purpose. The PCA dimensionality reduction of the subband autocorrelation was shown to be effective both for the reconstruction task and for classification.

Overall, three major contribution of this thesis are as follows:

First, subband autocorrelation PCA feature is developed. To selectively exploit the frequency subband information against loss or degradation of one or more subbands, a filterbank is used to generate subband audio channels. For compact representation without losing pitch information, PCA is used to reduce the dimensionality.

Second, two classification-based pitch tracking systems, SubSel and SAcC, are proposed. SubSel selectively integrates subband autocorrelation according to the pitch information in the subband. SAcC goes further to calculate the strength of periodicity directly from the feature instead of using summarized autocorrelation across subband selectively. It turns out that SAcC outperforms the state-of-the-art pitch trackers under specific training noise conditions and generalizes well on other generic noises.

Third, a performance metric for pitch tracking, PTE is proposed to overcome unbalanced conventional performance metric, GPE. The idea is that, since the objective of pitch tracking is both pitch estimation and VAD, the metric should provide the performance in both sub-tasks.

Fourth, a consensus method for creating pseudo ground-truth, `genPitchLabel`, is proposed. This is particularly useful when the new corpus has either clean source audio or relatively low noise audio. We applied `genPitchLabel` on RATS and Babel corpora. When the source audio is available, such as in RATS corpus, the pseudo label output of `genPitchLabel` is very reliable. Using the generated labels, we were able to train SAcC that performs well both on the RATS test subset and on the FDA corpus, demonstrating the specificity and generalization of the trained SAcC. This is possible with the reliable pseudo label by `genPitchLabel`.

Fifth, we made the sources of SAcC[1], our Matlab implementation of the Wu pitch tracking system[2], and `genPitchLabel`[3] available online.

In conclusion, this thesis presents the noise robust speech pitch tracking systems using subband autocorrelation classifications. We showed that the proposed SAcC performs significantly better by the conventional GPE metric, as well as by the proposed PTE metric, both on standard corpora and on the real-world large-scale RATS corpus. The proposed SAcC is shown to adapt to various radio-band transmitted channels, as well as to generalize to the various noise conditions. For any unknown noise condition, SAcC can be trained on the specific noise conditions.

[1] http://labrosa.ee.columbia.edu/projects/SAcC/

[2] http://www.ee.columbia.edu/~bsl/projects/wu/

[3] http://www.ee.columbia.edu/~bsl/projects/genPitchLabel/

6.1 Future Works

In this section, we list a few potential directions for further works.

The PCA was a good initial choice for dimensionality reduction; and it worked. We can examine the explicit benefits of PCA for pitch tracking. We can also examine the effect of training set size and network size.

As we showed a preliminary case in the Babelnet case, we can further investigate various mixed training schemes. We also can further examine the effect of consensus ground truth.

As we observed the applications of SAcC in RATS SID, we can use SAcC on various applications such as speech multi-pitch tracking and music transcription.

Bibliography

[Bagshaw *et al.*, 1993] 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*, pages 1003–1006, September 1993.

[Camacho and Harris, 2008] A. Camacho and J. G. Harris. A sawtooth waveform inspired pitch estimator for speech and music. *J. Acoust. Soc. Am.*, 124(3):1638–1652, September 2008.

[Camacho, 2007] Arturo Camacho. *SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*. PhD thesis, University of Florida, Florida, 2007.

[Chu and Alwan, 2009] W. Chu and A. Alwan. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In *IEEE ICASSP*, pages 3969–3972, 2009.

[Chu and Alwan, 2012] W. Chu and A. Alwan. Safe: A statistical approach to f0 estimation under clean and noisy conditions. *IEEE Tr. Audio, Speech, and Lang. Proc*, 20(3):933–967, March 2012.

[de Cheveigne and Kawahara, 2002] A. de Cheveigne and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111(4):1917–1930, April 2002.

[de Cheveign, 2005] Alain de Cheveign. Pitch perception models. In Christopher Plack, Richard Fay, Andrew Oxenham, and Arthur Popper, editors, *Pitch*, vol-

ume 24 of *Springer Handbook of Auditory Research*, pages 169–233. Springer New York, 2005.

[Gold *et al.*, 2011] Ben Gold, Nelson Morgan, and Dan Ellis. Pitch detection. In *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, pages 455–472. John Wiley & Sons, Inc., 2011.

[Joho *et al.*, 2007] Dominik Joho, Maren Bennewitz, and Sven Behnke. Pitch estimation using models of voiced speech on three levels. In *IEEE ICASSP*, pages 1077–1080, 2007.

[Klapuri, 2006] Anssi Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *in ISMIR*, pages 216–221, 2006.

[Klapuri, 2008] A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Tr. Audio, Speech, and Lang. Proc*, 16(2):255–266, February 2008.

[Lee and Ellis, 2006] Keansub Lee and Daniel P. W. Ellis. Voice activity detection in personal audio recordings using autocorrelogram compensation. In *Interspeech*, pages 1970–1973, Oct 2006.

[Lee and Ellis, 2012] Byung Suk Lee and Daniel P. W. Ellis. Noise robust pitch tracking by subband autocorrelation classification. In *Interspeech*, September 2012.

[Licklider, 1951] J. C. R. Licklider. A duplex theory of pitch perception. *Experientia*, 7:128–138, 1951.

[Meddis and Hewitt, 1991] R. Meddis and M.J. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. *J. Acoust. Soc. Am.*, 89(6):2866–2882, 1991.

[Nakatani and Irino, 2004] Tomohiro Nakatani and Toshio Irino. Robust and accurate fundamental frequency estimation based on dominant harmonic components. *The Journal of the Acoustical Society of America*, 116(6):3690–3700, 2004.

[Picone, 1993] Joseph W. Picone. Signal modeling techniques in speech recognition. In *PROCEEDINGS OF THE IEEE*, pages 1215–1247, 1993.

[Plante *et al.*, 1995] F. Plante, G. F. Meyer, and W. A. Ainsworth. A pitch extraction reference database. In *EUROSPEECH*, pages 837–840, September 1995.

[Sha *et al.*, 2004] Fei Sha, Ashley Burgoyne, and Lawrence Saul. Multiband statistical learning for f0 estimation in speech. In *IEEE ICASSP*, 2004.

[Slaney and Lyon, 1990] M. Slaney and R.F. Lyon. A perceptual pitch detector. In *IEEE ICASSP*, pages 357–360. IEEE, 1990.

[Talkin, 1995] D. Talkin. A robust algorithm for pitch tracking (RAPT). In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*, chapter 14. Elsevier Science B.V., 1995.

[Tan and Alwan, 2011] Lee Ngee Tan and Abeer Alwan. Noise-robust f0 estimation using snr-weighted summary correlograms from multi-band comb filters. In *IEEE ICASSP*, pages 4464–4467, 2011.

[Tolonen and Karjalainen, 2000] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE Tr. Speech and Audio Proc.*, 8(6):708–716, November 2000.

[Wu *et al.*, 2003] M. Wu, D.L. Wang, and G.J. Brown. A multipitch tracking algorithm for noisy speech. *IEEE Tr. Speech and Audio Proc.*, 11(3):229–241, May 2003.