# Vocal Melody Extraction from Polyphonic Audio with Pitched Accompaniment

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

by

**Vishweshwara Mohan Rao**

Roll No. 05407001

Supervisor

**Prof. Preeti Rao**


**DEPARTMENT OF ELECTRICAL ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY BOMBAY**

**2011**

# Dissertation Approval for Doctor of Philosophy

The thesis entitled **Vocal Melody Extraction from Polyphonic Audio with Pitched Accompaniment** by **Vishweshwara Mohan Rao** is approved for the degree of **Doctor of Philosophy**.

External Examiner

(Prof. Hema Murthy)

Internal Examiner

(Prof. P. C. Pandey)

Supervisor

(Prof. Preeti Rao)

Chairman

(Prof. D. N. Singh)

Date: June 13, 2011

Place: IIT Bombay, Mumbai

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and references the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

VISHWESHWARA MOHAN RAO

(Name of the student)

05407001

(Roll no.)

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY, INDIA

# CERTIFICATE OF COURSE WORK

This is to certify that **Vishweshwara Mohan Rao** (Roll No. 05407001) was admitted to the candidacy of Ph.D. degree in July 2006, after successfully completing all the courses required for the Ph.D. programme. The details of the course work done are given below.

| S. No. | Course Code | Course Name | Credits |
|---|---|---|---|
| 1 | EE 601 | Statistical Signal Analysis | 6 |
| 2 | EE 603 | Digital Signal Processing and its Applications | 6 |
| 3 | EE 679 | Speech Processing | 6 |
| 4 | EES 801 | Seminar | 4 |
| 5 | HS 699 | Communication and Presentation Skills | 0 |
| 6 | EE 608 | Adaptive Signal Processing | 6 |
| | | **Total Credits** | **28** |

IIT Bombay

Date: Dy. Registrar (Academic)

# Abstract

The melody of a song is an important musical attribute that finds use in music information retrieval, musicological and pedagogical applications. Melody extraction systems attempt to extract this attribute from polyphonic musical audio as the detected pitch of the lead melodic instrument, typically assumed to be the most salient or dominant instrument, in the polyphonic mixture. Contemporary melody extraction algorithms are particularly prone to error when the audio signal contains loud, pitched accompanying instruments, which compete for local dominance with the lead melodic instrument. Such accompaniment often occurs in non-Western or ethnic genres of music such as Indian classical music. This thesis addresses the problem of vocal melody extraction from polyphonic music in such a cross-cultural context. Vocal melody extraction involves two stages – the extraction of the pitch of the predominant pitched instrument at all time instants, called predominant-F0 extraction, and the detection of audio segments in which the singing voice is present, called singing voice detection. Algorithms for each of these stages are designed with a focus on robustness to the presence of loud pitched accompaniment and are subsequently evaluated using cross-cultural (Western and non-Western) musical audio. For both stages a sparse signal representation, sinusoidal frequencies and amplitudes, is used since it enhances tonal components in the music and suppresses percussive elements. In the design of the predominant-F0 extraction stage a measure of local salience, biased towards singing voice characteristics, and a dynamic programming-based optimum path finding technique with melodic smoothness constraints, are chosen. The possibility that an accompanying pitched instrument, instead of the singing voice, may dominate the polyphony is accounted for by tracking an

additional F0 trajectory. The final voice-pitch trajectory is identified based on its characteristic pitch dynamics. The singing voice detection stage follows a machine-learning framework for which a combination of features is proposed that is able to discriminate between singing voice and instrumental segments. *A priori* knowledge of the predominant-F0 trajectory was used to isolate the predominant source spectrum before feature extraction leading to the improved performance of conventionally used static timbral features. Further, complementary information that distinguishes singing voice from instrumental signal characteristics, such as the presence of articulatory (timbral) changes in lyric singing and voice pitch dynamics, was exploited by the design of dynamic timbral and F0-harmonic features respectively. The proposed predominant-F0 extraction and singing voice detection algorithms are shown to be robust to loud pitched accompaniment across different culturally-distinct genres of music and outperform some other contemporary systems. Finally, a novel graphical user interface, for the semi-automatic operation of the proposed melody extraction algorithm is designed and is shown to facilitate the extraction of high-accuracy melodic contours with minimal human intervention from commercially available music. An accompanying web-site with related sound examples can be accessed at http://www.ee.iitb.ac.in/daplab/VishuThesis

# Table of Contents

# List of Figures

# List of Tables

# List of Symbols

| | |
|---|---|
| ACF | Auto-Correlation Function |
| AME | Audio Melody Extraction |
| CA | Chroma Accuracy |
| CMNDF | Cumulative Mean Normalized Difference Function |
| CV | Cross-Validation |
| DP | Dynamic Programming |
| EM | Expectation Maximization |
| ESACF | Enhanced Summary Auto-Correlation Function |
| F0 | Fundamental frequency |
| FFT | Fast Fourier Transform |
| GDK | Gaussian Difference Kernel |
| GI | Gini Index |
| GMM | Gaussian Mixture Model |
| GUI | Graphical User Interface |
| HMM | Hidden Markov Model |
| HSM | Harmonic Sinusoidal Model |
| IDFT | Inverse Discrete Fourier Transform |
| IF | Instantaneous Frequency |
| KU | Kurtosis |
| LPC | Linear Predictive Coefficient |
| MER | Modulation Energy Ratio |
| MFCC | Mel-Frequency Cepstral Coefficient |
| MI | Mutual Information |
| MIDI | Musical Instrument Digital Interface |
| MIR | Music Information Retrieval |
| MIREX | Music Information Retrieval Evaluation eXchange |
| MLP | Multi-Layer Perceptron |
| NCPA | National Centre for Performing Arts |
| NHE | Normalized Harmonic Energy |
| PA | Pitch Accuracy |
| PDA | Pitch Detection Algorithm |
| PDF | Probability Distribution Function |
| PM | Pattern Matching |
| PT | Partial Tracking |
| QBSH | Query-by-Singing-Humming |
| SAR | Signal-to-Accompaniment Ratio |

| | |
|---|---|
| SC | Spectral Centroid |
| SCF | Spectral Change Function |
| SD | Standard Deviation |
| SE | Sub-band Energy |
| SER | Sub-band Energy Ratio |
| SF | Spectral Flatness |
| SHS | Sub-Harmonic Summation |
| SIR | Signal-to-Interference Ratio |
| SPS | SPectral Spread |
| SRA | Sangeet Research Academy |
| SRO | Spectral Roll-Off |
| SS | Spectral Subtraction |
| ST | Semi-Tone |
| STFT | Short-Time Fourier Transform |
| STHE | Sinusoidal Track Harmonic Energy |
| SVD | Singing Voice Detection |
| SVM | Support Vector Machine |
| T-F | Time-Frequency |
| TWM | Two-Way Mismatch |
| V | Voice |
| VT | Voice + Tabla |
| VTT | Voice + Tabla + Tanpura |
| VTTH | Voice + Tabla + Tanpura + Harmonium |

# Chapter 1

## Introduction

Music Information Retrieval (MIR) is a research area motivated by the need to provide music listeners, music professionals and the music industry with robust, effective and friendly tools to help them locate, retrieve and experience music. MIR is an interdisciplinary area, involving researchers from the disciplines of computer science, signal processing, musicology, cognitive science, library and information science, to name a few. A massive surge in MIR research has been primarily fueled by the tremendous growth in the digital music distribution industry propelled by the worldwide penetration of the Internet, portable music players and most recently mobile technology, and the availability of CD-quality compressed audio formats and increased Internet and mobile bandwidths. More recently, MIR has come out of the confines of information extraction for retrieval purposes only and extended to research involving music cognition and perception, music creation, music education and musicology.

Typical MIR systems operate in two stages: 1. the extraction of information from the music signal, 2. the use of this information for some meaningful application like search and retrieval. The techniques used in the first stage are referred to as music content analysis techniques. Research in this area ranges from the extraction of low-level signal descriptors (like spectral coefficients) to mid-level, musically meaningful, descriptors (such as melodic

and rhythmic attributes) to high level descriptors (such as artist, album, genre, and mood information). In this work the focus is on the automatic extraction of melodic attributes from commercially available music.

In this introductory chapter we present the main motivations, scope and contributions (major and minor) of this research work, and the overall organization of the dissertation.

## 1.1 Objective and Motivation

The primary aim of this thesis is to investigate the problem of automatic melody extraction from polyphonic music and provide a novel and effective solution thereof. The definition of melody in this context is the pitch[1] contour of the lead instrument, here considered to be the human singing voice. Polyphonic music is defined as audio in which multiple musical instruments (pitched and percussive) are simultaneously present. The motivations for studying this particular problem are given below.

### 1.1.1 Personal Interest

Most researchers in the field of MIR are either professionally trained engineers or musicians or often both. It is this inclination in both the scientific and creative disciplines that facilitates the design of relevant and effective MIR systems. As a singer, musician and engineer my personal motivation in this area was from the pedagogical perspective of Indian music.

From the compositional and listener standpoint, most Indian music, popular or classical, is always centered around the melody. Unlike western music, the concept of harmony, though not uncommon, is seldom the central focus of the song. I found that different singing skill sets were required when rendering the melody for the western rock/pop genres as compared to most Indian music genres. While western rock/pop music required strong voice projection, large pitch-range, good voice quality, singing stamina and smart microphone usage, the primary focus in Indian music was the usage of subtle pitch movements, which I found to be a more difficult aspect to gain control over than all the previous singing skills mentioned. Two types of pitch control were required in Indian singing. The first, which is common to western singing as well, is the accuracy of held notes with respect to note locations on an equally tempered musical scale. The second, and more difficult, is the correct rendition of pitch ornaments and inflections manifested as patterns of pitch modulations. These are extensively

[1] Although the term pitch is known to be a perceptual attribute of sound and fundamental frequency, referred to as F0, is considered its physical correlate, both terms are used interchangeably throughout this thesis to refer to the physical parameter.

used as they serve important aesthetic and musicological functions within the context of Indian music. The use of subtle pitch ornaments originated in Indian classical music but is also common practice in Indian folk and popular (film) music, since these are related to Indian classical music. In fact even singers in popular Indian music can be immediately identified as having gone through classical training or not, based on the degree of ornamentation used in their singing.

In the absence of a human teacher, the next best option in aiding a singer render such pitch ornaments accurately would be an interactive feedback system with the facility to record a user's singing and compare or evaluate his/her rendition to an 'idol' (reference) singer, and provide some meaningful feedback to the user. One of the essential components of such a system is a melody detection tool that can extract the pitch contour of the lead singer from a polyphonic recording, which is the subject of this thesis.

### 1.1.2 Relevance

#### *1.1.2.1 MIR Applications*

Classical MIR systems involve searching an audio database and retrieving a relevant audio document based on a query that is also audio (non-textual). When an exact match between the query and the reference is required, the audio representation used is often called a fingerprint and usually relies on low-level audio features such as spectral coefficients. Often the retrieval system is expected to return a result that is musically relevant/similar to the query. A typical case of such a system is a query-by-singing/humming (QBSH) system in which a user sings or hums the audio query into the system and the returned result will be the metadata information of the original song(s) with the same/similar tune. In such a context the invariant information, between the query and reference, which can be effectively utilized in the search, is the melody of the song. Indeed in her work on melodic comparison, E. Selfridge-Field (1998) states that "it is the melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text." One definition of melody, proposed by Levitin (1999), also highlights the invariant quality of the melody by describing it as being robust to transformations such as tempo changes, changes in singing style and instrument changes. A melody detection system for polyphonic music is essential to building a reference melodic template for songs against which all subsequent queries will be matched.

Other uses of an automatic melody extraction system that can be envisioned are:

1. Music edutainment: where the extracted melody from a professional polyphonic recording of music could be used as a reference against which the renditions of amateur musicians are evaluated along different dimensions such as pitch, rhythm and expression.
2. Plagiarism/version identification: where a comparison between the extracted melodies of 2 polyphonic songs could indicate musical similarity.
3. Structural segmentation: where repeated parts within a song, such as chorus' and verses, could be identified by a segmental analysis of the extracted melodic contour.
4. Music transcription: where the extracted melody can be fed into a transcription system to provide a musical representation that could be used by music composers and also for musicological analysis.
5. Source separation: where the prior knowledge of the melodic contour would help isolate/suppress the lead (melodic) instrument from the polyphonic mixture. An isolated lead instrument track would be beneficial for lyric-alignment and voice-morphing applications while audio in which the lead instrument is suppressed could be used for karaoke generation.

#### *1.1.2.2 Cross-Cultural Motivation*

MIR technologies have been by-and-large motivated by the concepts and identities of western music (Lidy, et al., 2010; Downie, 2003). This has been primarily attributed to Western music having the largest trans-cultural audience and the familiarity of the developers with Western music. However, more recently, the need for developing MIR tools specifically tailored for ethnic music[1] has resulted in some studies that evaluate the use of existing MIR techniques on ethnic music, primarily for the evaluation of genre classification (Tzanetakis, Kapur, Schloss, & Wright, 2007; Gomez & Herrera, 2008). In the context of content analysis, and in particular pitch processing, it has been recognized that MIR tools that are based on Western music concepts may not be suitable for analyzing ethnic music, which may be replete with a variety of pitch modulation patterns, such as Chinese guqin music, and/or which may not conform to the basic equal temperament tuning scale of Western music, such as Indian classical, Middle-eastern, Central-African and Indonesian music (Cornelis, Lesaffre, Moelants, & Leman, 2009). This emphasizes the need for incorporating culture-specific musical attributes in order

[1] Ethnic music in these studies is defined as any music that is of a non-Western genre. Examples of Western genres include rock, pop, blues, jazz, classical, opera, electronic, hip-hop, rap, hard-rock, metal heavy metal.

to increase the cross-cultural robustness of music analysis techniques. Alternatively, prior knowledge of distinct differences in musical concepts between various music cultures could be used to develop more effective music analysis tools specific to the individual cultures. This type of analysis would be especially useful in order to derive culture-specific musical information from the audio signals.

## 1.2 Scope

In this section we first give a brief overview of standard approaches to melody extraction. We then identify scenarios in which these approaches make errors and define the scope of the thesis in this context.

### 1.2.1 Previous Approaches

Pitch detection algorithms (PDAs) in audio signal processing, especially in speech processing, have been an active topic of research since the late twentieth century. A comprehensive review of the early approaches to pitch detection in speech signals is provided in (Hess, 1983) and a comparative evaluation of pitch detection algorithms in speech signals is provided in (Rabiner, Cheng, Rosenberg, & McGonegal, 1976). A more recent review of previous approaches to pitch detection in speech and music signals is provided in (Hess, 2004). The general recent consensus is that pitch detection or tracking for monophonic signals (speech or music) is practically a solved problem and most state-of-the-art approaches yield high quality and acceptable solutions (Hess, 2004; Klapuri, 2004). The problem of melody extraction from polyphony is different from the monophonic speech pitch detection problem in two major aspects: 1. Multiple sound sources (pitched and unpitched) are usually simultaneously present, and 2. the characteristics of the target source (here the singing voice) are a larger pitch range, more dynamic variation, and more expressive content than normal speech.

The last decade has seen a large volume of independent studies on the melody extraction problem. However there were no common benchmarks for comparative evaluation of various proposed algorithms. Recently a concerted effort by the MIR community to provide a common evaluation platform, in terms of common test datasets and evaluation metrics for different MIR tasks, one of them being audio melody extraction (AME), resulted in the Music Information Retrieval Evaluation eXchange (MIREX). The MIREX platform has found wide-spread popularity within the MIR community as an indicator of state-of-the-art in MIR. An overview of MIREX, in terms of the breadth of the tasks, past work and future

challenges, is available in (Downie, 2008). In the context of melody extraction a comprehensive review and evaluation of the algorithms submitted to the 2004 and 2005 AME tasks is available in (Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007).

The majority of algorithms for melody extraction from polyphonic music, described in previous literature and at MIREX, adopt the "understanding-without-separation" paradigm as described by Scheirer (2000) and by-and-large adhere to a standard framework, as depicted in Figure 1.1. Here, initially a short-time, usually spectral, signal representation is extracted from the input polyphonic audio signal. This representation is then input to a multi-F0 analysis block whose goal is to detect multiple candidate F0s and associated salience values independently in each analysis time-frame. The predominant-F0 trajectory extraction stage attempts to identify a trajectory through the F0 candidate-time space that describes the pitch contour of the locally dominant pitched source in the song. This is expected to represent the voice-pitch contour when the voice is present and any other, locally dominant, pitched instrument when the voice is absent. The singing voice detection block identifies which segments of the extracted predominant-F0 contour represent sung segments as opposed to instrumental segments. This block is often considered as an independent problem called singing voice detection (SVD) in polyphonic music.

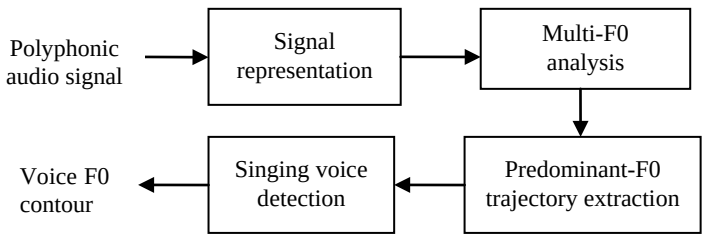


Figure 1.1: Block diagram of a typical melody extraction system

There also exist melody extraction algorithms that do not follow the above paradigm such as those that that attempt to first segregate the melodic source and then track the pitch of the extracted "monophonic" source (Lagrange, Gustavo Martins, Murdoch, & Tzanetakis, 2008). Alternatively, some systems adopt a machine-learning based approach to melody extraction (Poliner & Ellis, 2005).

### 1.2.2 Problems in Melody Extraction

As previously mentioned, polyphony (the simultaneous presence of multiple musical sources) was one of the major problems that needed to be addressed in the design of melody extraction algorithms. Dressler (2010) in her review of the trends in melody extraction at MIREX over the past 5 years states that there has been no ‘bold increase’ in system performance. She indicates that the main problems are faced during audio segments in which the instrumental accompaniment is of comparable or greater volume than the vocal parts and suggests the use of pitch-continuity and timbral characteristics for improving performance. In the conclusions of his thesis on "Melody Detection in Polyphonic Audio", Paiva (2006) states that his algorithm "shows some difficulties when the assumption that the melodic notes are usually salient in the mixture fails." This indicates that one of the challenges for melody extraction is the presence of multiple pitched instruments, as opposed to percussive instruments, which compete for local dominance in a polyphonic signal. He suggests incorporating melodic smoothness constraints to correct errors in his system. The presence of strong, pitched accompaniment that competes with the voice for local dominance is also mentioned as a major cause of errors in vocal-pitch tracking for other polyphonic melody extraction systems, some of which incorporate melodic smoothness constraints as well (Durrieu J. L., Richard, David, & Fevotte, 2010; Goto, 2004; Klapuri, 2008; Li & Wang, 2005; Paiva, Mendes, & Cardoso, 2006).

The presence of the dominant pitched instrument during segments in which the singing voice is absent often leads to errors in the singing voice detection stage. Typical approaches to singing voice detection require pre-training of a classifier on timbral features extracted from vocal and non-vocal segments. The variability in intra-voice and inter-instrument timbre often leads to misclassification for such signals (Durrieu J. L., Richard, David, & Fevotte, 2010).

Another complication in the melody extraction problem is the large diversity and variability in the dynamic characteristics of the singing voice. Specifically, music signals that display large vocal pitch ranges and temporal dynamic variations in loudness, pitch and syllabic rate, could degrade the performance of a melody extraction system. For example, female operatic singing, with very high pitches and large pitch dynamics (manifested as vibrato), has been identified as a problematic case for systems which use an autocorrelation function (ACF)-based approach to pitch detection, due to the limited resolution of the ACF at higher octaves (Li & Wang, 2007). Poliner et. al. (2007) stated that systems that incorporate

note-modeling i.e. attempt to round-off each pitch estimate to the nearest semitone (in a grid computed using standard Western tuning frequency) are not suitable when accurate pitch tracking of the human singing voice is a requirement.

Apart from the competing pitched instrument problem, Paiva (2006) also stresses the need to test melody extraction algorithms on larger and more diverse data-sets, since the data used at the MIREX evaluations is limited in size and diversity. The previous studies have only considered Western singing and their conclusions may not be relevant in the context of the variety of singing styles present across different musical cultures In order to address culture-specific peculiar musical characteristics, such as melodic ornamentations and non-standard tuning, melody extraction systems should specifically incorporate culture-specific musical information in their design.

### 1.2.3 Scope of this Thesis

In this investigation we consider the automatic extraction of the voice-pitch contour from polyphonic vocal music i.e. music in which the lead melodic instrument is the human singing voice. One aim of this work is to build a novel and practically useful melody extraction system that is on par with the state-of-the-art, which can be used as a front end for future QBSH or pedagogical applications. Another important aim is an attempt to address the identified open research areas in the context of melody extraction as described in the previous section, namely the presence of competing pitched accompaniment, singing-related issues such as F0 range and dynamics, and culture-dependent musical considerations, specifically for Indian music. Singing in Indian classical music is known to display large and rapid variations in pitch over time. The use of a loud pitched accompanying instrument, such as a harmonium or violin, which plays a melodic line sometimes similar to the singer's melody, is also common. This complicates the consistent identification of the voice-pitch contour as the predominant-pitch and the discrimination between isolated voice and accompaniment regions of the predominant-F0 contour.

## 1.3 System Overview

The central theme of this thesis is the design of a novel, practically useful system for high-accuracy melody (voice-pitch contour) extraction from polyphonic music. In the design of our system we adopt the paradigm of Figure 1.1. In this section we provide an overview of

the complete melody extraction system with a brief description of the functionality for each of the system modules.

The design of this system was achieved in two stages. In the first stage we designed a system, which was found to achieve performance on par, if not better, with other state-of-the-art melody extraction systems, in terms of accuracy and computational time, in internal evaluations and also at the MIREX 2008 and 2009 evaluations. In the second design iteration, we attempted improving the performance of the previous system for identified problematic categories of signals, such as the presence of loud competing pitched accompaniment, by incorporating novel design features in the predominant-F0 trajectory formation and singing voice detection stages.

1. *Signal Representation*

In the signal representation module we extract a sparse, frequency domain representation of the polyphonic signal in terms of local harmonic/sinusoidal components and their parameters (frequencies, magnitudes and phases). This approach can be thought of as a tonal content enhancement stage where any noisy sounds, such as un-pitched percussion, will not be represented at the output. The performance of different sinusoid detection algorithms was comparatively evaluated and a main-lobe matching algorithm was selected for its superior performance. We also investigated the use of signal sparsity measures to control the analysis frame length in order to obtain a minimally distorted sinusoidal representation for signal conditions such as polyphony and non-stationarity.

2. *Multi-F0 Analysis*

In the multi-F0 analysis module we comparatively evaluated several pitch salience functions, derived from known monophonic PDAs, in terms of the relative voice-pitch salience in the presence of harmonic interference. A spectral harmonic-matching salience function was found to be relatively robust to harmonic interference. It was also found that distinctly separating the F0 candidate identification and salience computation stages leads to superior voice-F0 detection accuracy.

3. *Predominant-F0 Trajectory Extraction*

This stage utilizes a dynamic programming (DP)-based optimal path finding algorithm, through the F0 candidate v/s time space, to compute a predominant-F0 contour. The local measurement costs of the F0 candidates are their salience values as computed by the previous module. A novel smoothness cost that applies increasing penalties to larger pitch jumps in log

frequency space is proposed. To increase robustness to pitched accompaniment the DP framework is extended to tracking pairs of F0-candidates and selecting one final trajectory.

4. *Singing Voice Detection*

Our singing voice detection module applies a known machine learning-based approach of labeling short-term frames of audio as vocal or non-vocal using a classifier previously trained on manually annotated signals. These frame-level decision labels are then combined over segments, automatically detected using a known boundary detection algorithm, to output segment-level voicing labels. To increase robustness to pitched accompaniment we augment our static timbral feature set with dynamic timbral and harmonic features.

5. *Semi-Automatic Interface*

In order to effectively evaluate our melody extraction system with different parameter settings for different signals we designed a semi-automatic graphical user interface. We incorporated several novel features such as the ability to change system parameters over different segments of the same audio file, combined visual (spectrogram) and auditory (natural re-synthesis) feedback and non-vocal segment labeling. This interface was found to be very useful for extracting high-resolution voice-pitch contours from music for building reference templates for a QBSH system and for a singing evaluation system.

## 1.4 Thesis Organization

The organization of this thesis is largely based on the order of the individual modules in the melody extraction system. Chapter 1 is an introductory chapter that discusses the main objectives, motivation and scope of this thesis. Chapter 2 presents an overview of the melody extraction problem in a cross-cultural scenario, specifically in the context of Indian music. Chapters 3, 4, 5 and 6 discuss the design considerations for each of the signal representation, multi-F0 analysis, predominant-F0 trajectory extraction and singing voice detection stages respectively. Each of these chapters starts with a description of the objective of that particular module and a review of the different approaches that have been described in the literature. Then a description of the different choices in design along with any investigations within this module is given. This is followed by a discussion and a summary of the chapter. Here the singing voice detection module is treated as a stand-alone system. Next we discuss the evaluation of our melody extraction system, which is divided into two parts. Chapter 7 describes the evaluation of our predominant-F0 extraction and singing voice detection

systems on real signals that led to the design of our MIREX submission. This is then followed by the evaluation of our system at MIREX 2008 and 2009. Chapter 8 evaluates the enhancements made to the previous system, specifically to the predominant-F0 trajectory extraction and singing voice detection modules, to address the problem of a competing pitched accompanying instrument. The new predominant-F0 extraction system is evaluated against a state-of- the-art sung melody extraction system using a database of signals that contain significantly loud pitched accompaniment. The new singing voice detection system is evaluated on similar signals taken from a cross-cultural dataset. Chapter 9 then describes the considerations that were involved in the design of the semi-automatic melody extraction interface. Chapter 10 concludes the thesis and gives directions for future work. This is followed by an Appendix, which contains pre-processing strategies for flat-note instrument suppression. Last a bibliography of all cited references and a list of publications that have arisen out of the thesis are listed. Related sound examples are available in an accompanying website at http://www.ee.iitb.ac.in/daplab/VishuThesis

# Chapter 2

# Melody Extraction in a Cross-Cultural Context: An Indian Perspective

In this chapter we provide an overview of the melody extraction problem in the context of Indian music. The goal of this chapter is to motivate an investigation in melody extraction in Indian music, and it is not intended to be a musicological discourse. We start by explaining some terminology that is particular to the general melody extraction problem. Then we give a brief overview of Indian music, emphasizing its diversity, and contrast it with Western music. Next we offer our perspective on some applications of melody extraction in the Indian context. Following this we describe the signal characteristics of a typical classical Indian music setting, and describe the musical signal complexity from the melody extraction perspective. We find that two particular characteristics of Indian music increase signal complexity and directly affect the melody extraction problem. These are the presence of loud, harmonically rich accompanying instruments and the use of rapid pitch modulations in

singing. Such characteristics are also found in Greek Rembetiko and Arabic music. Finally we describe the databases and metrics used for evaluation in this thesis.

## 2.1 Terminology

This thesis primarily deals with signal processing techniques for musical audio and uses jargon from the disciplines of electrical engineering (signal processing), music, musicology, and computer science. In order to provide clarity in the readability of this work we first define some often-used terms in the context of this thesis.

*Pitch and pitch detection*

The term ‘pitch’ itself is a perceptual quality of sound and has been originally defined as that auditory attribute according to which sounds can be ordered on a scale from low to high. The physical correlate of the pitch is the fundamental frequency (F0), which is defined as the inverse of the fundamental period (the smallest positive value of a time-shift that leaves a perfectly periodic signal unchanged.) From the perceptual viewpoint, the fundamental frequency is defined as the frequency of the sinusoid that evokes the same perceived pitch as the complex sound signal. Throughout this study both terms (pitch and F0) are used interchangeably to refer to the physical parameter. Such usage has been considered acceptable in previous literature on related topics. Pitch detection refers to the process of extracting the F0 information from an audio signal. However other terms such as pitch estimation and pitch tracking have also been used and have the same meaning as pitch detection. The pitch trajectory or pitch contour refers to a sequence of pitch values across time.

*Polyphony*

The original theoretical definition of polyphony or polyphonic music refers to musical audio containing two or more independent melodic lines. In this study, as in related music signal processing literature (Durrieu J. L., Richard, David, & Fevotte, 2010; Klapuri, 2008; Paiva, Mendes, & Cardoso, 2006), polyphony has been defined as musical audio in which several musical sources, such as singing voice, guitar and drums, are simultaneously present. In contrast monophonic music is defined as music in which there is only a single musical source. Homophonic music is a subset of polyphonic music in which one dominant melodic voice is accompanied by chords.

*Voice and Voicing*

In studies related to polyphonic music often there are references to multiple voices where ‘voice’ indicates a single musical source. However, in this study we have used the term ‘voice’ to specifically refer to the human singing voice. For other sound sources, such as other musical instruments, we have used the term ‘instrument’ or ‘accompaniment’.

In speech processing literature ‘voicing’ refers to the presence of utterances in which there is a periodic vibration of the vocal folds i.e. pitched content, such as vowels, diphthongs, glides, voiced stops and voiced fricatives. In the context of singing voice processing we use a broader (segmental) definition of voicing as the locations in the song where the singing voice is present. Voiced regions in the song are then delineated by the start and end locations of sung-phrase boundaries. Here a phrase is that continuous sung segment in which the singer does not make a long pause such as for a breath.

*Melody and Predominant-F0*

The term melody has been defined in different ways depending on its musical, musicological, cultural, emotional and perceptual background and implications (Paiva, 2006). This subjective characterization of the melody makes a universal definition of melody impractical. In the context of this work we define the melody as the pitch contour of the lead or dominant musical instrument in the song. All music which has one pitched instrument playing in the foreground with background accompaniment falls under the purview of this definition. This obviously excludes any purely rhythmic performances from our consideration. We assume that the lead instrument does not change over the course of a song. Since most popular music is vocal music we further limit the scope of this thesis by only considering music in which the lead musical instrument is the singing voice. Although the final algorithm designed in this thesis could be applied to other forms of music in which the lead instrument is not the voice e.g. lead saxophone in jazz, at each stage in our design we have attempted to integrate voice-specific features and functionality.

In related literature we often come across the term ‘predominant-F0’. This is defined as the pitch of a locally dominant source. When the melodic source/voice is present then the melody and the predominant-F0 are one and the same. When the lead instrument carrying the melody is not present then it is often the case that some other pitched instrument comes to the foreground, in which case its pitch contour becomes the predominant-F0 at that time. So while the melody is the pitch contour of only a single lead instrument the predominant-F0 is the pitch contour of the locally dominant pitched instrument.

*Harmonics/Partials/Sinusoids*

For a periodic sound the spectrum of the signal will usually consist of tonal or sinusoidal components at the fundamental frequency (F0) and its multiples. The components at multiples of the F0 are called harmonics. The tonal component at F0 is called the 1st harmonic, the tonal component at 2F0 is called the 2nd harmonic and so on. Throughout this thesis we have used the terms harmonics, partials and sinusoids interchangeably to indicate the tonal components of the spectra of a periodic sound. Note that it is also possible to have pure-tone real signals i.e. having only a single harmonic at F0, such as some flute or whistle sounds.

*Semitones and Cents*

The musical pitch space is logarithmic in nature. This is best described by Figure 2.1, which shows a typical organization of keys on a piano. The musical distance between two keys in adjacent different pitch ranges (e.g. C to C) is called an octave. In the linear scale the octave is represented by a doubling of frequency. In equal temperament tuning, which is a universal tuning system, an octave is divided into 12 equal intervals each of which is called a semitone (ST). 1 ST is the distance between any two adjacent keys on the piano keyboard. A finer resolution of the musical pitch space results in each semitone being divided equally into 100 divisions called cents. An octave is then equal to 1200 cents. Given a reference frequency $F_{ref}$ in Hz, any frequency $f_{Hz}$ can be transformed to the cents by

$$f_{cent} = 1200\left(\log_2\left(\frac{f_{Hz}}{F_{ref}}\right) - 0.25\right) \tag{2.1}$$

$F_{ref}$ is usually fixed to 13.75 Hz. This low value is 5 octaves below the standard tuning frequency of A-440Hz that is used in Western music and always results in a positive value of $f_{cent}$ .

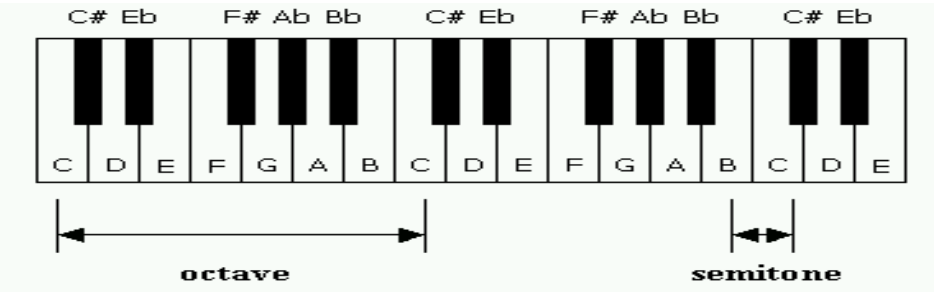


Figure 2.1: The musical scale

## 2.2 Overview of Indian Music

Music has been, and will continue to be, an integral part of Indian culture. In fact the Indian connection with music dates as far back as Vedic times (2000 BC). In a recent survey (Synovate, 2010) it was found that 82 % of Indian youth say that music is a very important part of their lives and their favorite artists were always local (not international) musicians. 62 % of the youth listened to music via their mobile phones. In the present age of increasingly penetrative broadband wireless connectivity it can be expected that the Indian music consumer will expect to have increased and easy access to large amounts of digital music.

Indian music can be categorized into multiple genres. Most audio content providers organize the genres geographically for ease of access. From a musicological perspective Indian music can be broadly categorized into three sub-genres – Classical, Popular (film) and Folk. Classical and Popular music are at two extremes of Indian music spectrum with the former being deeply rooted in musical tradition and the latter often attempting to break away from it outright. Although not entirely correct, here all music that is in between these two extremes, such as Bhajans, Qawwali & Shabad (devotional music), Ghazals (Urdu poetic songs), Dadra (semi-classical Indian songs) and Kirtan (Hindu musical chants), to name a few, have been put under the folk music category. Popular music can be categorized in multiple ways – by period (80's 90's), by geography (state-wise), by functionality (romantic, patriotic, wedding etc.) or by artist (singer). Classical Indian music can be divided into two major genres – North Indian (Hindustani) and South Indian (Carnatic) classical music. Each of these two genres has its own distinct underlying signal characteristics in terms of singing/playing style and instrumentation.

Indian music differs from Western music in several aspects such as playing/singing style and instrumentation, but all these differences stem from the undeniable fact that, irrespective of genre, the melody is the heart and soul of the song in all Indian music. The definition of melody here subsumes the rhythmical aspect as well since the melody not only contains pitch/interval information but also musical-note duration and location in time (rhythm information). In the context of our study the melodic line is always carried by a human singer (vocalist), which in any case is the more popular form of music. We next discuss the different contrasting aspects of Indian and Western music that arise from the Indian fixation on melody. Since much of Indian music is closely related to Indian classical music, we focus on the characteristics of this type of music in our discussion. Although there may be several musicological and philosophical contrasts between the two cultures, we only

discuss those aspects that have some bearing on the underlying signal characteristics and merit consideration from the perspective of the melody extraction problem.

1. *Harmony and Counterpoint*: Indian music, with its focus on the main melody, has no foundation of harmony and counterpoint that are vital to Western music. This does not however reduce the presence of multiple pitched sources in Indian music, as will be described in Section 2.4.1. The concept of harmony in music is the simultaneous playing/singing of multiple melodic lines that have a pleasing aural effect. In the barbershop, choir and gospel genres of Western music the practice of harmony is essential. Another western music concept 'counterpoint' involves the playing/singing of musical lines that sound very different and move independently from each other but sound harmonious when played simultaneously. Unlike harmony, counterpoint focuses on melodic interaction of multiple simultaneous melodic lines—only secondarily on the harmonies produced by that interaction (Wikipedia). Counterpoint is primarily used in Western classical music.
2. *Orchestration*: Except for contemporary popular music, heavy orchestration has never been a part of traditional Indian music culture. This can be contrasted with some genres of Western music, such as classical, rock, jazz and opera, where although the melody is important, the orchestration is also of equal, if not greater, importance. The importance given to the vocalist in Indian music can be judged by the number of reality singing contests on Indian television. All the top 5 Hindi general entertainment channels (GECs) contain such shows in varying formats in any given season. In addition regional television channels too broadcast their own singing reality shows. This can be contrasted with the absence of any such show for musical instrumental skill or the presence of a single televised competition for music bands. From a social perspective, such shows have brought about resurgence in Indian classical music learning, since having a classical background is often deemed as one of the essential qualities that lead to 'good' singing.

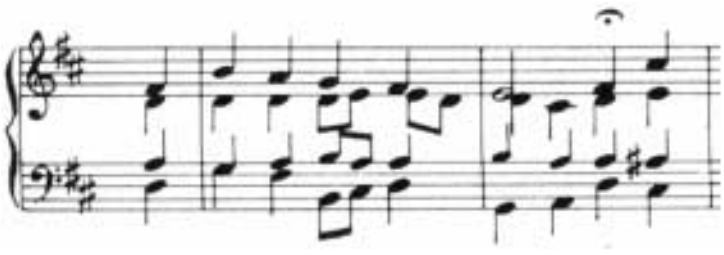

Figure 2.2: A Western musical score

3. *Improvisation/Lack of Notation*: In Western classical music, a composer first composes the music and puts it in notation; later the musicians play the music under the guidance of a conductor. There is not much room for improvisation, and the value of a performance lies in the uniformity and the pre-determined conduct of tone and speed of music. In an Indian musical performance, while the grammar of melody and rhythm is fixed, the skill and ingenuity of the musician lies in his improvisation and creativity. In a sense, Indian classical music can be likened to jazz music where the focus is on improvisation. So in Indian classical music, the composer and the performer are the same person. This focus on improvisation leads to every classical performance being unique, even if performed by the same artist at different times. An important outcome of this aspect of Indian music is the lack of a universally accepted notation system in theory and practice.
4. *Ornamentation/Microtones*: The emphasis on the melody in Indian music lead to the advanced development of subtle enhancements to the melodic line. In particular, the use of microtonal variations and ornamentation are often used to embellish a performance. Over the course of an Indian vocal performance, as the tempo increases, these embellishments become more pronounced and frequent.
5. *Tuning*: The importance given to the vocalist also manifests in the choice of the tonic or reference frequency to which all the other instruments in the performance are tuned. In Western music, this tuning frequency often has a standardized value of A 440 Hz. In Indian classical music, the vocalist chooses the tonic according to his/her comfort level for each performance, and the accompanying instruments are tuned to this tonic.
6. *Rap and Hip-hop*: In contemporary Western music the genres of Rap and Hip-hop have found widespread popularity. In both of these genres often the artist speaks lyrically, in rhyme and verse, generally to an instrumental or synthesized beat. However, Indian artists (singers) have not embraced such singing styles since there is no melodic component present. Although such music is internationally available, the majority of Indian listeners still prefer music with a dominant melodic component.

## 2.3 Applications of Melody Extraction in Indian Music

### 2.3.1 Music Transcription

One discipline that can benefit from melody extraction research is musicology. Music transcription is an essential step in musicological studies of non-Western music. Music transcription, as defined by Klapuri (Klapuri & Davy, 2006), is the process of analyzing an

acoustic musical signal so as to write down the parameters of the sounds that constitute the piece of music in question. Simply put, it is the process of going from an audio signal to a meaningful symbolic notation. From a western classical music perspective, performers use a traditional, universally accepted format of notation called a musical score. In a musical score, as shown in Figure 2.2, individual note symbols are used to indicate pitch (on a vertical scale), onset time (on a horizontal scale) and duration of individual notes (type of symbol). Alternate music representations that may be the output of a music transcription system are chord symbols, preferred by guitarists, or MIDI (Musical Instrument Digital Interface) files, which is a standard interface for exchanging performance data and parameters between electronic musical devices.

In some non-western music traditions, such as Indian classical music, the use of a written notation system has not been accepted as musicians of these traditions feel such music is primarily based on improvisation, where the composer and performer is the same person, and written notation is unequal to the task of representing the finer nuances of the musical performance. Another reason for the popular rejection of notation is the rift between Indian classical music theorists and practicing musicians because the former believe that contemporary music practice does not conform to the theories found in the historical treatises (Jairazbhoy, 1999) . This non-conformation to theoretical ideals was also brought to light by the study of Subramanian (2002) on different pitch ornaments used by different singers in Carnatic music. Van der Meer & Rao (1998) have proposed an alternative to a full transcription of Hindustani music in their system AUTRIM (Automated Transcription System for Indian Music), which displays the pitch contour of the performing artist integrated within a musically relevant pitch and timing framework with appropriate labels. A snapshot of their transcription display layout is shown in Figure 2.3.

### 2.3.2 Use of Indian Musical Information as Music Metadata

Music search and retrieval algorithms often rely on the musically meaningful metadata associated with music files, such as genre, artist, album and duration. However cultural musical information could also result in meaningful culture-specific metadata. In the context of Indian music the musical concept of *raga* is of great importance. A raga can be regarded as a tonal framework for composition and improvisation; a dynamic musical entity with a unique form, embodying a unique musical idea (Bor, Rao, & van der Meer, 1999). Apart from musical scale, there are features particular to each raga as the order and hierarchy of its tones,

their manner of intonation and ornamentation, their relative strength and duration, and specific approach. Two ragas having identical scale are differentiated by virtue of these musical characteristics. Each musical composition is linked to a specific raga. There are hundreds of known ragas popularly used in contemporary vocal performances.

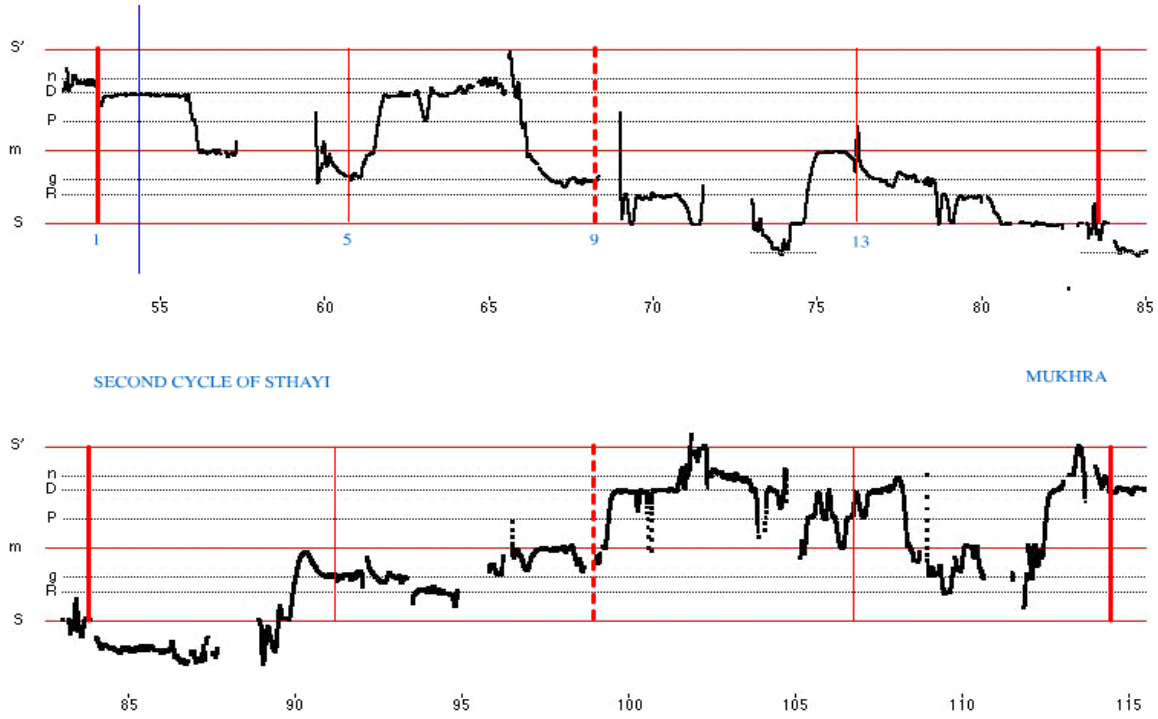


Figure 2.3: A snapshot of the AUTRIM output for an excerpt of *raag Baageshri* by Kishori Amonkar. The horizontal red lines are positioned at the lower and upper tonic (Sa), which are the primary tonal centers and at the fourth, which is a secondary tonal center. Other dotted horizontal lines are positioned at the notes used in the given *raag*. The black numbers indicate time (sec). The vertical red lines are located at important beats in the rhythm pattern (*taal)*. The blue numbers indicate the beat number in this 16 beat (*Tin taal)* framework. The location of the thin vertical blue line indicates the instantaneous position of the cursor during song playback. The solid black curves are the extracted melodic contours. (*Used with permission of the NCPA)*

## 2.4 Melody Extraction Problem Complexity: Indian Music

Here we describe some aspects of Indian music that have a bearing on the complexity of the melody extraction problem. We look at issues pertaining to the signal characteristics and to data collection and evaluation. Finally we present an example of applying a monophonic pitch detection algorithm (PDA) to melody extraction in Indian classical music. Most of the examples described in this section are taken from Hindustani classical vocal performances. However the scope of the signal characteristics extends to the Carnatic, folk and film sub-

genres of Indian music and also to other non-western music traditions such as Greek Rembetiko, Arabic and Turkish music.

### 2.4.1 Signal Characteristics

A typical Indian classical vocal performance consists of the following musical components - the singer, percussion, drone and secondary melodic instrument. Most Indian classical vocal performances are divided into three segments. They usually start with an extended session of slow non-metered singing (without rhythmic accompaniment) where the singer explores the musical space of the chosen raga using long vowel utterances and distinct phrases and patterns. This is followed by a medium tempo lyrical song-type segment of the performance. The final phase of the performance usually is in a faster tempo where the singer makes use of rapid and emphasized ornamentation manifested as pitch modulations.

We next describe the signal characteristics of each of the different musical components in an Indian classical vocal performance. It will be shown that although the orchestration is limited in Indian classical music, there may be upto seven simultaneous distinct pitches sounds present.

#### *2.4.1.1 Singing*

In singing, most pitched (i.e. voiced) utterances are intentionally lengthened at the expense of unvoiced utterances resulting in contiguous smooth pitch segments. Most PDAs make use of knowledge of temporal dynamics of voice-pitch in order to apply some pitch continuity or smoothness constraints to enhance their performance. Normal speech has a varying F0, since even the glottal source excitation is not perfectly periodic. However, in singing the presence of pitch ornamentation requires consideration in the analysis of pitch dynamics.

In western singing such ornamentation is often manifested as vibrato i.e. periodic modulation of the fundamental frequency. Vibrato can be described by two parameters namely, the rate of vibrato (the number of oscillations occurring per second) and the extent of vibrato (the depth of modulation as a percentage of the average frequency). It was found that the average rate of vibrato was 6.6 oscillations per second (Hz) and the average extent was 48 cents (Sundberg, 1987).

In Indian classical singing, such as Hindustani and Carnatic music, a variety of ornamentations apart from vibrato are used. Often these are found to be very rapid and large. One such example of rapid pitch modulations is shown in Figure 2.4. Here successive

fundamental periods of an excerpt of a song by a well-known Hindustani vocalist were noted by close visual inspection of the waveform. The point of measurement was taken to be the zero crossing before the maximum peak within each pitch period. It was observed that at the fastest change the artist changed his F0 by about 6 semitones in 72 milliseconds. Such a large range of possible pitch dynamics must be addressed effectively when designing pitch continuity constraints.

These pitch modulations, clearly perceived and recognized by experienced listeners, serve an important aesthetic function within the melodic contour and therefore need to be captured accurately. Further, unlike Western music, where the scale steps are fixed and grounded in the tempered scale, the location of the scale steps in Indian classical music is variable for different singers and also over different performances by the same singer. This is because at the start of every performance, the singer tunes the tonic location according to his/her comfort level. The exact intonation of a specific scale step with specified tonic may also change with the raga, although this is relatively infrequent in present day music (Belle, Rao, & Joshi, 2009). In any case the non-discrete, pitch-continuous nature of the musical performance disallows the use of any prior information with respect to the frequency location of musical notes in the design of a melody extraction system.

In terms of acoustic features, there is a large variability in the spectral distribution of energy across singers, perceived as differences in timbre. However, the locations of significant voice harmonics in the spectrum rarely cross 7 kHz.

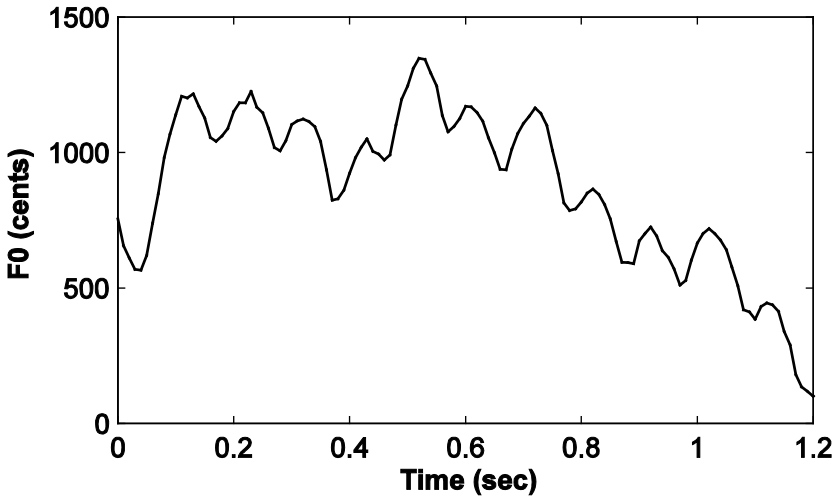


Figure 2.4: F0 contour, extracted by manual measurement, from an excerpt of an Indian classical vocal performance, depicting large and rapid pitch modulations.

#### *2.4.1.2 Percussion*

One of the most popular percussive instruments used in Hindustani music is the *tabla*. It consists of a pair of drums, one large bass drum, the *bayan*, and a smaller treble drum, the *dayan*. Tabla percussion consists of a variety of strokes played by hand, often played in rapid succession, each labeled with a mnemonic. Two broad classes, in terms of acoustic characteristics, are: 1. tonal strokes that decay slowly and have a near-harmonic spectral structure (thus eliciting a pitch percept) and 2. impulsive strokes that decay rapidly and have a noisy spectral structure. Both the bass and treble tablas are capable of producing both, tonal and noisy, types of strokes.

The acoustic characteristics of various tabla strokes were studied from Parag Chordia's database (http://ccrma.stanford.edu/~pchordia/tablaStrokes/). It was found that while all the impulsive strokes had similar acoustic characteristics, there was a large variability in those of the different tonal strokes. The acoustic features of three typical tonal strokes, associated with the mnemonics *ghe*, *na* and *tun*, are compared by means of narrowband spectrograms in Figure 2.5. Ghe is produced by the bayan while na and tun are produced by the dayan. All the three strokes, soon after onset, exhibit harmonics that lie in the same frequency range as those of the singing voice. The strokes ghe and tun have a more gradual decay than na, which decays quite rapidly, albeit still much slower than any of the impulsive strokes. The spectrograms of ghe and na exhibit up to five dominant harmonics for a brief period after the onset. Tun, on the other hand, is dominated by a single harmonic, giving it an almost sinusoidal timbre. Further, as ghe is produced by the bayan, its harmonics all lie in a low frequency range resulting in a very low pitch percept. In contrast, the pitch of the dayan is tuned to the tonic of the singer prior to the performance, which is reflected in the harmonics of its strokes occupying a higher region in the spectrum. Interestingly, the spectra of tun and na are almost complementary, with tun being dominated by the F0 and na missing the F0. If we consider the signal of interest to be the singing voice, the local Signal-to-Accompaniment Ratio (SAR) can dip as low -10 dB around a tabla stroke onset. In practice the pair of tablas is often struck simultaneously and therefore spectral components of the bass and treble strokes are present at the same time.

#### *2.4.1.3 Drone*

In both the Hindustani and Carnatic sub-genres of Indian classical music there is a drone present throughout the musical performance. The drone is provided by an instrument called

the *tanpura*, which is an overtone-rich stringed instrument, mainly pitched at the tonic. Its purpose is to provide a reference point in the musical framework to the singer. It usually has four strings, three pitched at the tonic (upper and lower octave) and one pitched at either the fifth or the fourth. Although the strings of the tanpura are plucked sequentially, it is a polyphonic instrument, since each note has a very long decay and the note of the previousstring(s) has not died down before the next string is plucked. The tanpura sound is audibly quite prominent relative to the singer's voice. However, this prominence can be attributed to the fact that its energy is spread over a very large number of partials throughout the spectrum up to 10 kHz as can be seen in Figure 2.6. This leads to frequency bands dominated entirely by tanpura partials, thus enhancing its perceived loudness. The SARs for the tanpura with respect to the voice usually range from 20 to 30 dB but can dip down to 10 dB in some cases.

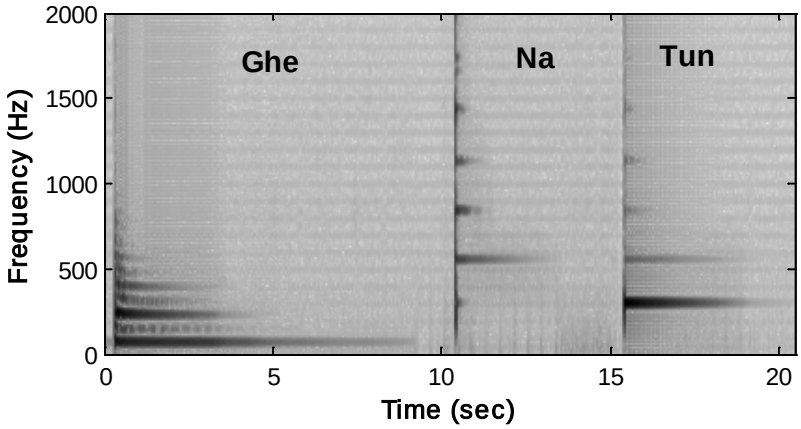


Figure 2.5: Spectrogram of three typical tonal strokes (Ghe, Na and Tun). The dayan is tuned to F0 = 283 Hz.

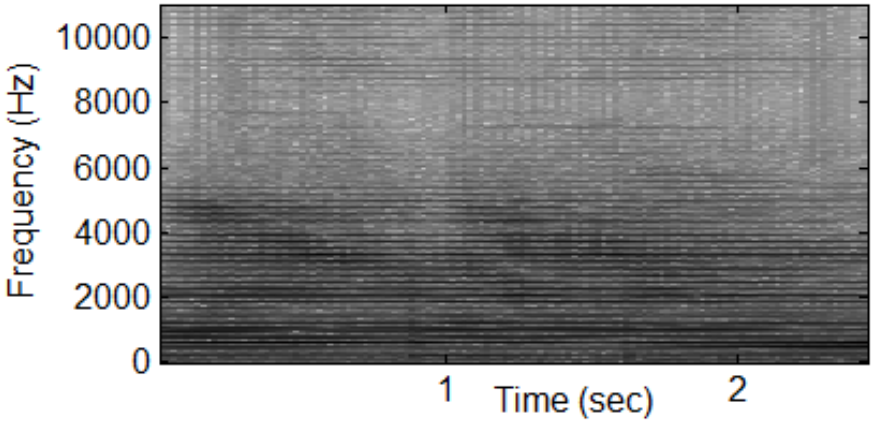


Figure 2.6: Spectrogram of a single cycle of tanpura playing (4 string plucks)

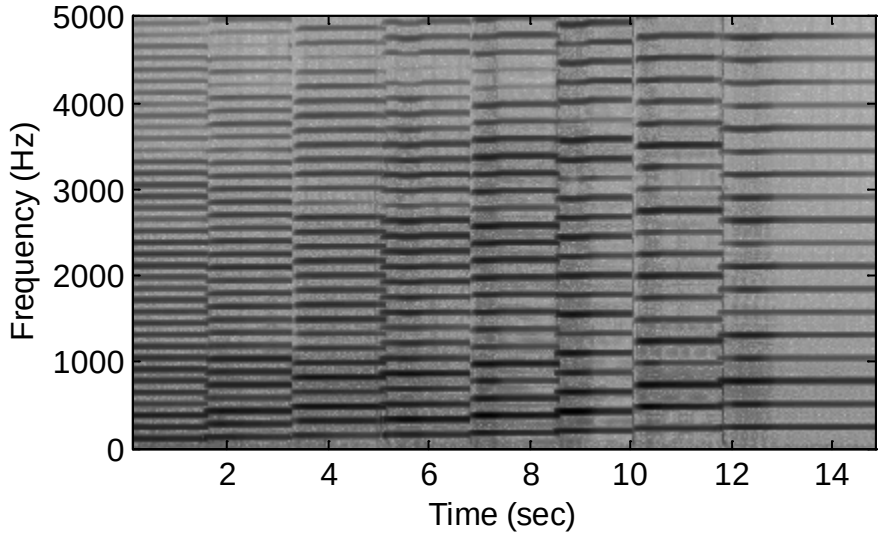


Figure 2.7: Spectrogram of a sequence of harmonium notes

#### *2.4.1.4 Secondary Melodic Instrument*

In Indian classical music, there are several instances of a loud pitched accompanying instrument that does not provide a chord-type accompaniment but instead often plays a melodic line, usually a delayed version of the singer's melody. We term these instruments as secondary melodic instruments. In Hindustani and Carnatic music this instrument is played in a manner such that it reinforces or attempts to follow the singer's melody. In the absence of a written score, because of the improvisatory nature of the Indian classical performance, the instrumentalist attempts to follow the singer's pitch contour as best he/she can. In some cases this secondary melodic line is played simultaneously with the voice, with a small delay, and in other cases it repeats the phrase of the singer's melody as a solo rendition after the sung phrase is completed. Along with the problem of identifying the voice-pitch contour as the predominant-F0 in the presence of a concurrent secondary melody it is also required to be able to discriminate between isolated voice and instrumental regions of the predominant-F0 contour. The popular secondary melodic instruments used in Carnatic and Hindustani music are the violin and *harmonium* respectively. The harmonium is a keyboard instrument similar to a reed organ. Sound is produced by air, supplied by hand-operated bellows, being blown through sets of free reeds, resulting in a sound similar to that of an accordion. The *harmonium* falls under the class of instruments that are incapable of continuous pitch variations and can only make discrete pitch jumps. Like the accordion the harmonium is a polyphonic instrument, capable of producing multiple simultaneously sounding pitches. This can be seen in Figure 2.7, which shows a sequence of harmonium notes extracted from single-channel

recording of the harmonium being played during a Hindustani vocal performance. Here we can see that multiple notes are sounding simultaneously around note onsets. It can also be seen that the harmonics of the harmonium are significantly strong even at higher frequencies. The SARs for the secondary melodic instruments with respect to the voice usually range from 0 to 10 dB.

### 2.4.2 Issues Pertaining to Audio Data Collection and Evaluation

One of the major stumbling blocks in melody extraction research is the evaluation of the designed algorithms. Ideally for evaluating the accuracy of a melody extraction system we need to have the ground truth values of the sung pitched utterances. These are hard to come-by unless multi-track recordings of the songs have been made and the single-track voice channel has been processed separately to extract these ground-truth voice pitch values. Alternatively the use of karaoke recordings also facilitates the separate recording of the voice track, but mixing the vocal and instrumental tracks should be done as realistically as possible. In the event that such ground-truth is unavailable, the underlying notation or score can also be used for evaluation, assuming there is a high-performance consistent algorithm available for going from the sung voice pitch contour to a musical note sequence.

The Indian music scenario suffers from severe drawbacks on the melody extraction evaluation front. In a classical Indian music performance much depends on the interaction between the instrumentalists and the vocalists. In fact a popular component of the performance is a mischievous one-upmanship type of interaction between the vocalist, the percussionist and the secondary melodic instrument player. For this interaction all the musicians need to be in close physical proximity of each other. So obtaining multi-track clean recordings is a problem. The improvisatory nature of the performance negates the possibility of creating karaoke tracks for sing-along and also nullifies the use of notation. In the event that such notation was available and a vocalist agreed to follow it, the inherent and extensive use of pitch embellishments and inflexions would degrade the performance of the pitch-contour-to-note algorithm, thereby negatively affecting the reliability of the evaluation.

### 2.4.3 Case Study: Using a Monophonic PDA for Hindustani Vocal Music

To demonstrate the need for separate pitch detection methods for polyphonic music, let us consider Figure 2.8, which displays a narrow-band spectrogram of a relatively simple polyphonic clip with a superimposed F0 contour (white solid curve) estimated by a modified

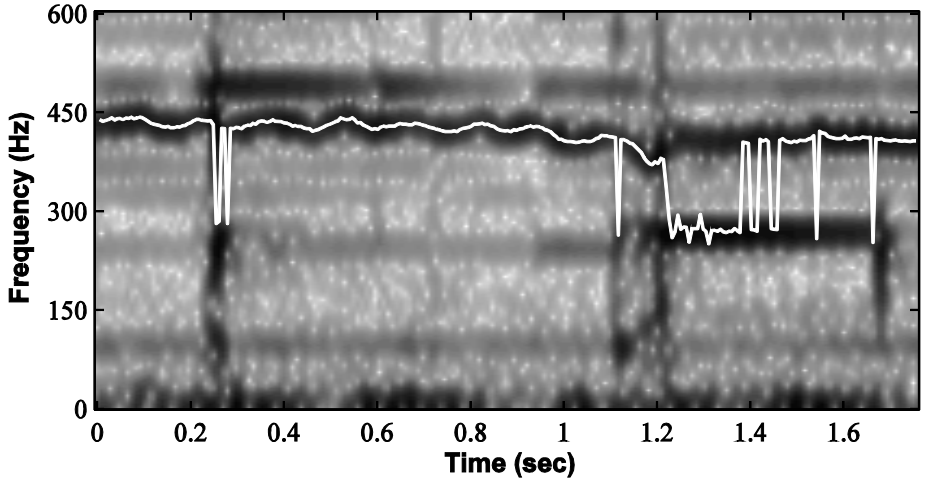


Figure 2.8: Pitch contour *(*white line*)* as detected by a modified ACF PDA superimposed on the zoomed in spectrogram of a segment of Hindustani music with a female voice, drone and intermittent tabla strokes.

autocorrelation function (ACF) PDA (Boersma, 1993) with recommended parameter settings. This clip is a short excerpt from a female Hindustani vocal performance in which the voice is accompanied by a soft drone and a percussive instrument (table). In this segment, the sequence of tabla strokes is as follows: impulsive stroke (0.22 sec), impulsive stroke (1.15 sec), tonal stroke (1.2-1.7 sec), and impulsive stroke (1.7 sec). The impulsive strokes appear as narrow, vertical, dark bands. The tonal stroke is marked by the presence of a dark (high intensity) horizontal band around 290 Hz, which corresponds to its F0. The other horizontal bands correspond to drone partials which, are in general of much lower amplitude than the voice or tabla harmonics when present.

Clearly, this particular PDA is able to accurately track the F0 of the voice in the presence of the drone, as indicated by the region where the pitch contour overlaps with the dark band in the spectrogram corresponding to the voice F0 (between 0.4 and 1 seconds). The presence of percussion, however, in the form of the intermittent tabla strokes, causes significant noise-like degradation of the pitch estimates, as seen in Figure 2.8. While the errors due to the impulsive strokes are localized, the tonal stroke causes errors that are spread over a longer segment possibly obscuring important pitch variations. The pitch errors are not just voice octave errors but interference errors i.e. when the F0 estimated is actually the tabla stroke F0, indicated by the lower dark band present between 1.2 and 1.7 sec.

The polyphonic problem described above is a relatively simple case and can be compounded by the presence of an increasing number of pitched and percussive musical

sources with varying signal characteristics. Clearly the problem is non-trivial and merits investigation.

## 2.5 Datasets used for Evaluation

Here we describe all the common datasets we have used at different stages of our evaluations. In case any data has been used to test one specific module it will be described within the chapter discussing that specific module. For the evaluations presented in the subsequent chapters the specific data used for each experiment, which is a subset of the data described in this section, will be described within the context of that experiment. Note that all data used in our experiments are in 22.05 kHz 16-bit Mono (.wav) format.

We present the data used for evaluation of the predominant-F0 extraction system and the singing voice detector system separately. The difficulty in accessing multi-track or karaoke recordings for predominant-F0 extraction evaluation has limited the size of any publicly available datasets for these evaluations till date. The singing voice detection (SVD) problem, on the other hand, requires sung phrase onset and offset locations as ground-truths. These can be manually marked relatively quickly on polyphonic audio directly. In addition the algorithms used for the SVD problem typically rely on a training-testing approach and so the corresponding data used is relatively large in size. The data used in the Predominant-F0 extraction evaluations is often used in the SVD evaluations as well; however the other way around is not possible for reasons mentioned above.

A list of all the different datasets available for the evaluation of the predominant-F0 extraction and singing voice detection systems is given in Table 2.1. Here we assign a label to each dataset and refer to these labels in subsequent chapters.

### 2.5.1 Predominant-F0 Extraction Datasets

All the dataset collections in this category either provide the ground-truth voice-pitch values or the monophonic (clean) voice audio file for processing. The ground-truth pitch values are available at intervals of 10 ms. In the case of the provision of a monophonic voice file, ground-truth voice pitches at 10 ms intervals are extracted using the YIN algorithm (de Cheveigne & Kawahara, YIN, a fundamental frequency estimator for speech and music, 2002) followed by a dynamic programming (DP)-based post-processing stage (Ney, 1983), and are further manually examined and hand-corrected in case of octave or voicing errors. For predominant-F0 evaluation only the pitch values during the sung segments are considered.

Table 2.1: List of different audio datasets used in the previous evaluations. (P – indicates ground-truth values were provided, E – indicates Ground-truth values were extracted by us)

| Label | Dataset |
|---|---|
| *Predominant-F0 Extraction Evaluation* | |
| PF0-ICM | Hindustani music from NCPA & SRA (E) |
| PF0-LiWang | Western music data used by Li & Wang (2007) (E) |
| PF0-ADC04 | Dataset used in the ADC'04 evaluation (P) |
| PF0-MIREX05-T | MIREX'05 Training data (P) |
| PF0-MIREX05-S | MIREX'05 Secret data |
| PF0-MIREX08 | Indian classical music data used in the MIREX'08 evaluation (E) |
| PF0-MIREX09 | MIR1k data of Chinese karaoke recordings (E) |
| PF0-Bolly | Recordings of Indian mainstream film music (E) |
| *Singing Voice Detection Evaluation* | |
| SVD-West | Western music data used by Ramona, Richard & David (2008) (P) |
| SVD-Greek | Greek Rembetiko data used by Markaki, Holzapfel & Stylianou (2008) (P) |
| SVD-Bolly | Examples from Indian mainstream film music (E) |
| SVD-Hind | Examples from Hindustani (north Indian classical) music (E) |
| SVD-Carn | Examples from Carnatic (south Indian classical) music (E) |

#### *2.5.1.1 Indian Classical Music*

As mentioned before obtaining multi-track data of an Indian classical vocal performance is very difficult because of the improvisatory nature of the performance. We were aided in this data collection by The National Centre for Performing Arts (NCPA), Mumbai, which is one of the leading Indian music performance centers. For their musicological analysis they had specially created recordings in which the voice, percussion, drone and secondary melodic audio signals were recorded in different channels. To ensure time-synchrony and acoustic isolation for each instrument the performing artists were spread out on the same stage with considerable distance between them and recorded on separate channels simultaneously.

We were given access to a limited quantity of this data. Specifically two 1-minute excerpts from each of two different vocal performances (one male singer and one female singer) were used. One excerpt is taken from the start of the performance where the tempo is slow and the other excerpt is taken towards the end of the performance where the tempo is fast and rapid modulations are present in the voice track. For each of these excerpts we had 4 single-channel tracks, one each for the voice, percussion (tabla), drone (tanpura) and secondary melodic instrument (harmonium).

For the subsequent evaluations of the predominant F0 extraction system, for each voice excerpt, its time-synchronized *tabla* counterpart was added at an audibly acceptable, global signal-to-interference ratio (SIR) of 5 dB. Further the time-synchronized *tanpura* is

added to the voice-*tabla* mixture such that the SIR for the voice with respect to the *tanpura* is 20 dB. Last, the case of a loud secondary melodic instrument is considered by adding the *harmonium* track to the voice-*tabla*-*tanpura* mixture at a voice-*harmonium* SIR of 5 dB. These signals are referred to as V (voice), VT (voice+tabla), VTT (voice+tabla+tanpura) and VTTH (voice+table+tanpura+harmonium) signals.

The Sangeet Research Academy (SRA), Kolkata, is another premier Hindustani music educational and performance center, who have shared some recordings of Hindustani music with us. This data consists of excerpts of only a single performance by a female singer. Also this data is not multi-track, i.e. does not have each instrument on a separate track, but the final polyphonic mix and the clean voice track were made available to us, so that ground-truth values can be obtained. In this particular polyphonic mix the harmonium is quite loud relative to the voice though we do not have access to the actual SAR values used. The excerpts used here correspond to the normal and fast tempo sung portions in the performance. The duration of the data is 3 and a half minutes.

#### *2.5.1.2 MIREX Datasets*

The MIREX audio melody extraction contests use different datasets for evaluation. These datasets are contributed by members of the MIR community. These datasets are listed below. Apart from the MIREX'05-Secret dataset, we have access to the audio files and ground-truth pitch values for all the remaining datasets.

1) *ADC'04* – This dataset was provided by the Music Technology Group of the Pompeu Fabra University. It contained a diverse set of 20 polyphonic musical audio pieces, each of 20 seconds approximate duration along with their corresponding ground-truth pitch values. These audio clips were taken from the different western genres as shown below.
    - 4 items consisting of a MIDI synthesized polyphonic sound with a predominant voice (MIDI)
    - 4 items of jazz saxophone melodic phrases plus background music (SAX)
    - 4 items generated using a singing voice synthesizer plus background music (SYNTH)
    - 4 items of opera singing, two with a male and another two with a female lead singer(OPERA)
    - 4 items of pop music with male singing voice (POP)

2) *MIREX'05-Training* - For the MIREX 2005 Melody Extraction Contest, the training dataset consisted of 13 excerpts of 20-40 seconds length each from the following Western genres: Rock, R&B, Pop, Jazz, along with their ground-truth values. There are
    - 4 items consisting of a MIDI synthesized polyphonic sound with a predominant voice (MIDI)
    - 9 items of music of different genres with singing voice (VOICE)
3) *MIREX'05-Secret* – This dataset consists of 25 phrase excerpts of 10-40 sec from the following genres: Rock, R&B, Pop, Jazz, Solo classical piano. No detailed information about this dataset was made available.
4) *MIREX'08* - 4 excerpts of 1 min. from the Hindustani vocal performances mentioned in Section 2.5.1.1, instruments: singing voice (male, female), tanpura (Indian instrument, perpetual background drone), harmonium (secondary melodic instrument) and tablas (pitched percussions). There are two different mixtures of each of the 4 excerpts with differing amounts of accompaniment for a total of 8 audio clips. These are basically the VTT and VTTH audio clips contributed by us, created from the NCPA multi-track data.
5) *MIR1k* - 374 Karaoke recordings of Chinese songs. Each recording is mixed at three different levels of Signal-to-Accompaniment Ratio {-5dB, 0dB, +5 dB} for a total of 1122 audio clips. Instruments: singing voice (male, female), synthetic accompaniment. These were provided by Hsu & Jang (2010).

#### *2.5.1.3 Li & Wang dataset*

This dataset, provided by Li & Wang, consists of the same audio examples as used by them for the evaluation of their predominant F0 extraction system in (Li & Wang, 2007). This set consists of 25 clips from 10 songs that include both male and female singers. 5 of these songs belong to the rock genre and the other 5 belong to the country music genre. The clean vocal and accompaniment tracks of these songs were extracted from karaoke CDs using de-multiplexing software. The total duration of this dataset is 97.5 seconds.

#### *2.5.1.4 Bollywood dataset*

This dataset consists of 30 clips each of popular Hindi mainstream film music sung by male and female singers respectively of average duration 31 sec. The total duration of the male and female datasets is 16 min 3 sec and 15 min 30 sec respectively, of which approximately 60 %

contains sung utterances. The F0 range of the male data is [80 375 Hz] and that of the female data is [180 785 Hz]. The ground-truth pitches and vocal/non-vocal regions were carefully extracted using the semi-automatic interface of Chapter 9.

### 2.5.2 Singing Voice Detection Datasets

For the SVD problem we have access to data from five different genres – Western, Greek Rembetiko, Bollywood, Hindustani and Carnatic. In the case of the Western and Greek genres we have obtained datasets previously reported in the literature where ground-truth sung-phrase annotations were also made available to us. In the case of un-availability of ground-truth annotations we have annotated the audio clips with sung-phrase onset and offset locations using PRAAT (Boersma & Weenink, 2005).

- *Western* – This is the same dataset as used by Ramona, Richard & David (2008). This consists of a set of 93 songs with Creative Commons license from a free music sharing website, which constitute a total of about 6 hours of music. The files are all from different artists and represent various genres from mainstream commercial music. Each file has been manually annotated by the same person with high precision. The audio files are originally coded in stereo Vorbis OGG 44.1kHz with 112kb/s bitrate, but have been down-sampled to 22 kHz 16-bit Mono (.wav) format.
- *Greek* – This data set contains historical and recent recordings of Greek Rembetiko music and was used by Markaki, Holzapfel & Stylianou (2008). It consists of 84 songs from 21 singers. This dataset was divided into a training set of 63 songs and a testing set of 21 songs, one from each singer.
- *Bollywood* – This dataset consists of excerpts from 13 songs selected from main-stream Indian film music, each performed by a different artist. The total duration of this data is about 18 minutes.
- *Hindustani* – This dataset contains 9 vocal performances, each performed by a different artist (4 female and 5 male), of the Khyal genre of north Indian classical music. We have segmented the performance into excerpts from the start, middle and end of recordings based on different signal characteristics at each stage. The total duration of this dataset is 27 minutes. The recordings are comprised of sounds generated by a single vocalist, a pair of *tabla*s, a *tanpura* and in some cases, a secondary melodic instrument.

- *Carnatic* – This dataset contains excerpts from 6 vocal performances, each performed by a different artist. The total duration of this dataset is about 13 minutes.

## 2.6 Evaluation Metrics

Although some of our system design considerations are genre-specific, in our investigations we utilize evaluation metrics previously reported in related literature (Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007) so that we can view our work in a general perspective.

### 2.6.1 Predominant-F0 Extraction

The extracted and ground-truth pitch values are first converted to a cents scale, since musical pitches in the equal temperament tuning are distributed along a logarithmic scale. Then two measures of performance are extracted

1. Pitch accuracy (PA): is the percentage of sung-frames of audio for which the extracted pitch is correct. Correctness is defined here as the extracted pitch value for a frame lying within $1/4^{th}$ tone (50 cents) of the reference pitch value in that frame.
2. Chroma accuracy (CA): This is similar to PA except that octave errors i.e. where the pitch error is an exact multiple of an octave, are ignored/forgiven.

### 2.6.2 Singing Voice Detection

For the SVD problem we compute evaluation measures borrowed from information theory. Each of these evaluation metrics can be computed separately for only vocal, only instrumental or all frames.

1. Recall: This is the ratio of correctly labeled frames to the actual number of frames with that label present.
2. Precision: This is the ratio of correctly labeled frames to the total number of frames that have been assigned that particular label by the algorithm.
3. False alarm rate: is the ratio of incorrectly labeled frames to the actual number of frames with the correct label present.

For singing voice detection, we pool the frames from all excerpts in a dataset to get an overall frame-level vocal detection performance. Because some excerpts had no unvoiced frames, averaging over the excerpts can give some misleading results.

### 2.6.3 MIREX

Consider a matrix of the per-frame voiced (Ground Truth or Detected values != 0) and unvoiced (Ground Truth or Detected values == 0) results, where the counts are:

| | | Detected | | |
|---|---|---|---|---|
| | | unvx | vx | sum |
| **Ground Truth** | unvoiced | TN | FP | GU |
| | voiced | FN | TP | GV |
| | sum | DU | DV | TO |

TP ("true positives", frames where the voicing was correctly detected) further breaks down into pitch correct and pitch incorrect, say TP = TPC + TPI.

Similarly, the ability to record pitch guesses even for frames judged as unvoiced breaks down FN ("false negatives", frames which were actually pitched but detected as unpitched) into pitch correct and pitch incorrect, say FN = FNC + FNI. In both these cases, we can also count the number of times the chroma was correct, i.e. ignoring octave errors, say $TP = TPC_{ch} + TPI_{ch}$ and $FN = FNC_{ch} + FNI_{ch}$.

The following metrics are then computed.

**Voicing Detection** is the probability that a frame which is truly voiced is labeled as voiced i.e. TP/GV (also known as "hit rate").

**Voicing False Alarm** is the probability that a frame which is not actually voiced is none the less labeled as voiced i.e. FP/GU.

Now we move on to the actual pitch detection.

**Raw Pitch Accuracy** is the probability of a correct pitch value (to within ± ¼ tone) given that the frame is indeed pitched. This includes the pitch guesses for frames that were judged unvoiced i.e. (TPC + FNC)/GV.

**Raw Chroma Accuracy** is the probability that the chroma (i.e. the note name) is correct over the voiced frames. This ignores errors where the pitch is wrong by an exact multiple of an octave (octave errors). It is $(TPC_{ch} + FNC_{ch})/GV$.

**Overall Accuracy** combines both the voicing detection and the pitch estimation to give the proportion of frames that were correctly labeled with both pitch and voicing, i.e. (TPC + TN)/TO.

# Chapter 3

## Signal Representation

An efficient signal representation stage is integral to the performance of any music signal processing system such as the multi-pitch detection module for a predominant-F0 extraction system. In previous work, the most common approach to arrive at a suitable signal representation is to first transform the time-domain audio signal into its frequency domain representation. The frequency domain not only affords better access to the perceptually important attributes of an isolated musical source (frequency component locations, timbre via spectral envelope) but also leads to improving the disjointedness of co-occurring sources due to the inherent sparseness of spectral representations for musical sources. The transformation to the frequency domain has been performed in various ways such as the short-time Fourier transform (STFT), (Cao, Li, Liu, & Yan, 2007; Klapuri, 2003; Rynnanen & Klapuri, 2008), multi-resolution STFT (Dressler, 2006; Fernandez-Cid & Casajus-Quiros, 1998; Goto, 2004), constant Q transform (Cancela P. , 2008) or using auditory filter-banks (Klapuri, 2008; Li & Wang, 2005; Paiva, Mendes, & Cardoso, 2006; Tolonen & Karjalainen, 2000).

For polyphonic music the signal components can be broadly categorized as pitched and transient sounds. It is well known that the frequency-domain analysis of a pitched musical note results in a set of harmonics at near-integer multiples of an F0. In order to enhance the pitched content of the polyphonic audio for multi-pitch extraction, often the transformed signal has been further processed to extract a sparse signal representation in the form of a set of sinusoidal partials and their parameters (frequencies, amplitudes and phases), widely referred to as sinusoidal modeling. (Dressler, 2006; Fernandez-Cid & Casajus-Quiros, 1998; Goto, 2004).

In this chapter we first review single and multi-resolution STFT based signal representations. We consider the possibility of adaptively varying the analysis parameters (in particular the window length) to better represent the singing voice partials in the context of polyphonic vocal music. To this end we describe different easily computable mathematical measures of signal sparsity for driving the window-length selection. Next we describe different methods for sinusoid detection and parameter estimation from the short-time magnitude and phase spectrum. We then evaluate the different sinusoidal representations and the suitability of different signal sparsity measures, in terms of optimality for window length selection, as determined by the measured accuracy of sinusoid detection and parameter estimation for simulated signals[1].

## 3.1 Frequency-Domain Signal Representation

In the context of melody extraction from polyphonic music the most commonly used signal representation is the short-time Fourier transform (STFT) (Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007). This involves computing the discrete Fourier transform (DFT) for successive, overlapping, windowed frames of the original signal (Allen, 1977). For a time-domain signal $x(m)$, its STFT at the analysis time instant $n$ is given by

$$X_n(k) = \sum_{m=0}^{M-1} x(m)\, w(n-m)\, e^{\frac{-j2\pi mk}{M}} \tag{3.1}$$

where $k$ is the frequency bin index, $w(n)$ is a window function of length $M$ samples. A commonly used window function in audio processing is the Hamming window given by

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right) \tag{3.2}$$

[1] The work on sparsity-driven window-length adaptation was done together with Pradeep Gaddipati.

The size of the window and time-advance between two successive windows are commonly referred to as the window-length and hop-size respectively. An efficient and commonly used method of computing the above DFT is by use of the Fast Fourier Transform (FFT) algorithm.

The bin-resolution is determined by the number of DFT points $N$ and the sampling frequency $F_s$ and is given by $\Delta f_{bin}=F_s/N$. $N$ is usually equal to the analysis window-length, however if greater bin-resolution is preferred then the analysis frame can be zero padded and a larger number of DFT points can be used. Irrespective of the number of DFT points, the frequency-resolution (the ability to distinguish between two closely spaced sinusoidal components) is determined by main-lobe width of the DFT of the window function. The frequency resolution for a Hamming window is given by $\Delta f_{freq}= 8\pi/M$ radians.

### 3.1.1 Multi-Resolution Frequency-Domain Representation

The polyphonic nature of music and the non-stationarity of the singing voice (manifested as pitch modulations) require contrastingly different analysis parameters, especially the duration of the short-time analysis window. For example, longer window-lengths serve to obtain higher frequency-resolution for individual harmonics of multiple, simultaneously present, pitched sources (Klapuri, 2003; Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007). Such long window lengths are also useful in the discriminating between logarithmically-spaced musical notes in low pitch ranges (<100 Hz). On the other hand, when the lead instrument/voice is heavily ornamented e.g. by the use of extensive vibrato or culture-specific musical ornamentation, manifested as large and rapid pitch modulations, the use of long windows usually results in a distortion of the voice harmonics, especially at higher frequencies. These contrasting conditions of polyphony and non-stationarity call for a trade-off in time-frequency resolution.

Motivated by the correspondingly larger frequency modulations seen in the higher harmonics, sometimes window durations are systematically reduced across frequency bands spanning the spectrum to obtain a multi-resolution analysis. This will maintain greater frequency resolution at lower frequencies and greater time resolution at higher frequencies. Goto (Goto, 2004) and Dressler (Dressler, 2006) have two different approaches to the above idea in the context of predominant-F0 extraction. The former makes use of a multi-rate filter bank along with a DFT computation at each rate. The latter involves the efficient computation of a DFT with different window-lengths for different frequency bands (whose limits are defined by forming groups of critical bands). The upper frequency limit considered by both is

approximately 5 kHz, beyond which significant voice harmonic content is not visible in the spectrum.

### 3.1.2 Window-Length Adaptation Using Signal Sparsity

The analysis parameters, in particular the window-length, in previous approaches to multi-resolution representations are essentially fixed in time. However signal conditions in terms of vocal and instrumental characteristics show large variations, within and across genres, which may not be suitably represented by fixed multi-resolution analysis. For example, in non-western music, such as Indian and Greek music, large and rapid pitch modulations are commonly used as are long stable notes. Further, instrumentation with significantly strong harmonic content throughout the voice spectral range (0-5kHz), such as many commonly used wood-wind instruments as the accordion and *harmonium*, may require high frequency resolution throughout the spectrum. The gamut of different underlying signal conditions merits investigating an adaptive approach.

Signal-driven window-length adaptation has been previously used in audio coding algorithms, such as MPEG I and AAC, for discriminating between stationary and transient audio segments (Painter & Spanias, 2000). However, in the context of singing voice analyses, the only common signal-driven adaptive window-length analysis has been pitch-adaptive windowing based on previous detected pitch (Kim & Hwang, 2004). This loses its relevance in polyphonic music where multiple pitched instruments co-occur. Jones (1990) used a kurtosis measure to adapt the window used in the computation of time-frequency (t-f) representations of non-stationary signals. However the evaluation was restricted to visual comparison of t-f representations of complicated signals. Goodwin (1997) used adaptive time segmentation in a signal modeling and synthesis application. The method has a very high computational cost since the window adaptation is based on minimizing the actual reconstruction error between the original and synthesized signals.

Here we investigate the use of some easily computable measures for automatically adapting window lengths to signal characteristics in the context of our application. Most of these measures have been previously proposed as indicators of signal sparsity (Hurley & Rickard, 2009). Based on the hypothesis that a sparse short-time spectrum, with its more “concentrated” components, would facilitate the detection and estimation of the signal harmonics, we apply the different sparsity measures to the task of window-length adaptation.

### *3.1.2.1 Measures of Signal Sparsity*

We review five different measures of signal sparsity– L2 norm (L2), normalized kurtosis (KU), Gini Index (GI), spectral flatness (SF) and the Hoyer measure (HO). Of these SF has been widely used for driving window switching in audio coding algorithms. For $X_n(k)$ the magnitude spectrum at time instant $n$ for frequency bin $k$ of $N$ total frequency bins, the definitions of different sparsity measures are given below

1. $\ell^2$ norm

$$\mathrm{L2} = \sqrt{\sum_k X_n^2(k)} \tag{3.3}$$

2. Normalized kurtosis

$$\mathrm{KU} = \frac{\frac{1}{N}\sum_k \left|X_n(k) - \overline{X}\right|^4}{\left(\frac{1}{N}\sum_k \left|X_n(k) - \overline{X}\right|^2\right)^2} \tag{3.4}$$

where $\overline{X}$ is the mean spectral amplitude.

3. Gini Index: The magnitude spectral coefficients $X_n(k)$ are first sorted in ascending order to give the ordered magnitude spectral coefficients $X_n^{(k)}$. The Gini Index is then given as

$$\mathrm{GI} = 1 - 2\sum_k \left(\frac{X_n^{(k)}}{\|X\|_1}\left(\frac{N-k+0.5}{N}\right)\right) \tag{3.5}$$

where $\|X\|_1$ is the $\ell^1$ norm of $X_n(k)$.

4. Hoyer measure: is a normalized version of $\frac{\ell^2}{\ell^1}$, and is defined as

$$\mathrm{HO} = \left(\sqrt{N} - \frac{\sum_k X_n(k)}{\sqrt{\sum_k X_n^2(k)}}\right)\left(\sqrt{N} - 1\right)^{-1} \tag{3.6}$$

5. Spectral flatness: has been used as a measure of tonality of a signal in perceptual audio coding (Johnston, 1988). Here we use it as an indicator of signal sparsity; the more peaky the spectrum of a signal, the more sparse it is. Spectral flatness is

defined as the ratio of geometric mean of the power spectrum to the arithmetic mean of the power spectrum, and is given as

$$\mathrm{SF} = \frac{\sqrt[N]{\prod_k X_n^2(k)}}{\frac{1}{N}\sum_k X_n^2(k)} \tag{3.7}$$

#### *3.1.2.2 Window-length Adaptation*

Each of the previously mentioned sparsity measures is individually used in a window-length adaptation scheme described next. For each frame of audio we would like to apply that window-length that maximizes signal sparsity, anticipating that this would improve sinusoid detection. For a particular analysis time instant this amounts to selecting that window length among a set {23.2, 46.4 and 92.9 ms} that maximizes the signal sparsity i.e. either maximizes the normalized kurtosis, Gini index and Hoyer sparsity measures OR minimizes the L2 and spectral flatness sparsity measures. Further, since we expect increased signal non-stationarity at higher frequencies, we compute fixed and adapted window analyses separately across three frequency bands, viz. 0–1.5 kHz, 1–3 kHz and 2.5–4 kHz.

The implementation of the adaptive window representation in our evaluation involves the initial computation of the full-band spectral representation using each of the three window lengths. Note that the analysis time instants are fixed (at window-centers) by the use of a fixed hop (10 ms). For all window lengths we use a fixed 2048 point DFT. For the 23.2 and 46.4 ms windows this involves zero-padding the windowed signal. Then for a given frequency band we compute a sparsity value from the frequency bins corresponding to the desired frequency range for each window-length representation. We select that window length that maximizes the signal sparsity for the given frequency band.

## 3.2 Sinusoidal Representation

In applications that require multiple sources in a polyphonic mixture to be well represented, such as separation of sources from polyphonic mixtures (Zhang & Zhang, 2005), the accurate and reliable detection of sinusoids and their parameters (frequencies and amplitudes) is required for each source in the mixture. The detected sinusoids can help to reveal underlying harmonic relationships and hence the pitch of each harmonic source. The amplitudes of the harmonics corresponding to each detected pitch represent the spectral envelope of the corresponding instrument, and may be used subsequently for singing voice detection or

instrument identification as well. In the processing of the singing voice in polyphony for applications such as vocal melody extraction and singing voice detection, it is essential to preserve the harmonic structure of the dominant, singing voice as faithfully as possible.

Several different approaches to sinusoid detection exist, the most popular of which are the Fourier analysis methods based on the common first step of computing the Fourier spectrum of the windowed signal. We consider Fourier based methods over alternate approaches such as subspace methods for parameter estimation (Badeau, Richard, & David, 2008), which require prior knowledge about the number of components, and non-linear least-squares based sinusoid detection, which has been shown to not work well for multi-pitch signals (Christensen, Stoica, Jakobsson, & Jensen, 2008). In order to apply Fourier analysis we assume signal stationarity within the analysis duration i.e. the audio signal within each analysis window is modeled by a set of stable sinusoidal components, which have constant amplitude and frequency, and noise. The underlying "sinusoids plus noise" model is given by

$$x(n) = \sum_{m=1}^{M} A_m \cos\left(2\pi f_m n + \phi_m\right) + i(n) \tag{3.8}$$

where $n$ is the time-index, $A_m$, $f_m$ and $\phi_m$ represent the amplitude, frequency and initial phase of the $m^{\text{th}}$ sinusoid and $M$ is the number of sinusoids (harmonics) present in the signal. $i(n)$ represents noise or other interference signal.

In the Fourier magnitude spectrum of the windowed signal, the local peaks are potential sinusoid candidates. The task is to distinguish the true sinusoidal candidates from noise and side-lobes arising due to windowing. Sinusoidal components in the Fourier spectrum can be detected based on either their magnitude or phase characteristics (Keiler & Marchand, 2002). Situations such as closely spaced components due to polyphony and time-varying pitches, however, are expected to influence the reliability of sinusoid identification. Several frame-level sinusoid parameter estimation methods proposed in the literature track the amplitude, frequency and modulation parameters under certain assumptions on the form of the modulation of the windowed sinusoidal signal (Betser, Collen, Richard, & David, 2008; Marchand & Depalle, 2008; Wells & Murphy, 2010). Constant or first-order AM and linear FM are common assumptions but it has been noted that such idealized trajectories will not be followed in real-world signals (Dressler, 2006). The influence of neighboring sinusoids in multi-component signals has typically been ignored by assuming that the window length is long enough to make it negligible.

Most sinusoid detection algorithms first detect all local maxima in the magnitude spectrum. A decision criterion (termed as a "sinusoidality" criterion), based on the spectral properties of the windowed ideal sinusoid, is then applied to the local maximum in order to decide whether it represents a sinusoid (as opposed to a window side-lobe or noise). In previous work on multi-pitch analysis these criteria have been computed from the magnitude spectrum itself (Fernandez-Cid & Casajus-Quiros, 1998) or from the phase-spectrum (Dressler, 2006; Goto, 2004). An additional criterion used for sinusoidal component identification is the validation of peaks across different resolutions (Fernandez-Cid & Casajus-Quiros, 1998).

We next describe three distinct methods of sinusoid detection from the short-time spectrum – two based on the magnitude spectrum (Every, Separation of musical sources and structure from single-channel polyphonic recordings, 2006; Griffin & Lim, 1988) and one based on the phase spectrum (Dressler, 2006). The inputs to all the methods is the magnitude spectrum $X(k)$ of the signal. All methods first search the short-time magnitude spectrum for 3-point local maxima to which they apply specific sinusoidality criteria. For the magnitude-spectrum based methods the frequency and amplitude estimates of a detected sinusoid are further refined using parabolic interpolation (Keiler & Marchand, 2002). Refinement of the sinusoidal frequency estimate is inherent in the phase-spectrum based method.

### 3.2.1 Amplitude Envelope Threshold

For musical spectra, partial amplitudes typically decay with frequency. Even for speech, the higher formants are typically weaker than the lower formants for voiced phonemes and also there are valleys between formants. So a possible peak-acceptance criterion is to use a frequency dependant threshold, preferably one that follows the spectral magnitude envelope. The method described here employs an amplitude threshold relative to the detected amplitude envelope (Every, Separation of musical sources and structure from single-channel polyphonic recordings, 2006).

The amplitude envelope of the magnitude spectrum $X_n(k)$ at time-instant $n$ is first obtained by convolving it with a Hamming window $H(k)$ in the frequency domain, as given below

$$A(k) = X_n(k) \otimes H(k) \tag{3.9}$$

where $H(k)$ is a normalized Hamming window of length $1+N/64$ frequency bins. Here $N$ is the number of points in the DFT. The length of the Hamming window used for computing the

amplitude envelope is suitably reduced when using shorter windows because the amount of smoothing required for computation of an accurate envelope is lesser for shorter window durations. Next $A(k)$ is flattened as follows

$$E(k) = \left(A(k)\right)^{c} \tag{3.10}$$

where $c$ is a compression factor. Smaller values of $c$ lead to a flatter envelope. The value $c$ = 0.8 works well in our implementation. Then a threshold height is computed as

$$\eta = K.\overline{X}^{(1-c)} \tag{3.11}$$

where $\overline{X}$ is the mean spectral amplitude and $K$ is a constant (0.7). The final threshold is given as $M\eta E(k)$, where $M$ is chosen such that the threshold is $L$ dB below $\eta E(k)$. All local maxima above this final threshold value are labeled as detected sinusoids. The sinusoidal frequency and amplitude estimate is refined by parabolic interpolation.

### 3.2.2 Main-Lobe Matching

For a stationary sound, sinusoidal components in the magnitude spectrum will have a well defined frequency representation i.e. the transform of the analysis window used to compute the Fourier transform. This second method, called main-lobe matching, utilizes a measure of closeness of a local spectral peak's shape to that of the ideal sinusoidal peak. This measure can be computed as the mean square difference (Griffin & Lim, 1988) or the cross-correlation (Lagrange, Marchand, & Rault, 2006) between the local magnitude spectrum and that of the analysis window main lobe. We use the former method based on matching the main-lobe of the window transform to the spectral region around local maxima in the magnitude spectrum. The deviation of the ideal window main-lobe magnitude-spectrum shape $W(k)$ to the spectral region around a local maxima in the magnitude spectrum $X_n(k)$ at time-instant $n$ is computed as an error function, given as

$$\varepsilon = \sum_{a}^{b}\left[X_n(k) - \left|A\right|W(k)\right]^2 \quad \text{where } A = \frac{\sum_{a}^{b} X_n(k)W(k)}{\sum_{a}^{b} W^2(k)} \tag{3.12}$$

Here $A$ is a scaling factor that minimizes $\varepsilon$ and $[a, b]$ is the interval of the main-lobe width around the local maximum. This error is normalized with the signal energy as follows

$$\xi = \frac{\varepsilon}{\sum_{a}^{b} X_n^2(k)} \tag{3.13}$$

The sinusoidality criterion, in this case a measure of the closeness of shape of the detected peak and the ideal main-lobe, is now defined as $S = 1- \xi$. Local maxima for which $S$ lies above a predefined threshold are marked as sinusoids. A strict threshold on $S$ was originally proposed (0.8) (Griffin & Lim, 1988). However we find that relaxing the threshold enables the detection of melodic F0 harmonics that may be distorted due to voice-pitch modulations, such as vibrato, while still maintaining a high side-lobe rejection. Note that a change in the window length results in a change in the shape of the ideal main lobe $W(k)$. The sinusoidal frequency estimate is refined by parabolic interpolation.

### 3.2.3 Weighted Bin-Offset Criterion

Phase-based sinusoidality criteria exploit the phase coherence of sinusoidal components by computing different instantaneous frequency (IF) estimates from phase spectra in the vicinity of the local maximum. Lagrange (2007) has demonstrated the theoretical equivalence of different IF estimation methods, which earlier were experimentally shown to perform similarly (Keiler & Marchand, 2002). We consider a version of the bin-offset method, in which the IF is computed from the derivative of the phase, further modified by Dressler (2006) to "weighted bin-offset" for the polyphonic context. This method applies thresholds to the bin offset $\kappa$, which is the deviation of the sinusoid's IF from the bin frequency of the local maxima. The bin offset at bin $k$ is given by

$$\kappa(k) = \frac{N}{2\pi L}\, princ\arg\left[\phi_n(k) - \phi_{n-1}(k) - \frac{2\pi L}{N}k\right] \tag{3.14}$$

where $\phi_n(k)$ is the phase spectrum of the $n^{\text{th}}$ frame, $N$ is the number of DFT points, $L$ is the hop length and *princarg* maps the phase to the ±π range. Local maxima are marked as detected sinusoids if

$$\kappa(k) < 0.7R \;\;,\;\; \left|\kappa(k) - \kappa(k+1) - 1\right| < 0.4 \cdot \frac{A_{peak}}{X_n(k+1)} \quad \text{and}$$
$$\left|\kappa(k) - \kappa(k-1) + 1\right| < 0.4 \cdot \frac{A_{peak}}{X_n(k-1)} \tag{3.15}$$

where $A_{peak}$ is the instantaneous magnitude of the local maxima, which is computed by applying bin-offset correction in the window transform. The bin-offset value is used to refine the sinusoidal frequency estimate $f(k)$, for sampling frequency $f_S$ using

$$f(k) = (k + \kappa(k))\frac{f_S}{N} \tag{3.16}$$

## 3.3 Evaluation

In this section we first comparatively evaluate the three methods of sinusoid identification from the short-time spectrum via simulations that exemplify polyphony and the non-stationarity of the vocal pitch in terms of sinusoid detection accuracy. We then compare the performance of the best-performing sinusoid detection method for the fixed and adaptive multi-resolution cases. In the adaptive case we compare the effect of the different sparsity measures, described in Sec. 3.1.2 for driving window-length adaptation, described in Sec. 3.1.2.2 using simulated and real data.

### 3.3.1 Signal Description

#### *3.3.1.1 Simulated Signals*

We use three simulated signals in these evaluations, all sampled at 22.05 kHz. The first two signals, described next, follow the model described in Eq. (3.8). The first signal is a representation of a steady pitched vocal utterance. The vocal signal is a vowel /a/ generated using a formant synthesizer (Slaney, 1998) at a fixed pitch of 325 Hz with harmonics upto 4 kHz ($M$=12). The second signal represents the polyphonic case by adding a relatively strong harmonic interference to the previous voice-only signal. The interference signal $i(n)$ is a complex tone, with 7 equal amplitude harmonics, with a pitch of 400 Hz. The signals are added at a Signal-to-Interference Ratio (SIR) of 0 dB. The equal-amplitude interference harmonics are, in general, stronger than the vowel harmonics that roll-off.

The third signal is an example of the time-varying nature of the voice pitch and does not fit the signal model of Eq. (3.8). This is a synthetic vocal utterance with no interference (same as the first signal) but the pitch of the vowel now contains vibrato leading to non-stationary harmonics. Vibrato for singing is described as a periodic, sinusoidal modulation of the phonation frequency (Sundberg, A rhapsody on perception, 1987). The pitch of the vibrato signal is given as

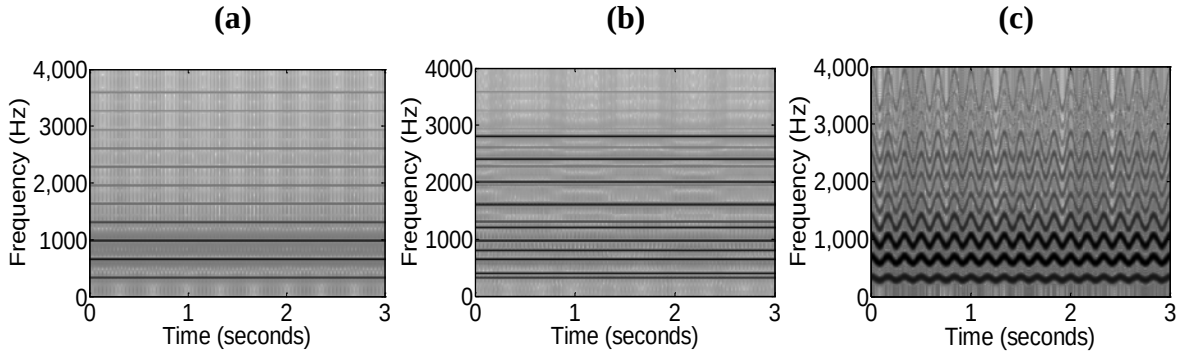


Figure 3.1: Spectrograms of (a) synthetic vowel /a/ at pitch 325 Hz, (b) mixture of previous synthetic vowel and harmonic interference (7 equal amplitude harmonics) added at 0 dB SIR, (c) synthetic vowel with base pitch 325 Hz and vibrato (extent 1 Semitone and rate 6.5 Hz).

$$f_{vib}(n) = f_{base} \cdot 2^{\left(\frac{A \cdot \sin(2\pi f_r \cdot n / F_S)}{1200}\right)} \tag{3.17}$$

where $f_{base}$ is the base frequency (325 Hz), $A$ is half the total vibrato extent, $f_r$ is vibrato rate and $F_S$ is the sampling frequency. The vibrato rates and extents we have used here are 6.5 Hz and 100 cents; these are typically measured values (Sundberg, 1987). The spectrograms of the simulated signals (each of duration 3 sec) are shown in Figure 3.1 (a), (b) & (c) respectively.

#### *3.3.1.2 Real Signals*

This data consists of vocal music where there are steady as well as pitch modulated notes. We use two datasets, sampled at 22.05 kHz, each of about 9.5 minutes duration of which the singing voice is present about 70 % of the time. The first dataset contains excerpts of polyphonic recordings of 9 Western pop songs of singers such as Mariah Carey and Whitney Houston, who are known for using extensive vibrato in their singing. The second dataset contains 5 polyphonic Indian classical vocal music recordings. Indian classical singing is known to be replete with pitch inflections and ornaments. The polyphony in Indian classical music is provided by the accompanying instruments – drone (*tanpura*), tonal percussion (*tabla*) and secondary melodic instrument (*harmonium* or *sarangi*). The expected harmonic locations are computed from the ground-truth voice-pitch, extracted at 10 ms intervals over the singing segments using a semi-automatic melody extraction tool described in Chapter 9.

### 3.3.2 Evaluation Criteria

The evaluation criteria used for the sinusoid identification methods are recall, precision and the average frequency deviation from expected (ground truth) harmonic frequency locations.

Recall is defined as the ratio of the number of correctly detected sinusoids to the true number of sinusoids present. Precision is the ratio of the number of correctly detected sinusoids to the total number of detected sinusoids. For each frame of the test signal a set of detected sinusoids (frequencies and amplitudes) is computed as those local spectral maxima that have satisfied the particular sinusoidality criterion for that method. Then the $n^{th}$ harmonic of the target signal, with known pitch *f0*, with frequency $f_n = n.f0$, is said to be correctly detected if at least one measured sinusoid, with estimated frequency $f_n^{'}$, satisfies

$$\left| f_n - f_n^{'} \right| < \min\left(0.03\, f_n,\ 50 \text{ Hz}\right) \tag{3.18}$$

Our acceptance criterion for evaluation of sinusoid detection performance is "musically related" (i.e. a percentage deviation) only in the lower frequency region. At frequencies beyond ≈1.5 kHz, it is set at the fixed value of 50 Hz. The motivation for a musically related tolerance is two-fold (1) since the sinusoids are part of harmonic structures, pitch variation causes larger deviation in the higher frequency harmonics relative to what appears in the lower harmonics; (2) human auditory sensitivity to frequency differences is frequency-dependent with higher sensitivity at lower frequencies.

If more than one measured sinusoid satisfies the above validation criterion, only that sinusoid with the smallest value of $\left| f_n - f_n^{'} \right|$ is labeled as correctly detected. All other detected sinusoids, including those that do not satisfy the validation criterion for any expected harmonic, are labeled as false alarms. So only a single measured sinusoid can be assigned to an expected harmonic. For the simulated polyphonic case, we specifically exclude the detected harmonics of the interference signal, representing musical accompaniment, from the list of false alarms. This is done by first computing the number of correct sinusoid detections for the *interference* signal, after applying the above validation criterion, and subsequently subtracting this number from the total number of false alarms for that frame.

The frequency error for the $n^{th}$ expected harmonic with frequency $f_n$ is given as

$$\begin{aligned} \text{FE}_n &= f_n - f_n^{'} \quad \text{; if a sinusoid is detected for } f_n \\ &= \quad 0 \qquad \text{; otherwise} \end{aligned}$$

We then compute the standard deviation ($\sigma_{\text{FE}}$) of the *FE* for all correctly detected harmonics for all analysis time-instants.

### 3.3.3 Comparison of Sinusoid Detection Methods

#### *3.3.3.1 Experimental Setup*

For this evaluation we only process the simulated signals using the single-resolution frequency-domain representation. For each of the simulated signals described in Sec. 3.3.1.1, we compute the evaluation metrics for each of the three sinusoid detection methods described in Sec. 3.2. within a 0 to 4 kHz frequency band. For each case we computed precision v/s recall curves by varying the parameter *M*, threshold on *S* and *R* for the amplitude-envelope, main-lobe matching and weighted bin-offset sinusoid detection methods respectively. We have reported that performance (recall & precision) that maximized the F-measure given by

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.19}$$

For the clean and polyphonic simulations we use a window-length of 92.9 ms. For the case of the vibrato vowel we have used a reduced window length (23.2 ms) rather than the 92.9 ms window. The window-length in this case was reduced to decrease the effect of signal non-stationarity within the window; all three methods showed very poor results with a 92.9 ms window for the vibrato case. The other analysis parameters for each of the methods were appropriately adjusted to provide the best possible performance with the reduced window length. In all cases, a fixed DFT size of 2048 points is retained.

Table 3.1: Performance (RE – Recall (%), PR – Precision (%), $\sigma_{FE}$ – Frequency error (Hz)) of different sinusoid detection methods for different simulated signals.

| **Signal** | | **Amplitude envelope** | **Main-lobe matching** | **Weighted bin-offset** |
|---|---|---|---|---|
| Clean vowel (92.9 ms) | RE | 100.0 | 100.0 | 93.3 |
| | PR | 100.0 | 100.0 | 93.9 |
| | $\sigma_{FE}$ | 0.3 | 0.3 | 0.5 |
| Vowel + Interference (92.9 ms) | RE | 72.72 | 98.75 | 75.7 |
| | PR | 99.53 | 100.0 | 88.7 |
| | $\sigma_{FE}$ | 15.7 | 15.3 | 14.4 |
| Vowel with vibrato (23.2 ms) | RE | 73.7 | 89.3 | 62.0 |
| | PR | 67.2 | 97.2 | 75.8 |
| | $\sigma_{FE}$ | 8.7 | 8.4 | 13.1 |

#### *3.3.3.2 Results*

The performances of the different methods for the different simulated signals appear in Table 3.1. It is observed that the main-lobe matching method is more robust to harmonic interference and pitch modulation than the other two methods. The superiority of main-lobe matching to other *single-frame* sinusoid identification methods has also been previously observed by Wells (2010). Here we note that the main-lobe matching method is also superior to the weighted bin-offset method, which relies on the phase computed from the present *and previous* analysis frame. The amplitude method suffers from distortions in the computation of the amplitude envelope itself for the polyphonic and non-stationary signals but performs well for the clean signal. The weighted bin-offset method is prone to false alarms even for the clean signal due to increased distortion in the phase spectrum of the weaker amplitude harmonics (Dressler, 2006). The frequency error metrics for all methods are similar in the cases of clean and polyphonic signals. For the vibrato signal however, this metric is higher for the bin-offset method. Since the frequency error is only computed for detected harmonic sinusoids, this indicates that the phase-spectrum is more severely affected by the presence of frequency modulation relative to the magnitude spectrum.

### 3.3.4 Comparison of Fixed- and Adaptive-Resolution Analysis

The previous section provided results only for fixed window lengths i.e. single-resolution. The choice of window length is expected to influence the reliability and accuracy of sinusoid detection and frequency estimation for the different signal conditions. In order to obtain the most accurate sinusoid detection, it is necessary to choose the window length so as to minimize the biasing of the computed sinusoidality measure due to the presence of pitch modulations and interfering components. These two non-idealities impose opposing constraints on the window length. In this section we investigate possible performance gains from using a multi-resolution representation and window-length adaptation based on measures of signal sparsity. Note that only the main-lobe matching based detection method is considered here for experimentation due to its superior performance although the important trends hold across sine identification methods.

#### *3.3.4.1 Experimental Setup*

The evaluation metrics are computed as before for three overlapping frequency bands 0–1.5 kHz, 1–3 kHz and 2.5–4 kHz, since we expect signal non-stationarity to increase at higher

frequencies. We process each signal using three single-resolution analyses (window-lengths of 23.2, 46.4 & 92.9 ms), a fixed multi-resolution analysis (where the above window-lengths are applied in descending order to the three increasing frequency bands respectively) and three adaptive multi-resolution analyses (in which the window-length adaptation described in Sec. 3.1.2.2, within each frequency band is driven individually by the L2 norm (L2), normalized kurtosis (KU), Gini index (GI), Hoyer measure (HO) and spectral flatness (SF) computed within that band). For the simulated signals we only process the polyphonic and vibrato signals. For real signals we only compute recall using the expected harmonic locations computed from the ground-truth voice-pitch. Although we could compute the precision as well from the number of false positives for the real signals, this would not be indicative of sinusoid detection performance since there could be various simultaneously present accompanying musical instruments which also have harmonic spectra. Recall is only computed during active frames i.e. those for which the singing voice is present.

#### *3.3.4.2 Results*

1. **Simulated signals**: The results for the polyphonic and vibrato simulated signals are shown in Table 3.2 and Table 3.3 respectively. From these tables it can be seen that, rather than the multi-resolution analysis, the 92.9 ms and 23.2 ms fixed resolution analyses consistently give the best performance across all bands for the polyphonic and vibrato simulations respectively. For the vibrato signal the disparity in performance across the different windows is more significant in the higher frequency band since the extent of non-stationarity in the signal is proportionately higher in this band. Of the different adaptive cases, the normalized kurtosis and Hoyer measure are observed to closely capture the longer-window superiority for the polyphonic signal and the shorter-window superiority for the vibrato signal across all frequency bands. A large difference in the performance of the sparsity measures is observed for the vibrato signal, especially in the highest frequency band.

   From these results it seems that using a signal sparsity-driven adaptive window analysis should lead to better sinusoid identification across varying signal conditions of polyphony and non-stationarity (in terms of pitch modulation) as compared to a multi-resolution approach. Since the singing voice would be the dominant source in vocal music, we expect that the above method should show good sinusoid identification performance for real music signals as well.

2. **Real signals**: The results for the Western pop and Indian classical datasets are presented in Figure 3.2. Adaptive windowing improves upon the performance of the fixed multi-resolution analysis as seen from the increased recall. Overall it can be seen that the kurtosis-driven window adapted sinusoid detection gives better performance than any fixed or adaptive window analysis method across the datasets.

Table 3.2: Performance of window main-lobe matching method (RE – Recall (%), PR – Precision (%), $\sigma_{FE}$ – Frequency error (Hz)) for different fixed windows (23.2, 46.4, 92.9 ms & multi-resolution) and sparsity (L2 norm, KU – Kurtosis, GI – Gini Index, SF – Spectral flatness and HO - Hoyer) driven adapted windows for simulated *polyphonic* signal.

| **Band** | **0-1.5 kHz** | | | **1-3 kHz** | | | **2.5-4 kHz** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **RE** | **PR** | $\sigma_{FE}$ | **RE** | **PR** | $\sigma_{FE}$ | **RE** | **PR** | $\sigma_{FE}$ |
| Fixed Single- and Multi-resolution analysis | | | | | | | | | |
| 23.2 ms | 50.0 | 100.0 | 2.1 | 36.8 | 80.0 | 27.2 | 40.9 | 98.0 | 13.1 |
| 46.4 ms | 100.0 | 100.0 | 0.4 | 78.1 | 100.0 | 23.7 | 62.7 | 99.1 | 3.4 |
| 92.9 ms | 100.0 | 100.0 | 0.1 | 98.6 | 100.0 | 18.1 | 96.6 | 100.0 | 0.3 |
| MR | 100.0 | 100.0 | 0.1 | 78.1 | 100.0 | 23.7 | 40.9 | 98.0 | 13.1 |
| Signal sparsity-driven adaptive windowing | | | | | | | | | |
| L2 | 100.0 | 100.0 | 0.1 | 98.6 | 100.0 | 18.1 | 96.6 | 100.0 | 0.3 |
| KU | 100.0 | 100.0 | 0.1 | 98.6 | 100.0 | 18.1 | 96.6 | 100.0 | 0.3 |
| GI | 100.0 | 100.0 | 0.1 | 97.4 | 100.0 | 18.1 | 89.4 | 100.0 | 0.8 |
| SF | 100.0 | 100.0 | 0.2 | 97.1 | 100.0 | 18.1 | 86.8 | 100.0 | 0.5 |
| HO | 100.0 | 100.0 | 0.1 | 97.5 | 100.0 | 18.1 | 92.8 | 100.0 | 0.3 |

Table 3.3: Performance of window main-lobe matching method (RE – Recall (%), PR – Precision (%),$\sigma_{FE}$ – Frequency error (Hz)) for different fixed windows (23.2, 46.4, 92.9 ms & multi-resolution) and sparsity (L2 norm, KU – Kurtosis, GI – Gini Index, SF – Spectral flatness and HO - Hoyer) driven adapted windows for simulated *vibrato* signal.

| **Band** | **0-1.5 kHz** | | | **1-3 kHz** | | | **2.5-4 kHz** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **RE** | **PR** | $\sigma_{FE}$ | **RE** | **PR** | $\sigma_{FE}$ | **RE** | **PR** | $\sigma_{FE}$ |
| Fixed and Multi-resolution analysis | | | | | | | | | |
| 23.2 ms | 97.1 | 100.0 | 1.4 | 90.0 | 97.9 | 6.4 | 82.7 | 98.0 | 8.1 |
| 46.4 ms | 97.4 | 100.0 | 2.3 | 56.2 | 96.3 | 9.4 | 46.9 | 81.7 | 15.1 |
| 92.9 ms | 64.8 | 100.0 | 6.2 | 54.0 | 86.7 | 17.8 | 48.8 | 52.9 | 18.9 |
| MR | 64.8 | 100.0 | 6.2 | 56.2 | 96.3 | 9.4 | 82.7 | 98.0 | 8.1 |
| Signal sparsity-driven adaptive windowing | | | | | | | | | |
| L2 | 65.3 | 100.0 | 5.4 | 55.4 | 80.7 | 13.0 | 48.3 | 82.4 | 16.3 |
| KU | 96.7 | 100.0 | 2.7 | 86.5 | 94.3 | 7.0 | 73.0 | 91.4 | 9.2 |
| GI | 72.4 | 100.0 | 4.1 | 50.9 | 84.6 | 9.3 | 56.2 | 79.8 | 11.5 |
| SF | 89.3 | 95.8 | 3.3 | 77.3 | 91.2 | 7.1 | 47.5 | 81.6 | 15.3 |
| HO | 96.9 | 100.0 | 2.9 | 63.9 | 92.5 | 8.1 | 60.2 | 88.6 | 10.7 |

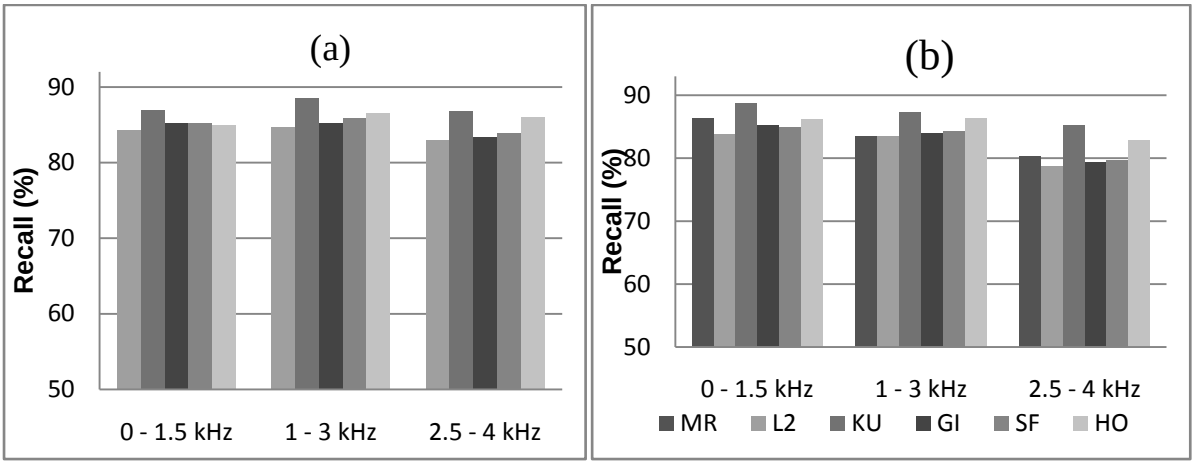


Figure 3.2: Performance of window main-lobe matching for multi-resolution (MR) and sparsity measures (L2 norm, KU – Kurtosis, GI – Gini Index, SF – Spectral flatness and HO – Hoyer measure) driven adapted windows for different frequency bands for (a) Western pop data and (b) Indian classical data.

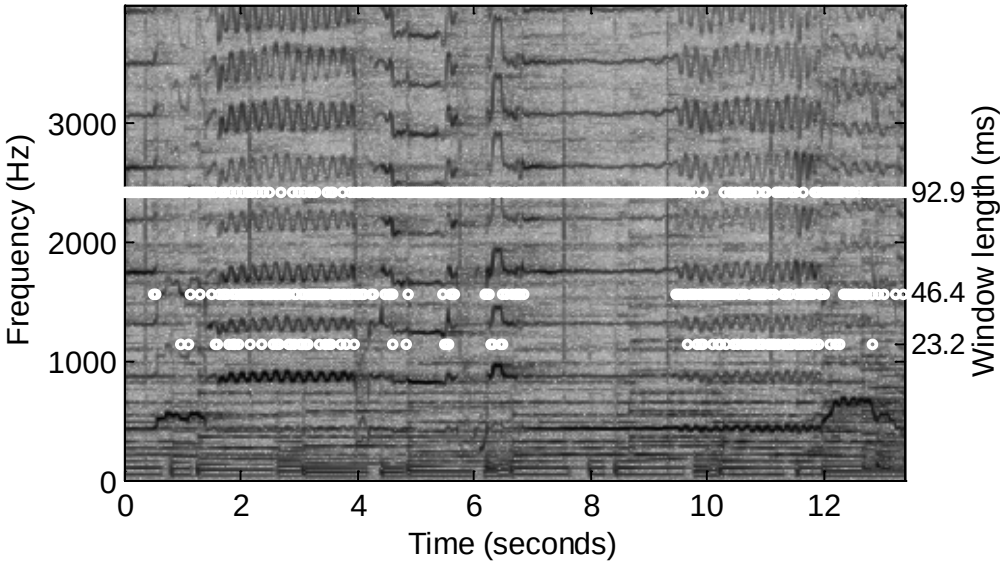


Figure 3.3: Spectrogram of an excerpt of Whitney Houston's "I will always love you". White circles represent window choice (92.9, 46.4 or 23.2 ms) driven by maximization of kurtosis in the 2.5-4 kHz frequency band.

An example of window adaptation using kurtosis for the highest frequency band is shown for an excerpt from the Western pop dataset in Figure 3.3. Here it can be seen that during the stable notes (from 3 to 5 sec) the measure is maximized for the longest window but during the vibrato regions (from 1 to 2 sec and 5 to 6 sec) the measure frequently favors lower window lengths. Further, during vibrato the longer windows are selected in frames corresponding to the peaks and valleys of the vibrato cycle, and shorter windows are chosen during the vibrato mean crossings where the rate of frequency variation is highest.

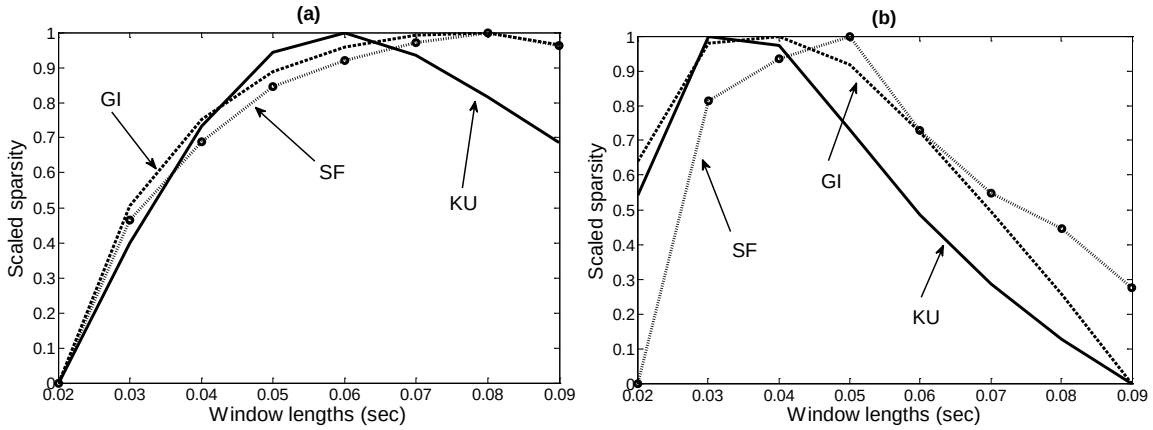


Figure 3.4: Scaled sparsity values (KU, GI and SF) computed for different window lengths for a pure-tone chirp for (a) slow and (b) fast chirp rates.

## 3.4 Discussion

The observed performance improvements from sparsity driven window-length adaptation suggest that certain sparsity measures do indeed serve to usefully quantify spectrum shape deviation from that of an ideal sinusoid. A simple example, provided next, demonstrates directly the relation between computed sparsity and the biasing of the spectrum from the specific trade-off between time and frequency resolutions in the representation of non-stationary signals. Consider linear chirp pure tones with fast and slow chirp-rate. Let the slow rate equal to one-eighth the fast chirp-rate, and both belong within the typical range of voice pitch modulations (e.g. vibrato). For each of the chirps we plot different sparsity measures (KU, GI and SF) versus window length, varying from 20 ms to 90 ms in steps of 10 ms, in Figure 3.4. We see that all three sparsity measures show the intuitively expected concave form, attaining a single maximum at a finite window length which itself decreases as the chirp rate increases. We observe that KU is most sensitive to chirp rate. We have not plotted the HO and L2 measures since the former shows similar trends as KU and the latter does not show any sensitivity to changing chirp rates but continues to increase in value with window length.

A closer inspection of the dependence of computed sparsity on spectrum shape revealed that the GI is affected by the shape of the main-lobe as well as the side-lobe roll-off whereas the KU reflects main-lobe spread mainly with the low amplitude side-lobes scarcely affecting the $4^{th}$ power average in Eq. 3.4. For similar main-lobe shapes, such as those that occur for the 30 and 40 ms windows for the fast chirp signal, the GI is found to have larger values when the side-lobes display greater roll-off i.e. for the larger (40 ms) window. This is in keeping with the graphical representation of GI shown in Fig. 1 in (Hurley & Rickard, 2009), where

lowering the more populated side-lobe values for similar main-lobes will result in an increase in the shaded area of the same figure, thereby resulting in a greater GI. On the other hand the KU is not affected by any change in the roll-off of the side-lobes but even a slight broadening of the main lobe reduces its value. Since sinusoid detection is all about identifying well-formed main-lobes the KU measure shows a greater sensitivity to signal non-stationarity. This explains, in part, the superiority of KU in the sinusoid detection context inspite of the general superiority of GI as a sparsity measure (Hurley & Rickard, 2009).

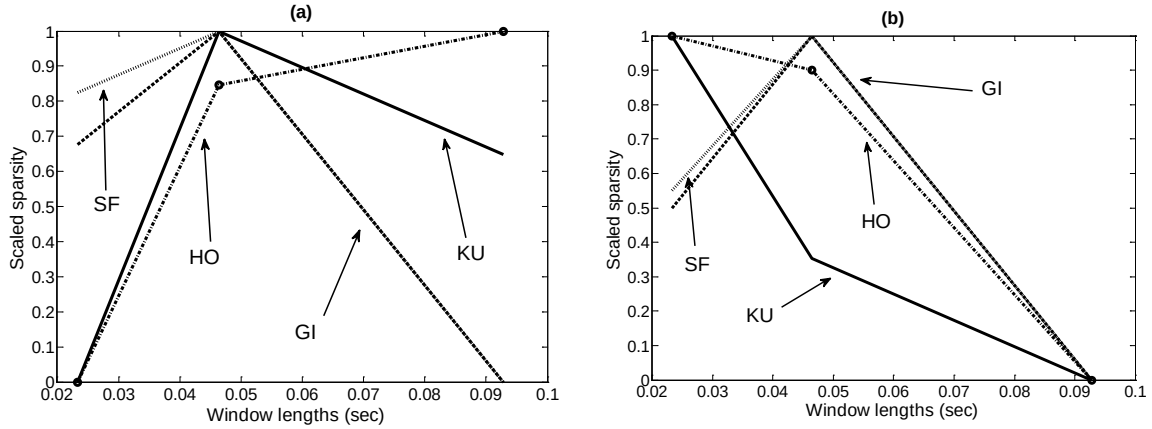


Figure 3.5: Sparsity measures (KU, GI, SF and HO) computed for different window lengths (23.2, 46.4 and 92.9 ms) for the vibrato signal at points of (a) minimum and (b) maximum frequency change. The measures are scaled to lie in [0, 1].

This sensitivity is also visible for the simulated vibrato vowel signal described in Sec. 3.3.1.1. A detailed observation of the behavior of the different sparsity measures for different analysis time-instants for the simulated vibrato signal, is presented in Figure 3.5. Here we compute the sparsity values for the different window lengths (23.2, 46.4 and 92.9 ms) at two isolated time-instants (frames) that display contrasting signal content. The first frame is centered at the point of minimal change i.e. at the peak (or valley) location of the frequency curve where the frequency is almost steady. The second frame is centered at the point of maximal change in vibrato i.e. at the point the pitch curve crosses the mean vibrato harmonic frequency. The displayed sparsity values for different windows are computed in the highest band (2.5 to 4 kHz) and are scaled to be between 0 and 1. From the figure we can see that the SF and GI measures show no sensitivity to the signal variation and maximize at the same window length (46.4 ms) for both cases. The L2 measure was also found to behave similarly but we have avoided plotting it for figure clarity. The KU measure maximizes at 46.4 and 23.2 ms, and the HO measure maximizes at 92.8 and 23.2 ms for the minimum and maximum frequency change points respectively. Since the duration of a single cycle of vibrato in our

signal is 153 ms, the 92.8 ms window spans more than half the cycle and leads to more spectral distortion, especially in the higher bands. So, of the different window adaptation approaches, we find the best sinusoid detection performance for the KU measure which switches between the 23.2 and 46.4 ms windows over the course of the vibrato signal.

This non-stationarity dependent window-length switching behavior of KU also comes out in the real signal example of Figure 3.3. For this example, GI does not switch window-lengths as often.

## 3.5 Summary and Conclusions

In this chapter we have described different approaches to effective spectral signal representation for music signal processing, specifically singing voice processing in polyphony, at various stages. At the frequency-domain representation stage the options available are fixed single-resolution, fixed multi-resolution and adaptive multi-resolution. We investigated the adaptation of window-length using different measures of signal sparsity such as L2 norm (L2), kurtosis (KU), Gini index (GI), Hoyer measure (HO) and spectral flatness (SF). At the sinusoidal stage, sinusoids may be extracted using amplitude envelope thresholding, main-lobe matching and bin-offset criterion. The above approaches can be combined appropriately; for example we can use main-lobe matching sinusoid detection on a kurtosis-driven adaptive multi-resolution frequency domain signal representation.

The above signal analysis methods have been evaluated using simulated and real signals that are indicative of the polyphony and non-stationary problems that arise in vocal music. Our data consists of excerpts of western pop and north Indian classical music in which stable notes and large pitch modulations are both present. One result of this chapter is that the main-lobe matching-based sinusoid detection method outperforms the amplitude envelope thresholding and bin-offset criterion based sinusoid detection method for the fixed resolution frequency-domain signal representation, when evaluated on simulated signals. Another result is that sparsity driven window-length adaptation consistently results in higher sinusoid detection rate and minimal frequency estimation error when compared with fixed window analysis in the context of sinusoid detection of the singing voice in polyphonic music. KU applied to the local magnitude spectrum is found to outperform alternate measures of signal sparsity.

Essentially, the work on sparsity-driven window adaptation in this chapter can be viewed as a way to introduce window adaptation within *any* chosen short-time analysis framework.

While we have employed the STFT, the standard method for time-frequency analysis for the detection and estimation of quasi-stationary sinusoids, it is quite possible to consider extending adaptive windowing to a method that estimates the chirp parameters of linear frequency and amplitude modulated sinusoids such as the work of Betser, Collen, Richard & David (2008) or the Fan Chirp Transform of Cancela, Lopez & Rocamora (2010). Since linear modulation of sinusoid parameters is again an idealization that would hold only over limited window length for real audio signals, the decrease in signal sparsity (reduced acuity) observed, for example, in the Fan Chirp Transform domain caused by deviations from ideality could be used as the basis for window length selection for optimal parameter estimation. Therefore the results of this chapter on STFT based sine detection and estimation can be considered as a beginning for future work on window adaptation for time-varying sinusoid detection and estimation.

Although the adaptive multi-resolution signal representation is shown to be superior to fixed resolution in this chapter, all subsequent modules in the melody extraction system use the fixed single resolution sinusoidal representation. The work on sparsity-driven window-length adaptation was done at the end of the research in this thesis and has not as yet been incorporated or evaluated in the melody extraction frame-work.

# Chapter 4

## Multi-F0 Analysis

The goal of this chapter is to process the signal representation of each audio frame in order to reliably detect potential F0 candidates, in that frame, with their associated salience values. The term “salience” is employed to designate the (approximate) relative quality of a potential pitch candidate, regardless of how it is computed. In the present context, there are two requirements of the multi-F0 extraction module: 1) the voice-F0 candidate should be reliably detected; 2) the salience of the voice-F0 candidate should be relatively high compared to the other F0 candidates.

As described by de Cheveigné (2006), frame-level signal analysis for detecting multiple F0s can be conducted in one of 3 possible ways: In the first, a single voice (i.e. monophonic) estimation algorithm is applied in the hope that it will find cues to several F0s. In the second strategy (iterative estimation), a single voice algorithm is applied to estimate the F0 of one voice, and that information is then used to suppress that voice from the mixture so that the F0s

of the other voices can be estimated. Suppression of those voices in turn may allow the first estimate to be refined, and so on. This is an iterative “estimate-cancel-estimate” mechanism. In a third strategy (joint multiple F0 estimation) all the voices are estimated at the same time. The latter approaches are described as being superior to the simple extension of a single-F0 algorithm to estimate multiple F0s e.g. identifying the largest and second largest peak in an autocorrelation function (ACF). Both the iterative and the joint estimation approaches are of significantly higher computational complexity than a single voice (monophonic) algorithm applied to find multiple F0s.

The salience function used by most multi-F0 analysis systems is based on matching some expected representation (mostly spectral) with the measured signal representation. The salience values usually correspond to the quality of the match. Some of these salience functions can be grouped under the “harmonic sieve” or “harmonic sum” category (Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007) and have been used by several monophonic PDAs. Cao, for instance, uses peak locations and peak strengths in the sub-harmonic sum function as F0 candidates and salience values (Cao, Li, Liu, & Yan, 2007). Rynnanen & Klapuri (2008) and Cancela (2008) almost directly implement a harmonic sum function while Klapuri (2003) implements band-wise harmonic sum functions and computes a global weight function by summing the squares of such functions across all bands. Goto (2004) uses the expectation-maximization (EM) algorithm to estimate weights for a set of harmonic tone-models for multiple F0. An F0 probability distribution function (PDF) is formed by summing the weights for all tone models at different F0s. The peak locations and values in this PDF are the F0 candidates and salience values respectively. Of the algorithms that make use of frequency domain representations motivated by auditory processes, two of them use peak locations and strengths in the sum of autocorrelation functions, also called ‘summary autocorrelation function’, computed within individual auditory channels as F0 candidates and salience values respectively (Paiva, Mendes, & Cardoso, 2006; Tolonen & Karjalainen, 2000). Klapuri (2008) uses a harmonic sum function on the sum of spectra in individual channels, also called ‘summary spectrum’. Li & Wang (2005) discriminate between clean and noisy channels and additionally integrate the periodicity information across channels using a statistical model.

In this chapter we first describe five different salience functions that are all obtained from PDAs designed primarily for monophonic pitch extraction. We then investigate the

robustness of these to harmonic interference by way of experimenting with simulated signals[1]. We finally describe the extension of one of the monophonic salience functions, which displayed stronger robustness in the previous experiment, to reliable multi-F0 analysis in the context of melody extraction. The advantage of extending a monophonic salience function for multi-F0 analysis without having to resort to iterative or joint multiple-F0 estimation or salience functions that apply band-wise processing of the spectrum, is a large saving in computational cost as well as independence from having to set accompaniment-dependent band-wise weighting.

## 4.1 Description of Different Salience Functions

The F0 candidates and their associated salience values are usually computed using a pitch detection algorithm (PDA). In this section we compare five different PDAs in terms of suitability for selection as a salience function robust to pitched interference. Recent approaches to melody (predominant-F0) extraction in Western music have made use of PDAs that are either correlation lag-based or spectrum magnitude-based (Poliner, Ellis, Ehmann, Gomez, Streich, & Ong, 2007). Accordingly, two of the PDAs considered belong to the former category and remaining fall in the latter. The two correlation-based PDAs are an implementation of the auto-correlation function (ACF) by Boersma (1993), and a derivative of the ACF called YIN (de Cheveigne & Kawahara, YIN, a fundamental frequency estimator for speech and music, 2002). In the spectral category the three PDAs considered are the sub-harmonic summation (SHS) (Hermes, Measurement of pitch by sub-harmonic summation, 1988), pattern matching (PM) (Brown, 1992), and the two-way mismatch method (TWM) (Maher & Beauchamp, Fundamental frequency estimation of musical signals using a two-way mismatch procedure, 1994). ACF and SHS have been designed for, and applied primarily to, speech while the others have been designed for musical signals.

A PDA produces a set of pitch estimates and associated reliability values at uniform time intervals. Here we briefly describe the implementation of each of the PDAs (as described in the original papers), their core salience functions as computed by the short-term analysis of windowed signal samples, and also describe any modifications made in order to optimize the performance for our context. Unless specifically mentioned, the values of the parameters used for each PDA are the same as recommended in the original reference. For each of the PDAs,

[1] The initial investigations for comparing different PDAs for salience function selection were done together with Ashutosh Bapat.

possible F0 candidates are the locations of local maxima, for ACF, PM and SHS, or local minima, for YIN and TWM, in their respective core functions, and the values of the core functions at these locations (also called local strengths) are the reliability measures or salience values.

The correlation-based PDAs operate on the windowed time-domain signal. The front end for all the frequency-domain PDAs considered here is a fixed window STFT magnitude. For all cases we used a high-resolution FFT (bin width 2.69 Hz) computed from a Hamming windowed signal segment of length chosen so as to reliably resolve the harmonics at the minimum expected F0. Also, only frequency content below 5 kHz, an acceptable upper limit for significant voice partials, was considered for real signals. The SHS and TWM PDAs require the detection of sinusoidal peaks in the spectrum. As mentioned in the previous chapter this was achieved by the main-lobe matching approach to sinusoid identification i.e. selecting those local maxima whose sinusoidality (Griffin & Lim, 1988), a measure of how closely the shape of a detected spectral peak matches the known shape of the window main lobe, was above a suitable threshold. The threshold value on sinusoidality used here was 0.6. Further, since the SHS and PM PDAs require the frequency bins of the spectrum to be logarithmically spaced, the magnitude spectral values at these locations were computed using cubic spline interpolation at a resolution of 48 points per octave.

### 4.1.1 Auto-Correlation Function (ACF)

The use of the autocorrelation function for pitch detection was originally proposed by Rabiner (1977). The modifications to the ACF proposed by Boersma (1993) improve its robustness to additive noise, large F0 ranges and decrease sensitivity to strong formants.

This method requires that the length of the segment of the signal (frame) should contain at least three periods of the minimum expected F0. The selected segment is then multiplied by a Hamming window. Further, the ACF is computed as the inverse DFT of the power spectrum, which is computed from the zero padded windowed signal segment. The ACF of the signal is then divided by the ACF of the Hamming window, which was computed in the same manner. Next, possible F0 candidates are computed as the interpolated values of the locations of local maxima in the ACF that lie within the F0 search range. The local strength of each candidate is computed by applying some biasing towards higher frequencies, using the OctaveCost parameter, to the interpolated value of the ACF strength at the candidate location. This biasing increases the robustness of the algorithm to additive noise, which

causes unwanted local downward octave jumps. The recommended value of this parameter is 0.01. The candidate with the highest local strength is the final F0 estimate.

Although the ACF is computed here as the IDFT of the power spectrum of the signal, we consider the core function of the ACF in the time domain. The autocorrelation function is computed for an *N*-sample long segment of a signal *x*, centered at time instant *t* and is then normalized by the zero-lag ACF as shown below

$$r_t(\tau) = \sum_{n=0}^{N-1} x(n)x(n+\tau); \qquad r_t^{'}(\tau) = \frac{r_t(\tau)}{r_t(0)} \tag{4.1}$$

The normalized auto-correlation of the signal is then divided by the normalized autocorrelation of the window function.

### 4.1.2 YIN

The YIN PDA (de Cheveigne & Kawahara, YIN, a fundamental frequency estimator for speech and music, 2002) is derived from the ACF PDA. The length of the signal segment is required to contain at least two periods of the minimum expected F0. From this segment the average squared difference function is computed as opposed to the ACF, as shown below.

$$d_t(\tau) = \sum_{j=0}^{N-1} \left(x(n) - x(n+\tau)\right)^2 \tag{4.2}$$

$d(\tau)$ can also be expressed in terms of $r(\tau)$, computed in Equation 4.1, as

$$d_t(\tau) = r_t(0) + r_{t+\tau}(0) + 2r_t(\tau) \tag{4.3}$$

This function is further modified to a cumulative mean normalized difference function (CMNDF), which reduces the sensitivity to strong first formants and removes the upper frequency limit on the F0 search range. The CMNDF is the core function of the YIN PDA and is given by

$$d_t^{'}(\tau) = \begin{cases} 1, \ if \ \ \tau = 0 \\ d_t(\tau) \Big/ \left[ (1/\tau) \sum_{j=1}^{\tau} d_t(j) \right] \ \ otherwise \end{cases} \tag{4.4}$$

Possible F0 candidates are computed as the interpolated values of the locations of local minima in the CMNDF that lie within the F0 search range. The smallest value of lag that

gives a minimum whose CMNDF value is below some absolute threshold (recommended value 0.1) is reported as the estimated F0.

### 4.1.3 Sub-Harmonic Summation (SHS)

The SHS PDA is based on a spectral compression PDA (Schroeder, 1968), but applies some modifications, which involve a transition from a linear to a logarithmic frequency abscissa, that 1) improve the accuracy of measurement and increases the upper limit on the rank of compressions practically viable and 2) brings the method in line with a simple auditory model that states that the perception of pitch arises from sub-harmonics generated in the central pitch processor that adds up. The implementation of the algorithm as in the original paper (Hermes, Measurement of pitch by sub-harmonic summation, 1988) is given below.

The length of the signal segment is 40 ms for speech sampled at 10 kHz. This segment is low pass filtered (cutoff frequency = 1250 Hz) by an averaging process. The segment is then multiplied by a Hamming window and zero-padded. The magnitude spectrum is calculated by a 256 point FFT applied to the resulting segment. All spectral magnitude values that are more than 2 frequency bins away from local maxima points are set to zero. The resulting spectrum is smoothed with a Hanning filter. Further, the values of the spectrum on a logarithmic frequency scale are computed for 48 equidistant points per octave using cubic spline interpolation, which is then multiplied by a raised arctangent function that represents the sensitivity of the auditory system for frequencies below 1250 Hz. The result is shifted along the logarithmic frequency axis, which is equivalent to compressing along a linear frequency axis, multiplied by an exponentially decaying factor, which gives lesser weighting to higher harmonics, and summed to give the sub-harmonic sum spectrum, which is the core function, given below.

$$H(s) = \sum_{n=1}^{N} h_n P(s + \log_2 n) \tag{4.5}$$

Here $P(s)$ is the magnitude spectrum with logarithmic frequency abscissa and $h_n$ is an exponentially decreasing function, given by $0.84^{n-1}$, which gives more importance to the lower harmonics. In order to give almost equal weighting to low and high frequency components the weighting function $h_n$ was changed to $0.99^{n-1}$. Additionally, the number of spectral shifts ($N$) was increased from 15, as originally proposed, to 30 in order to increase the number of voice partials included in the computation of $H(s)$.

The value of frequency for which this spectrum is strongest is the estimated F0. The local maxima locations in the sub harmonic sum spectrum are then the possible F0 candidates and their respective local strengths are the values of the sub harmonic sum spectrum at these locations.

### 4.1.4 Pattern Matching

The PM PDA, originally developed for music, exploits the fact that for a logarithmic frequency scale, corresponding to musical intervals, a harmonic structure always takes on the same pattern regardless of the value of its F0. Consequently, a pattern recognition algorithm was applied to detect such patterns in the spectrum by correlating with ideal spectral patterns expected for different trial F0.

In the original algorithm, no particular frame size is specified but in the experiments reported with instrument sounds, a frame size of 16 ms was used. The logarithmically-spaced magnitude spectrum with 24 points per octave was computed by means of a constant Q transform. Then a cross correlation function was computed between the measured magnitude spectrum and an ideal spectrum with a fixed number of components. If $X_n(k)$ is the signal magnitude spectrum at time instant $n$ and $I(k)$ is ideal magnitude spectrum, with $M$ frequency bins, then the cross correlation function is given as

$$C(\psi) = \sum_{k=0}^{M-1} I(k) X_n(k+\psi) \tag{4.6}$$

The optimal number of components was empirically determined for different musical instruments (e.g. Flute – 4 components, Violin – 11 components). The optimal number of components for the ideal spectral pattern of the singing voice was determined in a side experiment to be 10 and 6 components for low and high pitched synthetic vowels respectively. The value of frequency for which the cross correlation function is strongest is the estimated F0. The local maxima locations in the cross correlation function are then the possible F0 candidates and their respective local strengths are the values of the cross correlation function at these locations.

### 4.1.5 Two-Way Mismatch (TWM)

The TWM PDA qualifies as a spectral method for pitch detection. However it is different from the SHS and PM methods in that it minimizes an unconventional spectral mismatch

error which is a particular combination of an individual partial's frequency deviation from the ideal harmonic location and its relative strength. The original implementation is given next.

The magnitude spectrum of a short-time segment of the signal is computed using a 1024 point FFT, at a sampling frequency of 44.1 kHz. The parabolically interpolated locations and magnitudes of the sinusoids in the magnitude spectrum, identified using the main-lobe matching sinusoid detection method, are then the measured partials. Further, for each trial F0, within the search range, the TWM error is computed between a predicted spectrum with $N$ partials ($8 \leq N \leq 10$) and the sequence of measured partials. The TWM error function, for a given trial F0, is computed as shown below

$$\mathrm{Err}_{\mathrm{total}} = \mathrm{Err}_{\mathrm{p}\rightarrow\mathrm{m}} / N \; + \; \rho \mathrm{Err}_{\mathrm{m}\rightarrow\mathrm{p}} / K \tag{4.7}$$

Here $N$ and $K$ are the number of predicted and measured harmonics. The TWM error is a weighted combination of two errors, one based on the mismatch between each harmonic in the predicted sequence and its nearest neighbor in the measured partials ($\mathrm{Err}_{\mathrm{p}\rightarrow\mathrm{m}}$) and the other based on the frequency difference between each partial in the measured sequence and its nearest neighbor in the predicted sequence ($\mathrm{Err}_{\mathrm{m}\rightarrow\mathrm{p}}$). This two-way mismatch helps avoid octave errors in the absence of interference. The recommended value of $\rho$ is 0.33. The F0 candidate locations are the locations of local minima in the TWM error and their local strengths are $1 - \mathrm{Err}_{\mathrm{total}}$.

Both $\mathrm{Err}_{\mathrm{p}\rightarrow\mathrm{m}}$ and $\mathrm{Err}_{\mathrm{m}\rightarrow\mathrm{p}}$ share the same form. $\mathrm{Err}_{\mathrm{p}\rightarrow\mathrm{m}}$, which is the more important of the two, is defined below.

$$\mathrm{Err}_{\mathrm{p}\rightarrow\mathrm{m}} = \sum_{n=1}^{N} \left[ \frac{\Delta f_n}{(f_n)^p} + \left( \frac{a_n}{A_{\max}} \right) \left( q \frac{\Delta f_n}{(f_n)^p} \; - \; r \right) \right] \tag{4.8}$$

Here $f_n$ and $a_n$ are the frequency and magnitude of a single predicted harmonic. $\Delta f_n$ is the difference, in Hz, between this harmonic and its nearest neighbor in the list of measured partials. $A_{\max}$ is the magnitude of the strongest measured partial. Thus an amplitude weighted penalty is applied to a normalized frequency error ($\Delta f / f$) between measured and predicted partials for the given trial F0. Recommended values of $p$, $q$ and $r$ are 0.5, 1.4 and 0.5 respectively. Higher values of $p$ serve to emphasize low frequency region errors.

Unlike originally proposed, here $N$ is not fixed over all trial F0 but is computed as floor($F_{\max}/F0$), where $F_{\max}$, as stated before, is the upper limit above which the spectral content is considered not useful for voice F0 extraction (here $F_{\max}$ =5 kHz). Further, since we do not explicitly assume prior knowledge of the frequency region of the interference, we have

used a lower value of $p$ = 0.1 leading to more equal emphasis on low and high frequency regions. Additionally, it is found that using $\rho$ = 0.25 favours the target voice fundamental, when the interference is characterized by a few partials only, by placing higher emphasis on $Err_{p \rightarrow m}$.

## 4.2 Comparative Evaluation of Different Salience Functions

Most PDAs can be classified as spectral (spectral pattern matching) or temporal (maximization of a correlation-type function). These approaches have been shown to be equivalent i.e. minimizing the squared error between the actual windowed signal spectrum and an idealized harmonic spectrum is analytically equivalent to maximizing an autocorrelation function of the windowed signal (Griffin & Lim, 1988; Wise, Caprio, & Parks, 1976). The above PDAs fall under the "harmonic sieve" category (de Cheveigne, 2006). An important consequence of this is that both spectral and temporal methods put strong emphasis on high amplitude portions of the spectrum, and thus are sensitive to the presence of interference containing strong harmonics.

The aim of the experiment described in this section is to compare the PDAs in terms of the voice-pitch detection and salience in the context of dominant F0 extraction in the presence of a tonal harmonic interference. Synthetic signals which emulate the vocal and percussion combination are generated as described next so that signal characteristics can be varied systematically, and the ground truth pitch is known.

### 4.2.1 Generation of Test Signals

#### *4.2.1.1 Target signal*

The target signal is a sustained vowel (/a/), generated using a formant synthesizer, at a sampling frequency of 22050 Hz, with time-varying F0. In order to simulate the F0 variations in Indian classical singing and the typical vocal range of a singer (about 2 octaves), the time variation of the F0 of the synthetic vowel smoothly sweeps ± 1 octave from a chosen base F0 at a maximum rate of 3 semitones/sec. Two target signals are synthesized using low (150 Hz) and high (330 Hz) values of base F0 respectively. The synthetic vowels have duration 21 sec in which the instantaneous F0 completes six oscillations about the base F0.

#### *4.2.1.2 Interference signal*

The interference signal is modeled on the tonal tabla strokes. Since the tabla is tuned to the tonic of the singer, we can expect interference partials at the harmonics of the tonic. The interference signals for each of the base F0s, are complex tones having 1, 3, 5 and 7 equal magnitude harmonics at F0 equal to the target's base F0. The amplitude envelope of a sequence of tun strokes, each of which maximally decays over 1.5 seconds, is superimposed on the complex tone. This results in 14 strokes over the target signal duration. These complex tones are added to the target signals such that the worst-case local SIR around the onset of any stroke is -10 dB. For each base F0, there are five cases: 1 clean vowel and 4 noisy vowels. Spectrograms for the target with low base F0, the interference with 7 harmonics at the target base F0 and the mixed signal at -10 dB SIR are shown in Figure 4.1.

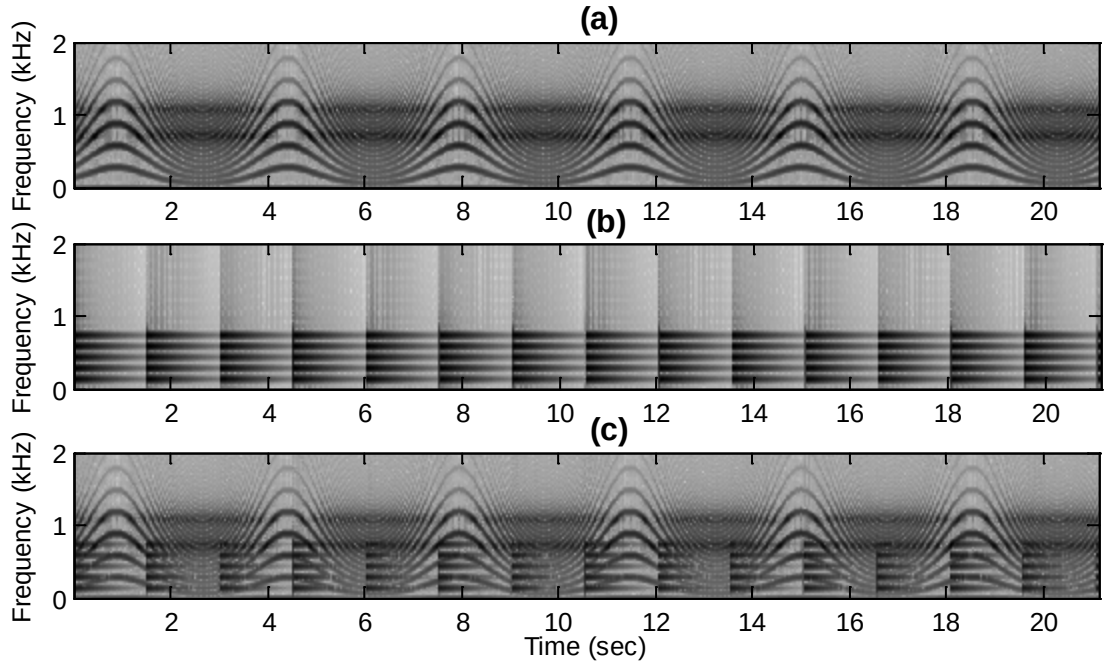


Figure 4.1: Spectrograms of (a) the target at low base F0, (b) the interference with 7 harmonics at the target F0 and (c) the mixed signal at -10 dB SIR. The target harmonics vary smoothly over 2 octaves. The vertical lines in the interference spectrogram mark the onset of each stroke after which the harmonics of the interference decay.

### 4.2.2 Evaluation Metrics

The pitch accuracy (PA) is defined as the proportion of voiced frames in which the estimated fundamental frequency is within ±1/4 tone (50 cents) of the reference pitch. As the local measurement cost provided by a PDA should represent the reliability of the corresponding F0 candidate, it is derived from the local strength of each PDA at local minima/maxima in its core function. In the context of predominant (melodic) F0 extraction, the suitability for

dynamic programming-based post-processing is determined by the quality of the measurement cost reflected by the salience of the underlying melodic F0 in the presence of typical interferences. Salience of a candidate at the melodic F0 is computed as shown below

$$\text{Salience} = 1 - \frac{\left(\text{LS}_{\text{tr}} - \text{LS}_{\text{mf}}\right)}{\text{LS}_{\text{tr}}} \tag{4.9}$$

where $LS_{tr}$ and $LS_{mf}$ are the local strengths of the top-ranked and the melodic (voice) F0 candidates respectively.

### 4.2.3 Experimental Setup

To keep the comparison between PDAs as fair as possible, the F0 search range is kept fixed for all PDAs for each target signal i.e. from 70 to 500 Hz for the low base F0, and from 150 to 700 Hz for the high base F0. All the PDAs use the same fixed analysis frame-length chosen so as to reliably resolve the harmonics at the minimum expected F0 (4 times the maximum expected period). For the low and high base-F0 target signals these are 57.1 and 26.7 ms respectively. The frequency domain PDAs do not use spectral content above 2.7 kHz since only the first three vowel formants were used for synthesis. Each PDA detects F0 every 10 ms resulting in 2013 estimates for each target signal case. Further, we used the optimal parameter settings for our context, as described in the previous section, for each PDA. The PDA parameter settings were kept fixed for the low and the high base F0 targets, except for the PM PDA, where the number of ideal spectral components is 10 and 6 for the low and high base F0 targets respectively. This is done to preserve the optimality of its performance.

### 4.2.4 Results

The comparison of PDAs, based on the pitch accuracy (PA) values expressed as percentages, appears in Table 4.1. We see from the table that all the PDAs display very high PA values for the clean signals. This indicates that they are all working under optimal parameter settings for monophonic signals. The addition of a single harmonic, tonal interference, a close approximation of the stroke tun, results in a severe degradation of the PA values for all PDAs except TWM, as indicated by row 2 of the table. In all cases with interference, except for the combination of the vowel at high base pitch and the interference with 7 harmonics, the PA values of TWM are the highest. It can also be seen that the TWM PAs, for the same number of interference harmonics, are lower for the target at higher base pitch.

The results of Table 4.1 indicate that the TWM is least sensitive to harmonic interference when the number of interference partials is low. While the PA values of the other PDAs appear to remain relatively constant with the changing spectral structure of the interference, the TWM PA values display a significant decrease. This decrease in accuracy with increase in number of interference partials is more prominent for the target with a higher base F0.

Since the identical smoothness cost was used for all PDAs, a better performance indicates a superior measurement cost, or equivalently, better salience of the underlying melodic pitch. To confirm this, the salience of the true F0 is computed for each frame using Equation 4.9. If the target F0 is not present in the list of candidates then its salience is set to 0. Figure 4.2 displays the melodic pitch salience computed by each PDA across the signal duration for the case of the target with low base pitch and a single harmonic interference. We observe that the salience values of each of the PDAs, except for TWM, are severely degraded around the onset of interference strokes. The corresponding degradation in target F0 salience for TWM is relatively mild. This is consistent with its performance in terms of PA. The more salient the melodic (voice) pitch, the better is the prospect of accurate reconstruction by DP-based post-processing, especially when the interference is intermittent.

Table 4.1: PA values (in percentage) of the different PDAs for the various target and interference signals

| | **Base F0 = 150 Hz** | | | | | **Base F0 = 330 Hz** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **ACF** | **YIN** | **SHS** | **PM** | **TWM** | **ACF** | **YIN** | **SHS** | **PM** | **TWM** |
| Clean | 100 | 99.7 | 95.6 | 95.0 | 99.4 | 98.1 | 98.9 | 98.7 | 95.9 | 97.4 |
| 1 harmonic | 66.5 | 68.4 | 69.3 | 64.7 | 97.8 | 68.9 | 74.0 | 78.0 | 65.0 | 91.9 |
| 3 harmonics | 66.8 | 68.9 | 63.0 | 58.5 | 94.9 | 68.4 | 73.0 | 73.7 | 61.4 | 89.7 |
| 5 harmonics | 67.6 | 69.1 | 60.6 | 55.7 | 92.6 | 72.1 | 76.3 | 69.9 | 57.6 | 76.9 |
| 7 harmonics | 68.0 | 69.6 | 58.8 | 50.7 | 88.4 | 73.2 | 76.4 | 68.3 | 62.3 | 54.9 |

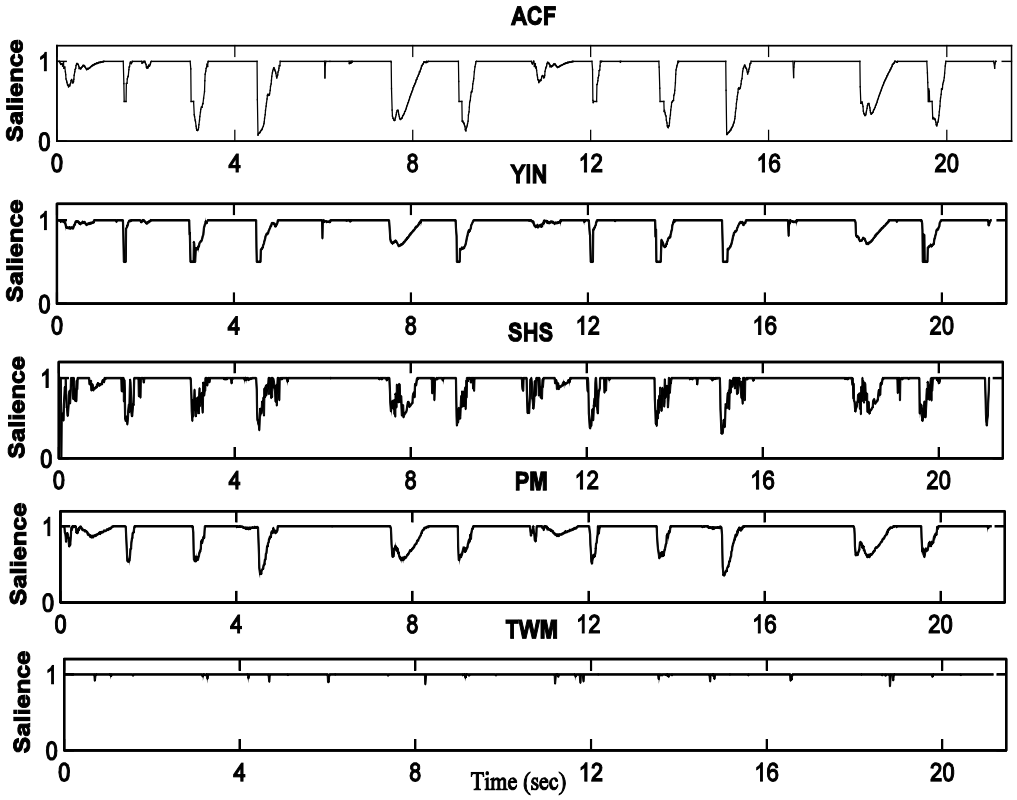


Figure 4.2: Salience contours of the target F0 for different PDAs for the target at low base F0 added to an intermittent interference with a single harmonic.

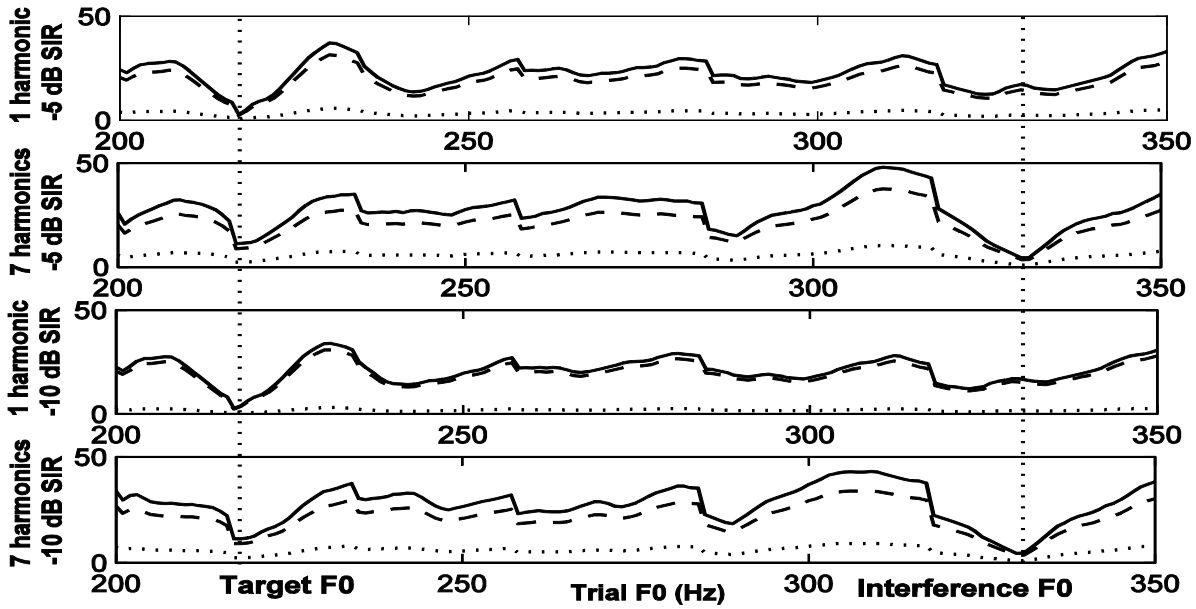


Figure 4.3: Plots of Term1 (dashed curve), Term2 (dotted curve) and $Err_{p \rightarrow m}$ (solid curve), vs. trial F0 for a single frame for the target at high base pitch for interferences with 1 and 7 harmonics added at -5 and -10 dB SIR

### 4.2.5 Discussion

The robustness of TWM to sparse tonal interferences and its sensitivity to the interference spectral structure can be attributed to the peculiar form of the TWM error defined in Equation 4.8. $\mathrm{Err}_{p\rightarrow m}$ can be viewed as a combination of two terms as shown below.

$$\mathrm{Err}_{p\rightarrow m} = \mathrm{term1} + \mathrm{term2};\ \ \mathrm{term1} = \sum_{n=1}^{N} \frac{\Delta f_n}{(f_n)^p};\ \ \mathrm{term2} = \sum_{n=1}^{N} \left( \frac{a_n}{A_{\max}} \right) \left( q \frac{\Delta f_n}{(f_n)^p} - r \right) \tag{4.10}$$

term1, called the frequency mismatch error, is only affected by location of partials. That is, it is maximum when $\Delta f/f$ is large. term2 is affected by relative amplitudes of the partials further weighted by the frequency mismatch error leading to minimum error when $\Delta f/f$ is small and $a_n/A_{\max}$ is large. Therefore, for a given trial F0, specific emphasis is placed on the presence of harmonics at the expected frequency locations.

To illustrate the importance of this point consider Figure 4.3, which displays plots of term1, term2 and $\mathrm{Err}_{p\rightarrow m}$ vs. trial F0, for a single frame of a target signal at high base pitch to which are added interferences with 1 and 7 harmonics at -5 and -10 dB SIR. In this frame, the target F0 is 217 Hz while the interference F0 is 330 Hz. For all four cases, we can clearly see that $\mathrm{Err}_{p\rightarrow m}$ is dominated by term1 and term2 is of lesser significance. The dominance of term1, which is only affected by partial locations, is responsible for the robustness of TWM to sparse tonal interferences.

For the interference with a single harmonic, the global minimum in $\mathrm{Err}_{p\rightarrow m}$ occurs at the target F0, independent of SIR, and is much lower than the value of $\mathrm{Err}_{p\rightarrow m}$ at the interference F0. This occurs because all the target harmonics result in low frequency mismatch terms but the numerous missing interference harmonics lead to large frequency mismatch terms irrespective of the overall strength of the interference. As the number of interference and target harmonics become comparable, the value of $\mathrm{Err}_{p\rightarrow m}$ at the interference F0 decreases in value and the global minimum shifts to the interference F0, again independent of SIR. This occurs because now all the interference harmonics result in lower frequency mismatch. There is a slight increase in the error at the target F0 due to some of the weaker target harmonics becoming distorted by interaction with the interference harmonic lobes in their close vicinity resulting in shifted or suppressed target harmonics. The low PA value of TWM for the case of the target at high base pitch combined with the interference having 7 harmonics is thus caused primarily by the number of interference harmonics, as compared to target harmonics, rather than their strengths.

In contrast, there is no significant variation in the PA values of the other PDAs with an increase in the number of interference harmonics. The SHS and PM PDAs compute an error measure that depends on the overall difference between the actual spectrum and an assumed harmonic spectrum at the trial F0. The overall spectral mismatch at the target F0 would be influenced by the presence of the interference harmonics depending chiefly on the interference power, and independent of whether it is concentrated in a few large partials or distributed over several smaller partials. This also holds true for the ACF PDA, since maximizing the ACF is related to finding that spectral comb filter, which passes the maximum signal energy (Wise et. al. 1976). This relation can also be extended to the YIN PDA if we consider it to be derived from the ACF PDA, as seen in Equation 4.3.

#### *4.2.5.1 Comparison of Salience of Different PDAs*

Figure 4.4 displays the distribution of salience values of the melodic pitch for different PDAs, in terms of histograms, for the target signal with high base F0 when intermittent harmonic interferences with 1 and 7 harmonics are present at local SIRs of -5 and -10 dB. The spread of salience for a single interference harmonic for TWM shows a negligible change when the SIR drops from -5 to -10 dB i.e. from the 1$^{st}$ to the 3$^{rd}$ row. However, the corresponding TWM histograms for the interference with 7 harmonics show a significant leftward shift in the spread of salience with a drop in SIR i.e. from the 2$^{nd}$ to the 4$^{th}$ row. This indicates that when salience of the voice-F0 for the TWM PDA is only negatively affected by a drop in SIR when the number of interference and target harmonics becomes comparable. On the other hand, the salience for the other PDAs is clearly adversely affected by increasing SIR i.e. from the 1$^{st}$ to the 3$^{rd}$ row and from the 2$^{nd}$ to the 4$^{th}$ row, more or less independent of the spectral distribution of the interference. This indicates that the TWM PDA particularly robust to strong, sparse tonal interferences such as tonal table strokes.

## 4.3 Extension of TWM Algorithm to Multi-F0 Analysis

Here we investigate the extension of the original TWM algorithm, found to be more robust to harmonic interferences in the previous section, to multi-F0 analysis and compare it with another multi-F0 analysis method. Some of the stages in this design also serve to reduce computation time significantly.

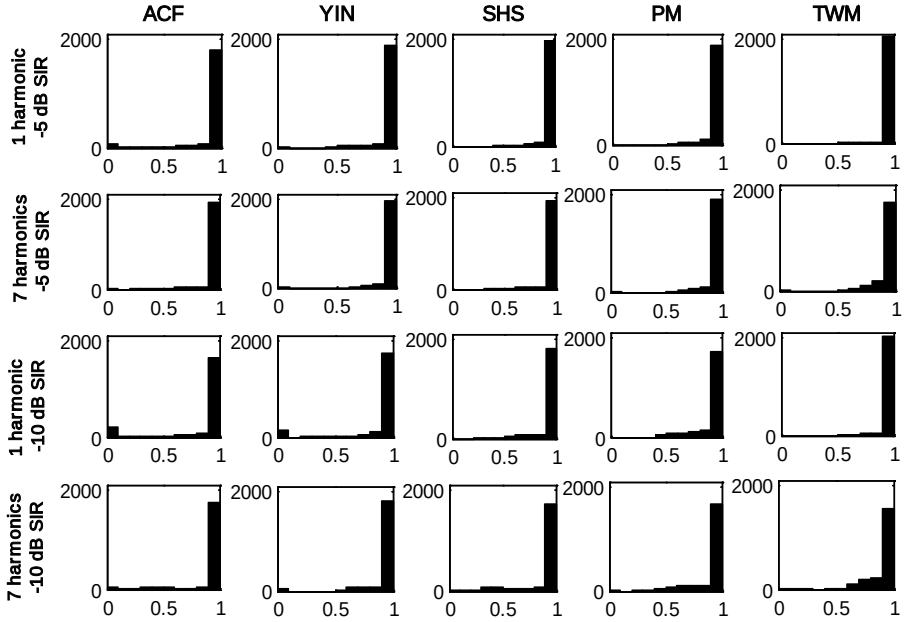


Figure 4.4: Salience histograms of different PDAs for the target with high base F0 added to an intermittent interference with 1 and 7 harmonics, at -5 and -10 dB SIRs.

## 4.3.1 Design of Multi-F0 Estimation

### *4.3.1.1 Stage 1- F0 Candidate Identification*

One of the reasons for the large computation time taken by the current implementation is that the TWM error is computed at all possible trial F0s ranging from a lower ($F0_{low}$) to upper ($F0_{high}$) value with very small frequency increments (1 Hz). Cano (1998) states that it would be faster to first find possible candidate F0s and apply the TWM algorithm to these ones only. The list of possible candidate F0s is a combination of the frequencies of the measured spectral peaks, frequencies related to them by simple integer ratios (e.g., 1/2, 1/3, 1/4), and the distances between well defined consecutive peaks.

So the first modification made is to compute possible candidate F0s from the detected spectral peaks and only compute TWM error at these F0s. We include all measured peaks and their sub-multiples (division factors ranging from 2 to 10) that lie within the F0 search range.

### *4.3.1.2 Stage 2 – Candidate Pruning*

Computation of F0 candidates as sub-multiples of sinusoids will typically result in a large number of F0 candidates clustered around (sub) multiples of the F0s of pitched sound sources, all having low values of TWM error. However, since we enforce a reasonable upper limit on the number of candidates (10-20), this may not allow some locally degraded melodic

F0 candidates with higher TWM error values. One way to reduce this error would be to increase the upper limit on the number of candidates but this would again increase the processing time and is avoidable.

Instead, the next modification involves the sorting of F0 candidates in ascending order of their TWM error values and the subsequent pruning of F0 candidates in close vicinity of each other i.e. the candidate with the least error value is retained and all candidates within a 3% (50 cent) vicinity having higher error values are deleted. This is done so as to include only the most relevant F0 candidates in the final list. Only the top 10 candidates and their corresponding salience (normalized $Err_{TWM}$) values are chosen for further processing.

### 4.3.2 Evaluation of Multi-F0 Estimation Algorithm

In the evaluation of our multi-F0 extraction module we used complex tone mixtures created for the study by Tolonen and Karjalainen (2000) in which two harmonic complexes, whose F0s are spaced a semitone apart (140 and 148.3 Hz), are added at different amplitude ratios of 0, 3, 6 and 10 dB. In their study, they showed that the identification of a peak at the weaker F0 candidate in an enhanced summary autocorrelation function (ESACF) progressively gets worse; while at 6 dB it is visible as a shoulder peak, at 10 dB it cannot be detected. For our study an evaluation metric of 'percentage presence' is defined as the percentage of frames that an F0 candidate is found within 15 cents of the ground-truth F0. We found that for all mixtures (0, 3, 6 and 10 dB) both F0s (140 and 148.3 Hz) were always detected by our multi-F0 extraction system i.e. percentage presence = 100%. We also found that the modifications made to the TWM algorithm to enhance multi-F0 extraction performance reduced the processing time by more than a factor of 2 without compromising accuracy. This is discussed in more detail in Section 7.1.1.3.

## 4.4 Summary

In this chapter we have designed a multi-F0 extraction analysis stage by extending a known monophonic algorithm, experimentally verified to be robust to strong, pitched interference.

### 4.4.1 Salience Function Selection

As mentioned before most PDAs are classified as either spectral or temporal. Recent multi-F0 extraction systems have used a spectro-temporal approach to F0 candidate estimation (Klapuri, 2008; Li & Wang, 2005; Tolonen & Karjalainen, 2000). This involves the

computation of independent correlation functions on multiple frequency channels (such as a multi-band ACF), usually motivated by an auditory model. Although this may overcome the high amplitude interference problem by allowing the weight of each channel to be adjusted to compensate for amplitude mismatches between spectral regions, such an approach requires suitable decisions to be made on the frequency bands and associated channel weighting. Such decisions may again be dependent on the nature of accompanying instruments.

Our choice of salience function is the Two Way Mismatch (TWM) error, as described originally by Maher & Beauchamp (1994) which, to the best of our knowledge, has not been previously explored in the context of melody extraction from polyphony. The TWM PDA qualifies as a spectral method for pitch detection. However it is different from the "pattern matching" methods (i.e. those that minimize the squared error or maximize the correlation between the actual and idealized spectra) in that it minimizes an unconventional spectral mismatch error which is a particular combination of an individual partial's frequency deviation from the ideal harmonic location and its relative strength. As described by Maher & Beauchamp (1994), this error function was designed to be sensitive to the deviation of measured partials/sinusoids from ideal harmonic locations. Relative amplitudes of partials too are used in the error function but play a significant role only when the aforementioned frequency deviations are small (see Section II.A. in (Maher & Beauchamp, Fundamental frequency estimation of musical signals using a two-way mismatch procedure, 1994)). Unlike the multi-band ACF, the TWM error computation does not require decisions on bands and weighting but is primarily dependent on the number of predicted harmonics ($N$) for a given F0. The choice of $N$ depends on the known or assumed spectral characteristics of the signal source. We have tuned $N$ for biasing the error function in favor of spectrally rich musical sources, such as the singing voice, by predicting harmonics upto 5 kHz (a previously used upper spectral limit for dominant harmonics of typical melody lines (Dressler, 2006; Goto, 2004). Although other parameters are used in the computation of the TWM error function for monophonic signals, these are unchanged in our implementation.

We have confirmed, using simulated signals, that in the presence of strong, spectrally sparse, tonal interferences, the melodic (voice-F0) candidate indeed displayed significantly higher salience on using the TWM PDA as compared to other harmonic matching or correlation-based PDAs. This was attributed to the dependence of TWM error values on the frequency extent of harmonics as opposed to the strengths of the harmonics (which is the case with most 'harmonic-sieve' based methods (de Cheveigne, Multiple F0 Estimation, 2006).

This has the advantage that F0s belonging to the spectrally-rich singing voice (having gentler roll-off than common pitched accompanying instruments such as the piano and the flute (Brown, 1992), are characterized by lower TWM errors i.e. better salience.

### 4.4.2 Reliable F0-Candidate Detection

In the presence of strong, harmonically rich (comparable spectral roll-off as the voice) accompanying instruments, neither TWM nor multi-band ACF can provide any solution to the "confusion" between instrument and voice F0s that inevitably occurs (Klapuri, 2008). In fact multi-band ACF-based methods are more susceptible to confusion errors in such situations since they require the isolation of spectral bands that are dominated by the voice-F0 for even the detection of the voice-F0 candidate, let alone a high voice-F0 salience. Such bands may not even exist if the accompaniment harmonics dominate all spectral bands. For reliable detection of multiple-F0s systems that use an iterative 'estimate-cancel-estimate' or joint multi-F0 estimation have been proposed (de Cheveigne, 2006).

Our multi-F0 extraction module is able to reliably extract target F0 candidates in the presence of pitched interference without having to resort to the iterative or joint estimation approaches. This is achieved by the distinct separation of the F0 candidate identification and salience computation parts. Rather than computing a salience function over a range of trial F0 and then picking F0 candidates as the locations of maxima of this function, we first identify potential candidates by an independent method that selects all integer sub-multiples of well-formed detected sinusoids and only compute the salience function at these candidates. This ensures that the voice-F0 candidate will be selected (and therefore actively considered in the next stage of predominant-F0 trajectory formation) even if a single well-formed higher harmonic of the voice-F0 is detected.

# Chapter 5

## Predominant-F0 Trajectory Extraction

The objective of the module discussed in this chapter is to accurately detect the predominant-F0 contour through the F0 candidate space. The design of this stage usually utilizes the F0 candidate salience values output from the multi-F0 analysis stage and further imposes pitch smoothness constraints. In this stage most algorithms take one of two approaches. The first approach involves finding an optimal path through the F0 candidate space over time by dynamically combining F0 salience values (also called measurement cost) and smoothness constraints (also called smoothness cost) using methods either based on the Viterbi algorithm (Forney, 1973) or dynamic programming (DP) (Ney, 1983; Secrest & Doddington, 1982). The local costs usually involve the detected salience values from the multi-F0 analysis chapter. Fujihara (2006) additionally augments the salience values with a vocal likelihood value generated from classifiers trained on vocal and non-vocal data. Smoothness costs depend on

the magnitude of F0 transition and can be Laplacian (Li & Wang, 2007) or Gaussian (Fujihara, Kitahara, Goto, Komatani, Ogata, & Okuno, 2006) in shape. Rynnanen & Klapuri (2008) defines smoothness costs using a musicological model that involves key estimation and note bigram probabilities. The second approach to melody identification uses variants of the partial tracking (PT) algorithm used in the classical sinusoidal modeling approach (McAulay & Quatieri, 1986). While PT was originally applied to sinusoidal partials, here they are applied to the F0 candidate space to form multiple F0 trajectories/contours (Cancela P. , 2008; Cao, Li, Liu, & Yan, 2007; Dressler, 2006; Fernandez-Cid & Casajus-Quiros, 1998; Goto, 2004; Paiva, Mendes, & Cardoso, 2006). The only criterion in linking an F0 candidate to a F0 'track' is the frequency proximity of the candidate to the last tracked F0. The salience of the F0 is used to evaluate track strength and is not used in the linking process except sometimes in the case of when multiple tracks are competing for a single F0 candidate. The final melodic contour is usually chosen as that track with the greatest salience/strength.

In our melody identification module we use the DP framework. The use of DP in melody extraction is attractive since it finds the optimal path by combining trajectory forming and melodic contour identification in one computationally-efficient, global framework i.e. a black box that outputs a single F0 contour given suitably defined local and smoothness costs. PT, on the other hand, first forms trajectories using local frequency proximity and subsequently identifies melodic tracks or fragments. Here multiple trajectories may be formed through different musical instrument F0 contours and their (sub) multiples leading to a densely populated track space (Paiva, Mendes, & Cardoso, 2006).

We next describe and evaluate the application of DP in our predominant-F0 trajectory extraction module[1] (referred to as single-F0 tracking), wherein we describe the design of a novel smoothness cost function. Situations in which this module is expected to suffer from irrecoverable degradations are then identified. Enhancements, in terms of dual-F0 tracking, within the DP framework are then described that may enable the retrieval of previously inaccessible melodic information. Finally we describe a voice-pitch identification framework that uses a novel feature that enables the identification of the voice pitch from the dual-F0 tracking output by detecting the temporal frequency-instability in the voice harmonics.

[1] The initial investigation on the use of DP for predominant F0 extraction was done together with Ashutosh Bapat.

## 5.1 Dynamic Programming-based Optimal Path Finding

PDAs typically employ post-processing of the local F0 estimates based on an assumed smoothness of the F0 contour. For instance, median filtering can easily correct for isolated F0 jumps, while low-pass filtering corrects corrupted values by interpolating from temporally adjacent regions. However, such post-processing methods will be ineffective in reconstructing underlying melodic pitch variation obscured by long-duration perturbations caused by tonal interferences, such as that in Figure 2.8.

A more systematic approach to post-processing are dynamic-programming (DP)-based methods, applied to continuously voiced segments, which take into account candidate F0 estimates from the PDA other than just the locally optimal estimate. This amounts to combining suitably defined local measurement and smoothness costs into a global cost, which can then be optimized over a continuous voiced segment by the use of DP. Since DP-based smoothing was first proposed as a post-processing method for F0 contours (Ney, 1983; Secrest & Doddington, 1982), there have been several PDAs that have successfully used it for error correction (Boersma, Accurate Short-term Analysis of the Fundamental Frequency and the Harmonics-to-Noise, 1993; Hermes, 1993; Luengo, Saratxaga, Navas, Hernaez, Sanchez, & Sainz, 2007; Talkin, 1995; Van Hemert, 1988).

### 5.1.1 DP Algorithm

The operation of DP can be explained by considering a state space, as illustrated in Figure 5.1, where, for a given frame ($j$), each state ($p$,$j$) represents a possible F0 candidate. Any F0 contour can be seen as the path (($p$(1),1), ($p$(2),2),…, (p(j),j),…, ($p$($N$),$N$)) through this state space where $p(j)$ is the F0 estimate at the $j^{th}$ frame and $N$ is the total number of frames in the given voiced segment. The measurement cost is the cost incurred while passing through each state i.e. $E(p,j)$ is the measurement cost incurred at frame $j$ for candidate $p$. For the time evolution of F0, a smoothness cost $W(p,p')$ is defined as the cost of making a transition from state ($p$,$j$) to state ($p'$,$j+1$) where $p$ and $p'$ can be any candidate values in successive frames only. A local transition cost $T$ is defined as the combination of these two costs over successive frames as shown below.

$$T(p(j+1), p(j), j+1) = E(p(j+1), j+1) + W(p(j), p(j+1)) \quad (5.1)$$

Finally, an optimality criterion to represent the trade-off between the measurement and the smoothness costs is defined in terms of a global transition cost ($S$), which is the cost of a path

passing through the state space, by combining local transition costs across a segment (singing spurt) with *N* frames, as shown below,

$$S = \sum_{j=1}^{N-1} T(p(j+1), p(j), j+1) \tag{5.2}$$

The path, or F0 contour, with the minimum global transition cost, for a given singing spurt, is then the estimated F0 contour. A computationally efficient way of computing the globally optimal path, by decomposing the global optimization problem into a number of local optimization stages, is described by Ney (1983).

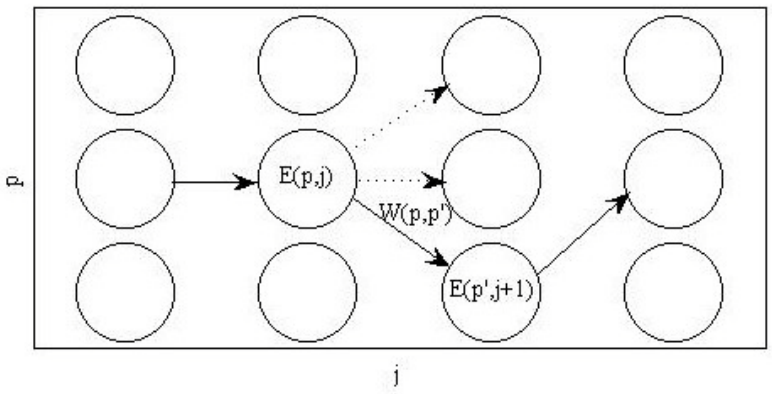


Figure 5.1: State space representation of dynamic programming. The states and transitions are labeled by their costs. Possible transitions (*dotted lines*) for state (p,j) and the minimum cost path (*solid lines*) found by DP are shown.

### 5.1.2 Smoothness Cost

With a view to determine a smoothness cost that is musicological knowledge-based, a distribution of inter-frame F0 transitions was obtained from F0 contours extracted from 20 minutes of continuous monophonic singing segments of two male and two female Indian semi-classical singers. The normalized distribution, indicated by the solid line in Figure 5.2, indicates that most F0 transitions are in a close neighborhood, and the probability of a given transition decreases rapidly (but nonlinearly) with increasing magnitude. At larger magnitudes of F0 transition, the probability falls off very slowly to near zero.

The smoothness cost must reflect the characteristics of typical voice pitch transitions and should be designed based on the following musical considerations. 1) Since musical pitches are known to be logarithmically related, such a cost must be symmetric in log-space. 2) Smaller, more probable pitch transitions (found to be <2 ST from the solid line in Figure

5.2) must be assigned a near zero penalty since these are especially common over short durations (such as the time between consecutive analysis time instants). 3) The cost function should steeply (non-linearly) increase from probable to improbable pitch transitions, and apply a fixed ceiling penalty for very large pitch transitions.

One smoothness cost found in the DP formulation of (Boersma, 1993) is given below

$$W(p, p') = \mathrm{OCJ} \times \log_2\left(\frac{p'}{p}\right) \tag{5.3}$$

where $p$ and $p'$ are the pitch estimates for the previous and current frames, *OJC* is a parameter called OctaveJumpCost. Higher values of *OJC* correspond to increasing penalties for the same pitch transitions. This function is displayed as the dashed line in Figure 5.2. This function does not satisfy criteria 2 and 3 described before.

We propose an alternative cost function that satisfies all 3 of the required criteria. This function is Gaussian in nature, and is defined as

$$W(p, p') = 1 - e^{\frac{-\left(\log_2(p') - \log_2(p)\right)^2}{2\sigma}} \tag{5.4}$$

where $p$ and $p'$ are F0 candidates for the previous and current frames. This function is indicated by the dotted line in Figure 5.2. The Gaussian cost function applies a smaller penalty to very small, highly likely, F0 transitions than the former, as indicated by its flatter shape around this region in Figure 5.2. A value of $\sigma$ = 0.1 results in a function that assigns very low penalties to pitch transitions below 2 semitones. Larger rates of pitch transition (in the 10 ms frame interval chosen in this work) are improbable even during rapid singing pitch modulations and are penalized accordingly. With this in mind, the second of the two cost functions would be a better choice for the smoothness cost and is used in the subsequent experiments.

### 5.1.3 System Integration and Evaluation

The application of DP for single-F0 tracking in our melody identification module is quite straightforward. The local measurement cost for each pitch candidate is given by the normalized TWM error of the F0 candidates obtained in the multi-F0 extraction stage. The smoothness cost between two F0 candidates in adjacent frames is given by Equation 5.4.

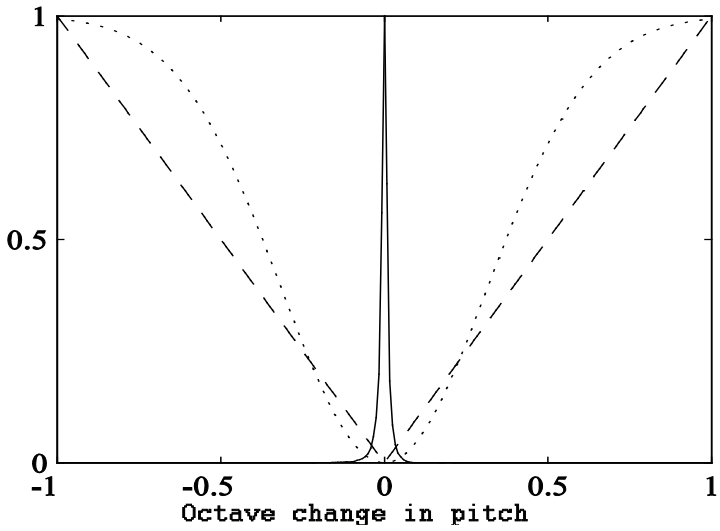


Figure 5.2: Normalized distribution (solid curve) of log pitch transitions between adjacent frames (at 10 ms intervals) computed from true pitch contours of 20 min. of singing by male and female singers. Log cost function (dashed curve) and Gaussian cost function ($\sigma$ = 0.1) (dotted curve) respectively.

Table 5.1: PA values (in percentage) of the different PDAs after DP-based post-processing for the mixtures of the simulated target and different simulated interference signals

| | **Base F0 = 150 Hz** | | | | | **Base F0 = 330 Hz** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **ACF** | **YIN** | **SHS** | **PM** | **TWM** | **ACF** | **YIN** | **SHS** | **PM** | **TWM** |
| Clean | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 1 harmonic | 64.9 | 65.8 | 66.3 | 64.3 | 100 | 67.7 | 62.7 | 71.8 | 70.0 | 100 |
| 3 harmonics | 62.7 | 75.2 | 62.7 | 60.0 | 100 | 68.0 | 65.5 | 68.4 | 66.0 | 100 |
| 5 harmonics | 73.2 | 75.4 | 62.5 | 53.2 | 100 | 80.0 | 72.8 | 62.6 | 66.6 | 86.0 |
| 7 harmonics | 74.5 | 75.9 | 53.4 | 47.5 | 93.8 | 82.0 | 82.0 | 57.1 | 66.0 | 60.0 |

#### *5.1.3.1 Simulated Signals*

The DP stage was applied to the output of the different PDAs for the experiment described in Section 4.2, which uses simulated voice and interference signals. Table 5.1 displays the PA results of applying DP-based post-processing to all the PDAs for the same synthetic signals as described before. For the clean target, first row of the table, we can see that the combination of any PDA with DP-based post-processing results in 100% accuracy. This indicates that the choice of measurement costs for each PDA is appropriate since DP is able to correct all the errors when there is no interference signal present. In the presence of the

tonal interference, it is clear that the best results, indicated by the highest PA, are obtained for the combination of TWM and DP, except for the case of the target at high base F0 and the interference with 7 harmonics.

We can also see that in some cases DP reduces the PA further, as in the case with the ACF PDA for the target at low base F0 and the interference with 1 and 3 harmonics. This occurs because of long duration persistent errors i.e. a large number of non-target F0 candidates that are successively more salient than the target F0 candidate, resulting in DP finding an optimal path through the erroneous pitch estimates.

#### *5.1.3.2 Real Signal*

The motivation for the design of the synthetic signals used in Table 5.1 (the experiment of Section 4.2) was the degradation in performance, caused by a strong harmonic interference, of the ACF pitch tracker operating on a real music signal, as described in Figure 2.8. We now compare the performance of the ACF and TWM PDAs before and after DP on the same real music signal, in Figure 5.3 to see if the results of the experiment on simulated signals with harmonic interference can be extended to real signals. From Figure 5.3.a we can see that the ACF PDA before DP makes errors at the impulsive and tonal tabla stroke locations. DP corrects the impulsive errors but further degrades the pitch contour at the tonal stroke location. On the other hand the TWM PDA, in Figure 5.3.b, is unaffected by the tonal stroke and makes only one error at an impulsive stroke location which is further corrected by DP. We can judge the correctness of the output pitch contour by looking at its overlap with the voice harmonic in the spectrogram. This indicates that the results of our experiments on simulated signals with harmonic interferences can be extended to real music signals.

## 5.2 Shortcomings of Single-F0 Tracking

The above melodic identification module may output an (partially) incorrect melody when either the measurement and/or the smoothness costs are in favor of the accompanying instrument F0 rather than the melodic F0. The bias in measurement cost occurs when an accompanying, pitched instrument has a salience comparable to that of the voice. This may cause the output pitch contour to incorrectly identify accompanying instrument F0 contour segments as the melody. An example of such an occurrence is seen in Figure 5.4.a. This figure shows the ground truth voice and instrument pitch contours (thin and dashed) along with the F0-contour output by the single-F0 DP algorithm (thick) for an excerpt from a clip of

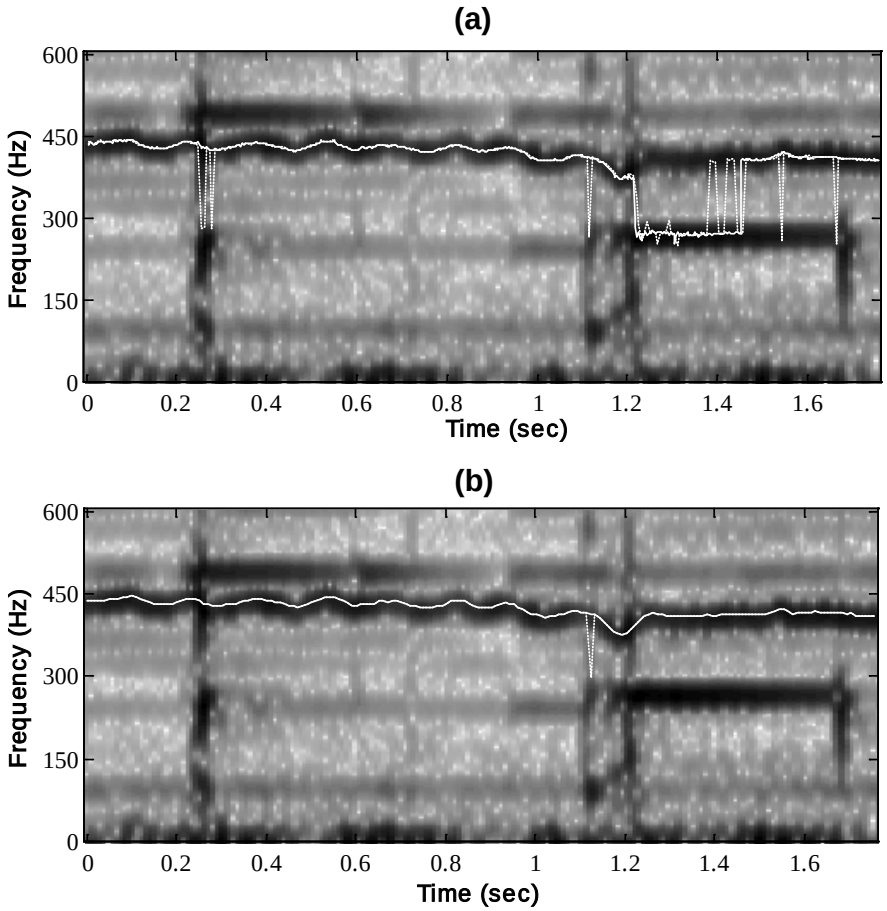


Figure 5.3 Pitch contour detected by (a) modified ACF PDA and (b) TWM PDA, before (dotted) and after (solid) DP, superimposed on the zoomed in spectrogram of a segment of Hindustani music with a female voice, drone and intermittent tabla strokes of Figure 2.8.

Western pop music. It can be seen that the single-F0 contour often switches between tracking the voice and instrument pitch contours.

Smoothness costs are normally biased towards musical instruments which are capable of producing sustained, stable-pitch notes. It is well known that the human voice suffers from natural, involuntary pitch instability called jitter in speech and flutter in singing (Cook, 1999). Further in singing, pitch instability is much more emphasized in the form of voluntary, large, pitch modulations that occur during embellishments and ornaments such as vibrato. So the presence of stable-pitch instruments, such as most keyed instruments e.g. the piano and accordion (especially when the voice pitch is undergoing rapid and large modulations) could also lead to incorrect identification of the melodic fragments. Such errors are more likely to occur when the F0s of the voice and instrument intersect since at the point of intersection, the F0 candidates for both sources are one and the same with a single salience. Around this time

instant, the smoothness cost will more than likely dominate the transition cost given in Equation 5.1.

Interestingly, all of the above conditions may simultaneously occur in Hindustani (North Indian classical) vocal performances where large and rapid voice-pitch modulations are a frequent occurrence. A commonly used accompanying instrument is the harmonium, very similar to the accordion, whose harmonics have a large frequency extent (similar to the voice) and is also keyed. The harmonium accompaniment is meant to reinforce the melody sung by the singer. Since each vocal performance is a complete improvisation without the presence of a musical score, the instrumentalist attempts to follow the singer's pitch, resulting in frequent F0 collisions.

In cases of incorrect melodic identification for PT based approaches, the recovery of the actual melodic tracks may still be possible based on the assumption that correct melodic fragments have been formed but not identified. DP, on the other hand, is forced to output only a single, possibly 'confused', contour with no mechanism for recovering the correct melodic F0s. This information may be retrieved if DP is extended to tracking multiple F0 contours simultaneously.

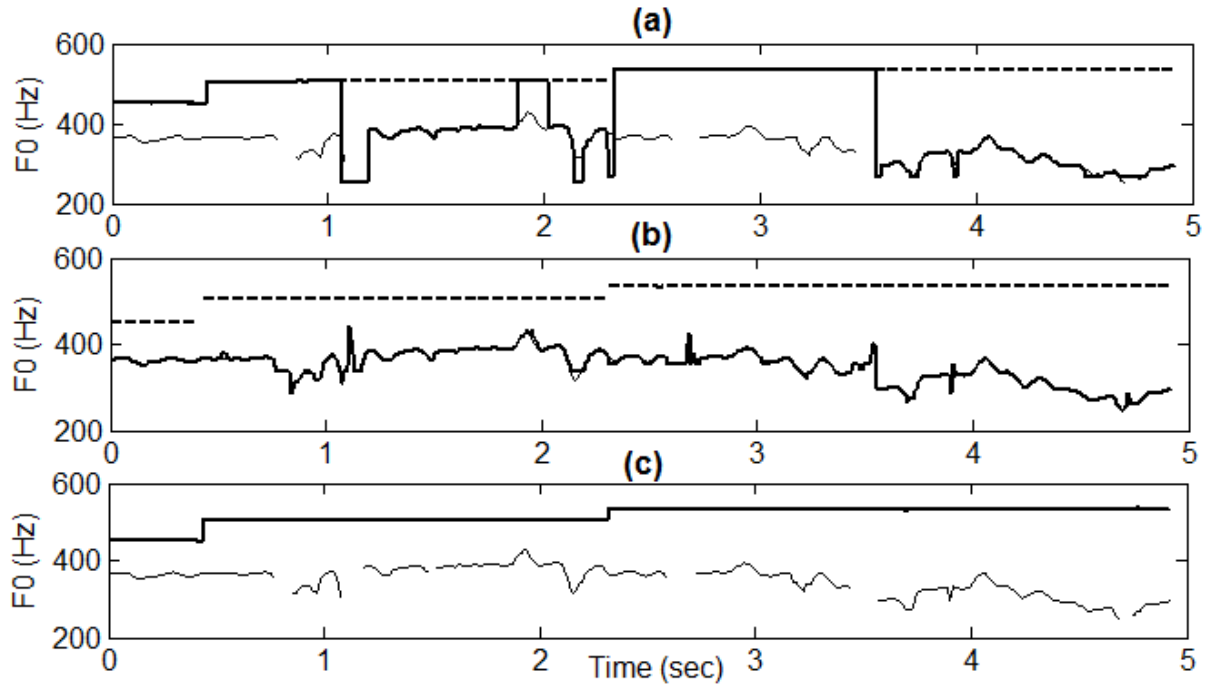


Figure 5.4: Example of melodic recovery using the dual-F0 tracking approach for an excerpt of an audio clip from dataset 2. Ground truth voice-pitch (thin), (a) single-F0 output (thick) and dual-F0 output contours (b) and (c). Single-F0 output often switches between tracking voice and instrument pitches. Each of dual-F0 contours track the voice and instrument pitch separately.

## 5.3 Dual-F0 Tracking Approach

In order to address the F0 contour 'confusion' problem for situations in which the F0 of a dominant pitched accompaniment is tracked instead of the voice-pitch, we propose a novel dynamic programming (DP) based dual-F0 tracking approach in the melody identification stage. Here we describe an enhancement to the DP formulation that simultaneously tracks two F0 contours (hereafter referred to as dual-F0 tracking) with the aim to better deal with accompanying pitched instruments. We restrict ourselves to tracking only two pitches simultaneously on the realistic assumption that in vocal music, there is at most only one instrument which is more dominant than the voice at any time (Li & Wang, 2007).

### 5.3.1 Previous Work on Multiple-F0 Tracking

There is very little precedent in the concept of tracking multiple F0s simultaneously. Every and Jackson (2006) designed a DP framework to simultaneously track the pitches of multiple speakers, where they used average pitch of each speaker as a prior (i.e. in the cost function in DP). While DP itself is a well-established framework, it is the design of the cost functions that dictate its performance on any specific data and task. The singing/music scenario is very different from speech. The design of the measurement and smoothness cost functions therefore require completely different considerations. Maher (1990) experimented with a dual-F0 estimation system for duet tracking. Emiya, Badeau & David (2008) attempted to track piano chords where each chord is considered to be a combination of different F0s. However, both of these approaches made very broad assumptions, mostly based on western instrumental music, that are not applicable here. The former assumes that the F0 ranges of the two sources are non-overlapping and that only two voices are present. The latter was developed specifically for piano chords (F0 groups) and assumed that F0 candidates only lie in the vicinity of note locations based on the 12-tone equal tempered scale and that chord transitions can only occur to subsets of chords.

Of greater relevance is an algorithm proposed by Li and Wang (2007) extends the original algorithm by Wu, Wang and Brown (2003) (designed for multiple speakers) to track the F0 of the singing voice in polyphonic audio. Their system initially processes the signal using an auditory model and correlation-based periodicity analysis, following which different observation likelihoods are defined for the cases of 0, 1 and 2 (jointly estimated) F0s. A hidden Markov model (HMM) is then employed to model, both, the continuity of F0 tracks and also the jump probabilities between the state spaces of 0, 1 or 2 F0s. The 2-pitch

hypothesis is introduced to deal with the interference from concurrent pitched sounds where all possible pairs of locally salient F0 candidates are considered. This can lead to the irrelevant and unnecessary tracking of an F0 and its (sub)-multiple, which often tend to have similar local salience. Also when two pitches are tracked the first (dominant) pitch is always considered to be the voice pitch. All model parameters are learnt from training data.

In our modifications to DP for dual-F0 tracking we use the joint TWM error as the measurement cost of the F0 candidate pair and also a novel harmonic relationship constraint to avoid the tracking of an F0 candidate and it's multiple, since this would defeat the purpose of using DP to track the F0 of multiple distinct sources. These new enhancements are described next.

### 5.3.2 Modifications in DP for Dual-F0 Tracking

We extend our previously described single-F0 tracking DP algorithm to track ordered F0 pairs called nodes. The additional F0 members of the node help to better deal with the accompanying pitched instrument(s). If we consider all possible pairs of F0 candidates the combinatory space becomes very large (Number of permutations of F0 pairs formed from 10 F0 candidates is ${}_{10}P_2$ = 90 permutations) and tracking will be computationally intensive. More importantly, we may end up tracking an F0 and its (sub)-multiples rather than two F0s from separate musical sources. Our method to overcome this is to explicitly prohibit the pairing of harmonically related F0s during node generation. Specifically, two local F0 candidates (*f1* and *f2*) will be paired only if

$$\min_k \left( \left| f_1 - k.f_2 \right| \right) > T; \quad k.f_2 \in \left[ F_{\text{low}}, F_{\text{high}} \right] \tag{5.5}$$

where *k.f2* represents all possible multiples and sub-multiples of *f2*, *T* is the harmonic relationship threshold and $F_{low}$ and $F_{high}$ are the lower and upper limit on the F0 search range. Using a low threshold (*T*) of 5 cents does not allow F0s to be paired with their multiples but allows pairing of two distinct source F0s that are playing an octave apart, which typically suffer from slight detuning especially if one of the F0 sources is the singing voice (Duan, Zhang, Zhang, & Shi, 2004).

The measurement cost of a node is defined as the jointly computed TWM error of its constituent F0 candidates (Maher, 1990). In the interest of computational efficiency the joint TWM error for two F0 candidates, *f1* and *f2*, is computed as shown below

$$Err_{\mathrm{TWM}}(f_1, f_2) = \frac{Err_{p\to m}(f_1)}{N_1} + \frac{Err_{p\to m}(f_2)}{N_2} + \rho\frac{Err_{m\to p}(f_1, f_2)}{M} \quad (5.6)$$

where *N1* and *N2* are the number of predicted partials for $f_1$ and $f_2$ resp. and *M* is the number of measured partials. The first two terms in Equation 5.6 will have the same values as during the single-F0 TWM error computation in Equation 4.7. Only the last term i.e. the mismatch between all measured partials and the predicted partials of both F0s ($f_1$ and $f_2$), has to be computed. Note that here we use a larger value of $\rho$ (0.25) than before. This is done so as to reduce octave errors by increasing the weight of $Err_{m\to p}$ thereby ensuring that $Err_{TWM}$ for the true F0 pair is lower than that of the pair that contains either of their respective (sub)-multiples.

The smoothness costs between nodes are computed as the sum of smoothness costs between the constituent F0 candidates, given previously in Equation 5.4. A globally optimum path is finally computed through the node-time space using the DP algorithm. Two pitch contours are available in this minimum cost node-path.

### 5.3.3 Evaluation of Dual-F0 Tracking Performance

While a formal evaluation of the dual-F0 tracking performance is given in Chapter 8, here we just provide some illustrative examples to highlight the advantage of this method along with some issues that arise from using this method.

#### *5.3.3.1 Melodic Recovery using Dual-F0 Tracking*

In Figure 5.4.a. we had seen how a single-F0 tracking system could output a 'confused' predominant-F0 contour, which often missed tracking the voice-F0 and tracked the F0 of a dominant pitched accompaniment instead. Figure 5.4.b and Figure 5.4.c respectively show the two output contours from the dual-F0 tracking system when presented with the same case. Here we can see that one of the contours output by the dual-F0 tracking consistently tracks the voice-pitch since the other contour is actively tracking the instrumental pitch trajectory. So it appears that the dual-F0 output is indeed able to recover segments of the voice pitch contour that were lost to its single-F0 tracking counterpart.

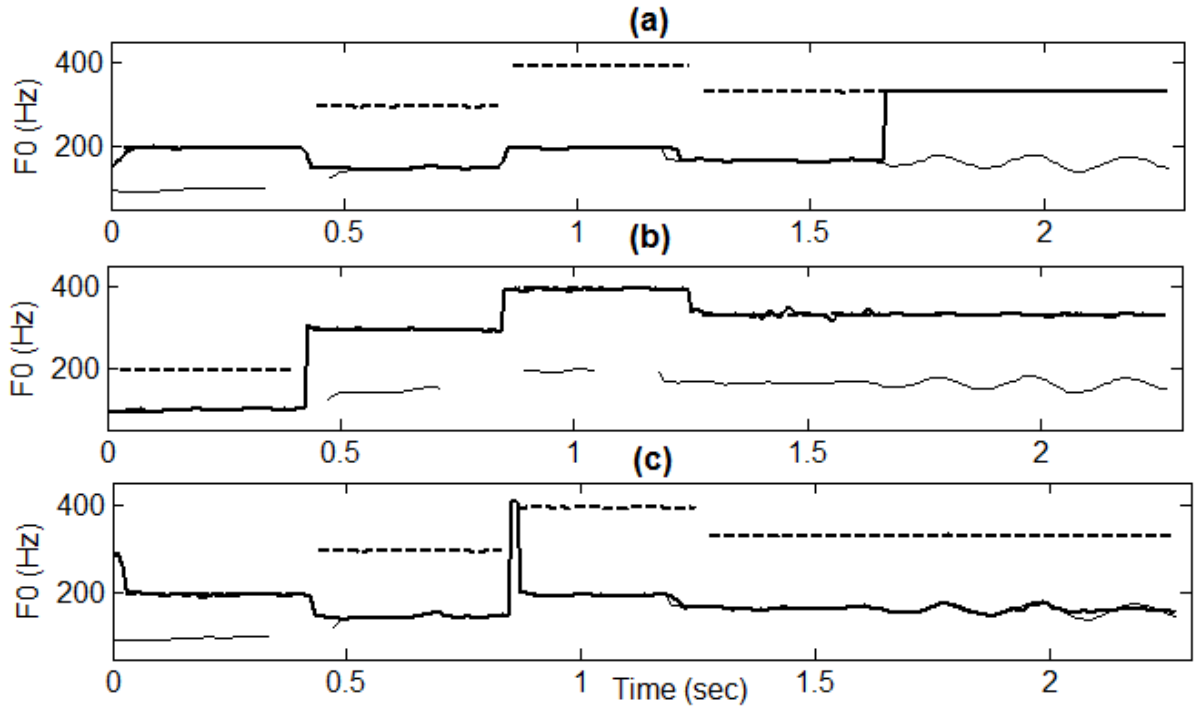


Figure 5.5: Extracted F0 contours (thick) v/s ground truth F0s voice (thin) and organ (dashed) for (a) single-F0 tracking, (b) dual-F0 tracking: contour 1 and (c) contour 2 for a Western example in which note changes occur in the voice and instrument simultaneously.

#### *5.3.3.2 Switching error due to simultaneous transients/silences in voice and instrument*

In some cases we found that both the dual-F0 contours switch between tracking the pitch trajectories of voice and instrument. One signal condition under which this 'switching' is found to occur is when the instrument note change occurs simultaneously with the voice note change. Instances of vocal note changes are often marked by an unvoiced (un-pitched) sung utterance. Figure 5.5 contrasts the performance of the single- and dual-F0 tracking system for an audio clip, which is a mix of a male voice singing lyrics and a synthetic organ. The convention for the different contours is the same as for Figure 5.4. Figure 5.5.a, b & c indicate the contour output by the single-F0 tracking system, contour 1 and contour 2 of the dual-F0 tracking system respectively.

The co-incident gaps in the thin and dashed contours indicate segments locations of note transients. Figure 5.5.a indicates that the output of the single-F0 tracking system is again 'confused' between the F0s of the two sources. However, even the dual-F0 output contours (Figure 5.5.b and Figure 5.5.c) show similar degradation. It can be seen that contour 1 of the dual-F0 tracking system tracks the first note of the voice but then 'switches' to the organ F0 while the reverse happens for contour 2.

The current system cannot ensure that the contours will remain faithful to their respective sound sources across regions in which no clear pitched sound exists. Even if a zero-pitch hypothesis was made during these regions it would be difficult to ensure faithfulness, especially if the next note of the different source rather than the same source is closer to the previous note of a sound source. Further, it is seen occasionally that the slight detuning required for the correct clustering of pitches for the DP node formation does not always hold in the octave separated mixture. In such cases, spurious candidates are tracked instead as can be seen by the small fluctuations in the output contours of the dual-F0 tracking system (Figure 5.5.b. and c.). Such fine errors do not occur in the cases of vocal harmony tracking.

#### *5.3.3.3 Switching error due to F0 collisions*

Dual-F0 contour switching errors also occur in cases where the voice and instrument F0 trajectories collide. To illustrate this problem consider Figure 5.6.b and Figure 5.6.c, which show the ground truth voice and instrument F0s along with the dual-F0 system output for a voice and harmonium mix from this category. For clarity, we have avoided plotting the single-F0 system output pitch contour in Figure 5.6.a, which now only shows the voice and harmonium ground truth values.

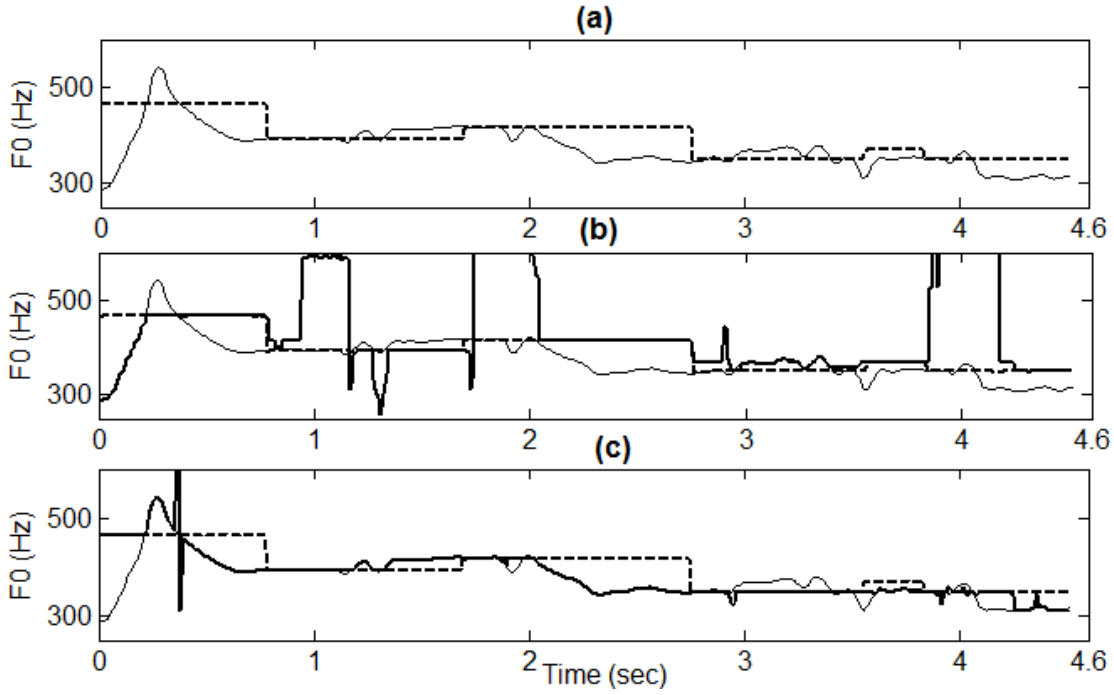


Figure 5.6: (a) Ground truth F0s voice (thin) and harmonium (dashed) v/s (b) extracted F0 contours (thick) dual-F0 tracking: contour 1 and (c) contour 2 for an excerpt of Hindustani music in which there are frequent collisions between the F0 contours of the voice and harmonium.

Figure 5.6.a brings out a peculiarity of Hindustani music that causes F0 collisions to be a frequent rather than a rare occurrence. In this genre of music the harmonium accompaniment is meant to reinforce the melody sung by the singer. There is no score present as each vocal performance is a complete improvisation. So the instrumentalist attempts to follow the singer's pitch contour as best he/she can. Since the harmonium is a keyed instrument, it cannot mimic the finer graces and ornamentation that characterize Indian classical singing but attempts to follow the steady held voice notes. This pitch following nature of the harmonium pitch is visible as the dashed contour following the thin contour in the figure.

At the locations of harmonium note change, the harmonium F0 intersecting with the voice F0 is similar to the previous case during unvoiced utterances when instead of two F0s only one true F0 is present. Here the contour tracking the harmonium will in all probability start tracking some spurious F0 candidates. During these instances the chances of switching are high since when the voice moves away from the harmonium after such a collision, the pitch-proximity based smoothness cost may cause the present contour to continue tracking harmonium while the contour tracking the spurious candidate may start tracking the voice F0.

Cases of the voice crossing a steady harmonium note should not usually result in a switch for the same reason that switching occurred in the previous case. The smoothness cost should allow the contour tracking harmonium to continue tracking harmonium. However the first collision in Figure 5.6, which is an example of voice F0 cross steady harmonium F0, causes a switch. This happened because of multiple conditions being simultaneously satisfied. The crossing is rapid and takes place exactly between the analysis time instants, the harmonium and voice F0 candidates are present on either side of the crossing but slightly deviated from their correct values due to the rapid pitch modulation. As Indian classical singing is replete with such rapid, large pitch modulations such a situation may not be a rare occurrence.

## 5.3.4 Solutions for Problems in Dual-F0 Tracking

### *5.3.4.1 Pitch Correction for Exact Octave Relationships*

In the formation of nodes (F0 pairs) we have explicitly prohibited the pairing of F0 candidates that are harmonically related, within a threshold of 5 cents, in order to avoid the pairing of F0 candidates and their (sub) multiples. However in isolated cases, when the instrument and voice are playing an octave apart, especially with unsynchronized vibrato, there will be

instants when the F0s of both sources will be near-exactly octave related. In such cases, incorrect node formation may lead to erroneous values for one of the two contours output in the minimum-cost node-path. Such a situation was illustrated in Section 5.3.3.2. We next describe a method that uses well-formed sinusoids to correct such errors.

For each F0, output in the minimum-cost node-path, we search for the nearest sinusoids, within a half main-lobe width range (50 Hz for a 40 ms Hamming window), to predicted harmonic locations. Of the detected sinusoidal components we choose the best formed one i.e. with the highest sinusoidality (see Section 3.3.2). The current F0 value is then replaced by the nearest F0 candidate (available from the multi-F0 extraction module) to the appropriate sub-multiple of the frequency of the above sinusoidal component.

Note that for the lower of the two F0s we search for sinusoids only in the vicinity of predicted odd-harmonics. For the specific case of near perfect octave relationship of the two F0s, the measured even-harmonic frequency values, of the lower F0, may be unreliable for correction as they will be the result of overlap of harmonics of both sources. Also, only frames for which the two F0s, in the minimum cost node-path, are separated by some minimum distance (here 50 Hz) are subjected to pitch correction. When the two F0s are close to each other, we found that the above method of correction sometimes resulted in same values for both F0s, thereby degrading performance.

#### *5.3.4.2 Switching Correction for Non-Overlapping Voice and Instrument*

From the previous results it has been shown that when short silences/unvoiced utterances or note transients are present simultaneously for both, the voice and pitched instrumental, sound sources or when the F0 trajectories of the two sources ‘collide’, individual contours tracking the F0s of either source may ‘switch’ over to tracking the F0s of the other source. One simple solution to this problem proposed here is applicable when the F0 contours of the melodic and accompanying instruments do not collide.

Often in western music, for the mixture of the tonal accompaniment and the melody to sound pleasing, their respective pitches must be musically related. Further, as opposed to Indian classical music, western (especially pop) music does not display particularly large and rapid pitch modulations. As a result, F0 collisions most often do not occur. This is also the case with musical harmony and duet songs.

With the above knowledge we implement switching correction by forcing one of the two F0 contours to always be higher or lower, in pitch, than the other F0 contour. To make the

initial decision about which contour is lower/higher than the other we use a majority voting rule across the entire contour.

## 5.4 Selection of One Predominant-F0 Trajectory

The Predominant-F0 extraction module is required to output a single-F0 contour from the dual-F0 DP stage as the final predominant-F0 contour. One possible approach to solving the above problem would be to adopt a source discrimination approach, as proposed by Marolt (2008) which attempts the unsupervised clustering of melodic fragments using timbral features. In such an approach the selection of the final contour after clustering is still unresolved.

Although melodic smoothness constraints are imposed in the dual-F0 tracking system, each of the output contours cannot be expected to faithfully track the F0 of the same source across silence regions in singing or instrument playing. Therefore choosing one of the two output contours as the final output is unreliable. Rather we rely on the continuity of these contours over short, non-overlapping windows and make voice-F0 segment decisions for each of these 'fragments'. Here we make use of the STHE feature described in the appendix (Section A.2) for the identification of a single predominant-F0 contour. This procedure is briefly explained next.

Each of the dual-F0 output contours is divided into short-time (200 ms long) non-overlapping F0 fragments. For each contour segment we build a Harmonic Sinusoidal Model representation, as described in Section 6.2.1. Now we make use of the fact that the voice harmonics are usually relatively unstable in frequency as compared to most keyed instrument harmonics. Therefore for each of these harmonic sinusoidal models we next prune/erase tracks whose standard deviations in frequency are below a specified threshold (here 2 Hz), indicating stability in frequency. The total energy of the residual signal within the analysis window is then indicative of the presence of vocal harmonics. The fragment with the higher energy is therefore selected as the final voice-F0 fragment. This method is expected to fail when the accompanying instrument is also capable of smooth continuous pitch transitions.

## 5.5 Summary and Conclusions

In this chapter we investigated the use of a DP-based path finding algorithm for outputting a single predominant-F0 contour. In the context of melody extraction for vocal performances, it was found that such an approach results in a single, degraded melodic contour when a strong,

pitched accompanying instrument is present. This degradation is caused by the incorrect identification of the instrument pitch as the melody. In order to enable the recovery of the actual melodic contour it is proposed to extend the use of DP to tracking multiple pitch contours simultaneously. Specifically, a system that dynamically tracks F0-candidate pairs, generated by imposing specific harmonic relation-related constraints, is proposed to alleviate the above degradations.

It is found that when the proposed system is evaluated on mixtures of melodic singing voice and one loud pitched instrument the melodic voice pitch is tracked with increased accuracy by at least one of the contours at any given instant. This is an improvement over the previous single-F0 tracking system where the voice pitch was unrecoverable during pitch errors. It is possible that the simultaneous tracking of more than 2 F0s may lead to even better melodic recovery if there is more than one loud, pitched accompanying instrument. However such an approach is not expected to result in as significant an improvement in voice-pitch tracking accuracy as the improvement resulting in the transition from single- to dual-F0 tracking. This hypothesis is based on our premise that in vocal music the voice is already the 'dominant' sound source. On occasion, an accompanying instrument may be more locally dominant than the voice however we feel that the chances that two pitched instruments are simultaneously of higher salience than the voice are relatively small.

A problem pending investigation is that of F0 collisions, such as those in Figure 5.6. Such collisions, found to occur frequently in Indian classical music, induce contour switching and also the same pitch values have to be assigned to both contours during extended collisions. The latter condition can be achieved by pairing F0 candidates with themselves. But an indication of when such an exception should be made is required. It may be possible to investigate the use of predictive models of F0 contours, similar to those used for sinusoidal modeling in polyphony (Lagrange, Marchand, & Rault, 2007), and also possibly musicological rules to detect F0 collisions.

# Chapter 6

## Singing Voice Detection in Polyphonic Music

The task of identifying locations of singing voice segments within the predominant-F0 contour is usually a part of the melody extraction system but may also be an independent process. In either case we refer to the problem as Singing Voice Detection (SVD) henceforth. Examples of SVD integrated into melody extraction systems are systems that use note models such as HMMs (Li & Wang, 2005; Rynnanen & Klapuri, 2008), which also include a silence or zero-pitch model, and also systems that use PT algorithms in the melodic identification stage, which lead to gaps in time where no suitable trajectories are formed (Cancela P. , 2008; Dressler, 2006; Li & Wang, 2005; Paiva, Mendes, & Cardoso, 2006). Some melody extractions systems do not attempt to make a voicing detection (Cao, Li, Liu, & Yan, 2007; Goto, 2004).

As an independent process, SVD is required for several Music Information Retrieval (MIR) applications such as artist identification (Berenzweig, Ellis, & Lawrence, 2002), voice separation (Li & Wang, 2007) and lyrics alignment (Fujihara & Goto, 2008). The last decade

has witnessed a significant increase in research interest in the SVD problem. Figure 6.1 shows a block diagram of a typical SVD system. SVD is typically viewed as an audio classification problem where features that distinguish vocal segments from purely instrumental segments in music are first concatenated into a feature vector, and then are fed to a machine-learning algorithm/classifier previously trained on manually labeled data. The labels output by the classifier, for each feature vector, may then be post-processed to obtain smooth segmental label transitions.


Figure 6.1: Block diagram of a typical singing voice detection system

While a number of sophisticated machine-learning methods are available, it is well known that “the methods of statistical pattern recognition can only realize their full power when real-world data are first distilled to their most relevant and essential form.” (Berenzweig, Ellis, & Lawrence, 2002). This emphasizes the importance of the design and selection of features that, for the task of interest here, demonstrate the ability to distinguish between singing voice, in the presence of accompanying instruments, and the instrumentation alone.

The main focus of this chapter is to identify and design features that show high discriminative power in the SVD context. In this chapter we investigate the design of SVD system modules that effectively leverage the availability of the predominant F0 contour for the extraction and effective utilization of both static and dynamic features. While most research results in MIR are reported on collections drawn from one or another culture (mostly Western), we are especially interested in features that work cross culturally. It would be expected that certain features are more discriminative on particular music collections than on others, depending on the musical content (Lidy, et al., 2010). However, a recent study on cross-globe dataset for the particular task of vocal-instrumental classification obtained encouraging results with a standard set of features on ethnographic collections suggesting that this type of classification can be achieved independently of the origin of musical material and styles (Proutskova & Casey, 2009). This suggests a deeper study of cross-cultural performance, with its inherent diversity of both singing styles and instrumentation textures, across distinct feature sets.

## 6.1 Previous Work on Singing Voice Detection

### 6.1.1 Features

Until recently, singing voice detection algorithms employed solely static features, typically comprising frame-level spectral measurements, such as combinations of mel-frequency cepstral coefficients (MFCCs) (Berenzweig, Ellis, & Lawrence, 2002; Fujihara, Kitahara, Goto, Komatani, Ogata, & Okuno, 2006; Li & Wang, 2007; Lukashevich, Grunhe, & Dittmar, 2007; Xiao, Zhou, & Zhang, 2008), warped or perceptually derived linear predictive coefficients (LPCs) (Berenzweig, Ellis, & Lawrence, 2002; Berenzweig & Ellis, 2001; Ramona, Richard, & David, 2008), log frequency power coefficients (LFPC) (Nwe & Li, 2007), harmonicity related features (Chou & Gu, 2001; Kim & Whitman, 2004; Nwe & Li, 2008) and other spectral features such as flux, centroid and roll-off (Tzanetakis G. , 2004; Zhang T. , 2003). Rocomora and Herrera (2007) compared the performance of the above sets of features in an SVD task and found that MFCCs resulted in the best classification performance.

A consideration of acoustic attributes necessary for the detection of vocal segments in music fragments by humans could be interesting, apart from its potential in guiding the search for suitable features for the task. An experiment in which subjects listened to short excerpts (less than 500 ms long) of music from across diverse genres showed that human listeners can reliably detect the presence of vocal content in such brief excerpts (Vallet & McKinney, 2007). The presence of note transients in the excerpt was found especially useful, indicating that both static features and dynamic features (changes) provide important perceptual cues to the presence of vocalization

The explicit modeling of temporal dynamics, as an important component of perceived timbre, has found a place in recent research on musical instrument recognition (Burred, Robel, & Sikora, 2010; Lagrange, Raspaud, Badeau, & Richard, 2010). Similar to the considerations in the instrument recognition application, features linked to the temporal evolution of the spectral envelope can be designed to capture specific attributes of the instrument sound. For example, in view of the well-known source-filter model of sound production applicable to human speech and to musical instruments, variations in spectral envelope with time represent variations of the filter component independent of the source component (i.e. F0). Such a dynamic feature could potentially discriminate between singing voice and (even similarly pitch-modulated expressive) musical instruments due to the absence

of formant articulation dynamics in the latter. Another distinguishing aspect could be the attack-decay envelope peculiar to a particular instrument as reflected in the variation of spectral amplitudes over note durations.

The use of dynamic features in SVD research has largely been confined to feature derivatives, representing very short-time dynamics. A few studies have explored the observed pitch instability of the voice relative to most accompanying instruments in the form of features representing longer-term dynamics of pitch and/ or the harmonics such as arising from vibrato and tremolo in singing. Shenoy, Wu and Wang (2005) exploit pitch instability in an indirect way by applying a bank of inverse comb filters to suppress the spectral content (harmonics) of stable-pitch instruments. Nwe and Li (2007)made use of a bank of band-pass filters to explicitly capture the extent of vibrato within individual harmonics upto 16 kHz. Regnier & Peeters (2009) attempted a more direct use of frequency and amplitude instability of voice harmonics (manifested as vibrato and tremolo respectively in western music). Their method is based on the observation that the extents of vibrato and tremolo in singing are different than those for most instruments.

More recently, the full modulation spectrogram of an audio segment was used with a view to capturing the characteristics of time-varying quantities such as pitch and syllabic rates for singing voice detection (Markaki, Holzapfel, & Stylianou, 2008). The high dimensional modulation spectrogram was reduced by principal component analysis and used in a support vector machine (SVM) classifier for singing voice detection. Although inferior to MFCCs, their addition to MFCCs improved vocal detection accuracies slightly. Clearly, there is scope for improvement in terms of capturing the essential differences in the dynamics of the singing voice and musical instruments in a compact and effective way.

### 6.1.2 Classifiers

Statistical classification methods are very effective in exploiting the overall information provided about the underlying class by the set of diverse features if suitable data is provided for the training of the statistical models. Previous studies on SVD have employed a variety of classifiers including Gaussian mixture models (GMM) (Chou & Gu, 2001; Fujihara, Kitahara, Goto, Komatani, Ogata, & Okuno, 2006; Li & Wang, 2007; Lukashevich, Grunhe, & Dittmar, 2007; Vallet & McKinney, 2007), support vector machines (SVM) (Maddage, Xu, & Wang, 2003; Ramona, Richard, & David, 2008), multi-layer perceptrons (MLP) (Berenzweig, Ellis,

& Lawrence, 2002) and hidden Markov models (HMM) (Berenzweig & Ellis, 2001; Nwe & Li, 2007).

### 6.1.3 Post-processing

Frame-level decisions of a statistical classifier for SVD are known to be noisy due to local variability in the underlying real-signal. Some SVD systems incorporate a post-processing stage, which smoothes frame-level classification labels between known or detected boundaries (Li & Wang, 2007). This smoothening is done by either combining frame-level classifier likelihoods over segments or alternatively applying a majority vote. Other systems do not require segmental boundaries before hand and attempt to smooth the decision function (log likelihood ratio in case of a GMM) to achieve smoother results (Lukashevich, Grunhe, & Dittmar, 2007; Xiao, Zhou, & Zhang, 2008). The use of a post-processing stage typically improves the frame-level accuracy significantly.

## 6.2 Feature Extraction

In this section we first describe a pre-processing stage that enables the isolation of the predominant source harmonic structure from a polyphonic spectrum. We then describe different static and dynamic features considered for use in the SVD problem. Static features are computed locally over short sliding analysis frames while dynamic features are computed over larger non-overlapping time windows called texture windows. To avoid confusion, we will use the term frame to refer to the analysis frame and window for the texture windows. Finally we describe some methods of feature selection that are used to reduce the dimensionality of feature sets without reducing the voice/instrument discriminative information content.

### 6.2.1 Harmonic Sinusoidal Model (HSM)

As mentioned before, features for discriminatory tasks such as instrument classification and singing voice detection in polyphonic music have been extracted from unprocessed spectra of the polyphonic audio signal. However, prior knowledge of the underlying predominant-F0 enables the isolation of the dominant source-specific harmonic components by using a harmonic sinusoidal model (HSM), which is expected to result in more compact and relevant features.

#### *6.2.1.1 Frame-level Harmonic Detection*

Very recently Fujihara et. al. (Fujihara, Goto, Kitahara, & Okuno, 2010) have described an approach to identify local source-specific harmonics in polyphony with prior knowledge of the predominant-F0. In their approach spectral local maxima in the vicinity of expected harmonic locations are labeled as detected harmonics. Instead of using spectral local maxima we utilize the sinusoidal representation of Section 3.2. We search the sinusoidal space in a 50-cent neighborhood of expected harmonic locations to identify local harmonic components. A hard upper limit of 50 Hz is applied to the neighborhood value so as to avoid the labeling of far-away sinusoids at higher frequencies as harmonic components. This results in a harmonic line spectrum for each frame. The use of sinusoids as opposed to local maxima for harmonic identification should reduce the chances of corrupting the isolated dominant source spectrum with side-lobes and missing/distorted harmonics.

#### *6.2.1.2 Harmonic Partial Tracking*

The predominant-F0 trajectory is expected to be temporally smooth. Consequently we apply a partial tracking (PT) stage to the harmonic line spectra over time. PT involves linking detected sinusoids across time to form long-term sinusoidal tracks. This track formation is expected to further help in source spectral isolation by removing redundant spectral content during large pitch jumps during transients and note changes. We adopt an approach similar to Serra (Serra, 1997) in which partial tracking is improved by biasing trajectory formation towards expected harmonic locations based on detected pitch. Specifically tracks are formed and ordered by their respective harmonic index.

We apply a one semitone threshold on track continuation i.e. a track will ‘die’ if there does not exist any sinusoid within 1 semitone of the last tracked frequency. Further, competition between multiple sinusoids for being linked to a given track is not resolved by a criterion based purely on frequency proximity of the sinusoids to the frequency of the last tracked sinusoid, as originally proposed, since it has been observed that often high-amplitude peaks that correspond to genuine partials are completely missed in favor of spurious, nearby low-amplitude peaks. We propose instead a cost function that takes into account sinusoidal amplitudes, in addition to the frequency, given by

$$J = \left| \left( \omega_n^k - \omega_{n-1}^m \right) \times \log \left( A_n^k / A_{n-1}^m \right) \right| \tag{6.1}$$

where $\omega_{n-1}^{m}$ and $A_{n-1}^{m}$ are the frequency and amplitude resp. of the $m^{\text{th}}$ harmonic track in frame $n$-1, $\omega_{n}^{k}$ and $A_{n}^{k}$ are the frequency and amplitude resp. of the $k^{\text{th}}$ local sinusoid in the frame $n$ that is competing for joining the track.

The HSM representation now consists of a set of harmonic trajectories over time, which themselves can be analyzed for dynamic feature extraction. Rather than extract features from the discrete frame-level frequency spectra obtained after HSM, we use smooth spectra derived by the log-linear interpolation of the harmonic spectral amplitudes as follows. Given a set of estimated amplitudes $S(\omega_1)$, $S(\omega_2)$,…, $S(\omega_L)$ at $L$ uniformly spaced harmonic frequencies $\omega_1$, $\omega_1$,…, $\omega_L$ we generate spectral amplitudes $Q(\theta)$ at fixed DFT bin spacing as shown below:

$$Q\left(\theta_j\right) = 10^{\log\left|S\left(\omega_k\right)\right| + \left(\frac{\theta_j - \omega_k}{\omega_{k+1} - \omega_k}\right)\left(\log\left|S\left(\omega_{k+1}\right)\right| - \log\left|S\left(\omega_k\right)\right|\right)} \tag{6.2}$$

where $\omega_k << \omega_{k+1}$. The interpolation serves to make any spectral envelope features extracted subsequently less dependant on the F0 and thus more representative of source timbre.

## 6.2.2 Static Features

### *6.2.2.1 Timbral Features*

In previous work on SVD the most commonly used features are static timbral descriptors, which attempt to capture the timbral aspects of musical sounds and do not utilize any specific long-term traits of the singing voice or instruments. Rocomora and Herrera (2007) compared the performance of a comprehensive list of static features in an SVD task and found that MFCCs (specifically the first 13 coefficients using 40 mel scale frequency bands) resulted in the best classification accuracies. Consequently MFCC coefficients constitute one of the feature sets that we experiment with for the SVD problem. We use the MFCC implementation of Slaney (1998) in which the mel filter bank is constructed using a set of 40 triangular filters: 13 linearly-spaced filters (133.33Hz between center frequencies,) followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency.)

Apart from these we also extract several other static timbral features. These features were chosen out of a larger set of features, which have all been defined by Peeters (2004) but, in some cases, have been used with modifications. For all feature computation formulae, $X_n(k)$ is used to denote the magnitude spectral value of the $k^{\text{th}}$ spectral bin for the $n^{\text{th}}$ frame. When

used $k_f$ denotes the bin number closest to frequency $f$ (in Hz), $f(k)$ is the frequency of the $k^{th}$ spectral bin center and $N$ is the number of FFT points/bins. In some cases the features are computed over a sub-band $b$ whose upper and lower frequency limits are denoted by *fh(b)* and *fl(b)* respectively.

*Sub-band Spectral Flatness (SF)*

*SF* of a spectral band $b$ is defined as the ratio of the geometric and the arithmetic means of the power spectrum coefficients within that band. It is given by

$$\mathrm{SF}_n = \frac{\sqrt[(fh(b)-fl(b)+1)]{\prod_{k=k_{fl(b)}}^{k_{fh(b)}} X_n^2(k)}}{\sum_{k=k_{fl(b)}}^{k_{fh(b)}} X_n^2(k) \Big/ (fh(b)-fl(b)+1)} \tag{6.3}$$

It is a measure of deviation of the spectral form from that of a flat spectrum. Flat spectra correspond to noise or impulse-like signals. Thus high flatness indicates noisiness. Low flatness values generally indicate the presence of harmonic components. The band chosen was 1.5 kHz – 3 kHz. Voice has a harmonic structure until 5 kHz and hence gives a high value for this feature. Instruments generally display a much sharper roll-off and are expected to have low values in this band.

*Spectral roll-off* (*SRO*)

It is the frequency below which $P$ % of the signal energy is concentrated. This feature is defined as that frequency $F_r$ for which

$$\sum_{k=1}^{k_{Fr}} X_n^2(k) = \frac{P}{100} \times \sum_{k=1}^{N/2} X_n^2(k) \tag{6.4}$$

$k_F$ is the spectral bin number whose center is nearest to $F$ Hz. A value of $P$=70 was used here.

*Spectral centroid* (*SC*)

It is a perceptually adapted definition of the centroid, which introduces a logarithmic frequency scaling, centered at 1 kHz. It is given by

$$\mathrm{SC}_n = \frac{\sum_{k=1}^{N/2} \log_2(f(k)/1000) X_n(k)}{\sum_{k=1}^{N/2} X_n(k)} \tag{6.5}$$

Spectral centroid of the voice is known to vary over a region of 1 kHz to 2.5 kHz.

*Sub-band Flux (Z)*

Spectral Flux is a measure of frame-to-frame spectral change and is defined as the squared difference between the normalized magnitudes of successive spectral distributions (Tzanetakis & Cook, 2002). We compute it over a given frequency band *b* as

$$Z_n = \sum_{k=k_{fl(b)}}^{k_{fh(b)}} \left( X_n(k) - X_{n-1}(k) \right)^2 \tag{6.6}$$

Here *fl(b)* and *fh(b)* are 1.5 and 5 kHz respectively. It is expected that this feature will generally have a high value during the sung segments, whose spectra show significant inter-frame variation as compared to the more stationary purely instrumental segments.

*Spectral spread (SPS)*

It is defined as the spread of the spectrum around its mean value (i.e. the variance of the above-defined distribution). It is a statistical measure, which describes how the spectrum is concentrated around the centroid. It is given by

$$\mathrm{SS}_n = \sqrt{\frac{\sum_{k=1}^{N/2} \left( \log_2(f(k)/1000) - \mathrm{SC}_n \right)^2 \cdot X_n(k)}{\sum_{k=1}^{N/2} X_n(k)}} \tag{6.7}$$

Low values of the spread indicate that the spectrum is highly concentrated near the centroid; high values mean that it is distributed across a wider range at both sides of the centroid.

*Sub-band Energy (SE)*

It is the average energy of the spectral sub-band (*b*). It is given by

$$\mathrm{SE}_n = \frac{1}{N_b} \sum_{k=k_{fl(b)}}^{k_{fh(b)}} X_n^2(k) \tag{6.8}$$

Here *fl(b)* and *fh(b)* are 1.5 and 5 kHz respectively.

*Sub-band Energy Ratio (SER)*

It is the ratio of the energies of two frequency bands (*b1* and *b2*) in spectrum It is given by

$$\mathrm{SER}_n = \frac{\sum_{k=k_{fl(b1)}}^{k_{fh(b1)}} X_n^2(k)}{\sum_{k=k_{fl(b2)}}^{k_{fh(b2)}} X_n^2(k)} \tag{6.9}$$

Here *fl(b1)* and *fh(b1)* are 5 and 8 kHz resp. and *fl(b2)* and *fh(b2)* are 0 and 1.5 kHz resp. Lower values of this feature are expected during frames where the voice is present since the concentration of harmonic energy for the voice is below 5 kHz.

#### *6.2.2.2 F0-Harmonic Features*

The previously described static timbral features do not use predominant-F0 information. Here we define some static features that take advantage of the access to predominant-F0 information provided by the previous modules.

First we define a feature, called normalized harmonic energy (*NHE*), based on detecting the singing voice by the energy of the predominant pitch source i.e., since the voice, when present, is expected to be dominant in the song, the energy of the predominant source can be used as a cue for detecting the presence of the singing voice. First, the harmonic energy (*HE*), defined as the sum of the energies of individual harmonics (multiples of the pitch) in the frequency region up to 5 kHz, is computed as

$$\mathrm{HE} = \sum_{i=1}^{N} X_n^2(f_i) \tag{6.10}$$

where $X_n(f_i)$ is the spectral magnitude, at time instant $n$, at the frequency bin corresponding to the closest detected sinusoid, with frequency $f_i$, within a 50 cent neighborhood of the expected location of the $i^{\text{th}}$ harmonic. $N$ is the total number of expected harmonics below 5 kHz for the estimated pitch. A value of 5 kHz is used since significant voice harmonics are rarely found above this limit. The *HE* is normalized by its maximum attained value over a single musical performance to give the *NHE*.

Recently Fujihara & Goto (2008) proposed that the feature set of F0 and harmonic powers were found to be highly discriminatory for SVD when source spectral isolation was possible because of the availability of the predominant source F0 values. The harmonic powers were normalized for each song. The normalized power of the $h^{\text{th}}$ component at time $t$ is given by

$$p_h^{'t} = \log p_h^t - \frac{\sum_t \sum_h \log p_h^t}{T \times H} \tag{6.11}$$

where $p_h^t$ represents the original power, $T$ is the total number of frames in the song, and $H$ is the number of harmonic components considered per frame.

## 6.2.3 Dynamic Features

Dynamic features are divided into two categories – timbral dynamics and F0-harmonic dynamics, since these categories are expected to complement each other for contrasting signal conditions.

### *6.2.3.1 Timbral Features*

One of the problems with the effective use of timbral dynamics for instrument classification in polyphony was found to be the lack of an ability to pay selective attention to isolated instrumental spectra (Aucouturier & Patchet, 2007). We expect that the access to the harmonic sinusoidal model (HSM) representation for extracting these features should overcome this particular problem. Here we describe the extraction of the dynamics of two static timbral features – spectral centroid (SC) and sub-band energy (SE). These are defined differently from Section 6.2.2.1. These dynamic features were found to capture transitions in the spectra of sung utterances.

*Spectral Centroid* (*SC*)

The first of these is the sub-band spectral centroid (*SC*) and is computed as shown below

$$\mathrm{SC}_n = \frac{\sum_{k=k_{low}}^{k_{high}} f(k) X_n(k)}{\sum_{k=k_{low}}^{k_{high}} X_n(k)}. \tag{6.12}$$

where $f(k)$ and $X_n(k)$ are frequency and magnitude spectral value of the $k^{\text{th}}$ frequency bin at time instant $n$, and $k_{\text{low}}$ and $k_{high}$ are the nearest frequency bins to the lower and upper frequency limits on the sub-band respectively. The specific sub-band chosen [1.2–4.5 kHz] in this case is expected to enhance the variations in the 2nd, 3rd and 4th formants across phone transitions in the singing voice. This feature is expected to remain relatively invariant over

note transitions in the instrument. Although the band-limits are restrictive, even very high pitched instruments will have at least 3-4 harmonics present within this band so that their spectral envelope can be relatively well represented.

*Sub-band Energy (SE)*

The second feature for dynamic analysis is the sub-band energy (*SE*) and is computed using the same formula as in Equation 6.8. The band limits for *SE*, in this case, are chosen as [300-900 Hz]. This is expected to enhance the fluctuations between voiced and unvoiced utterances while remaining relatively invariant to small instrumental note transitions. Fluctuations in this feature should be evident even if the signal representation captures some pitched accompanying instrument information during unvoiced sung sounds.

To capture meaningful temporal variations in the dynamics of the above timbral features, it is necessary to choose the duration of the observation interval appropriately (Lagrange, Raspaud, Badeau, & Richard, 2010). We choose three different time scales for our feature set: 0.5 sec (long note duration), 1 and 2 sec intervals (to capture note articulation changes in both fast and slow singing). We represent the dynamics via the standard deviation (std. dev.) and specific modulation energies over the different observation intervals. These modulation energies are represented by a modulation energy ratio (*MER*). The *MER* is extracted by computing the DFT of the feature trajectory over a texture window and then computing the ratio of the energy in the 1-6 Hz region in this modulation spectrum to that in the 1-20 Hz region as shown below:

$$\mathrm{MER} = \frac{\sum_{k=k_{1\mathrm{Hz}}}^{k_{6\mathrm{Hz}}} |Z(k)|^2}{\sum_{k=k_{1\mathrm{Hz}}}^{k_{20\mathrm{Hz}}} |Z(k)|^2} \tag{6.13}$$

where $Z(k)$ is the DFT of the mean-subtracted feature trajectory $z(n)$ and $k_{fHz}$ is the frequency bin closest to $f$ Hz. We assume that the fastest syllabic rate possible, if we link each uttered phone to a different note in normal singing, should not exceed 6 Hz. Steady note durations are not expected to cross 2 seconds. The std. dev. and *MER* of the above features are expected to be higher for singing than instrumentation.

#### *6.2.3.2 F0-HarmonicFeatures*

Singing differs from several musical instruments in its expressivity, which is physically manifested as the instability of its pitch contour. In western singing, especially operatic singing, voice pitch instability is marked by the widespread use of vibrato.. Within non-western forms of music, such as Greek Rembetiko and Indian classical music, voice pitch in ections and ornamentation are extensively used as they serve important aesthetic and musicological functions. On the other hand, the pitch contours of several accompanying musical instruments, especially keyed instruments, are usually very stable and incapable of producing pitch modulation.

Dynamic F0-Harmonic features are expected to capture differences in the shape of F0/harmonic trajectories between the singing voice and other musical instruments. These differences are emphasized when the singing voice is replete with pitch modulations and the accompanying instruments are mostly stable-note (keyed) instruments. Here we would not like to restrict ourselves to targeting particular types of pitch modulation such as vibrato but extract some statistical descriptors (mean, median, std. dev.) of general pitch instability-based features over texture windows of expected minimum note duration (here 200 ms). These features are the first-order differences of the predominant-F0 contour and the subsequently formed harmonic frequency tracks. The track frequencies are first normalized by harmonic index and then converted to the logarithmic cents scale so as to maintain the same range of variation across harmonics and singers' pitch ranges. For the latter we group the tracks by harmonic index (harmonics 1-5, harmonics 6-10, harmonics 1-10) and also by low and high frequency bands ([0–2 kHz] and [2–5 kHz]). This separation of lower and higher harmonics/frequency bands stems from the observation that when the voice-pitch is quite stable the lower harmonics do not display much instability but this is clearly visible in the higher harmonics. However when the voice-pitch exhibits large modulations the instability in the lower harmonic tracks is much more clearly observed but often the higher harmonic tracks are distorted and broken because of the inability of the sinusoidal model to reliably track their proportionately larger fluctuations. We also compute the ratio of the statistics of the lower harmonic tracks to those of the higher harmonic tracks since we expect these to be much lesser than 1 for the voice but nearly equal to 1 for flat-note instruments.

### 6.2.4 Feature Selection

Feature selection is the process of identifying a small number of highly predictive features and removing as much redundant information as possible in order to avoid over fitting the training data. Reducing the dimensionality of the data reduces the size of the hypothesis space and allows machine learning algorithms to operate faster and more effectively. Feature selection involves the generation of a ranked list of features based on some criterion using some labeled training data and the subsequent selection of the top-N ranked features.

One such criterion is provided in (Hall, Frank, Holmes, Pfarhringer, Reutemann, & Witten, 2009), which evaluates a feature by measuring the information gain ratio of the feature with respect to a class, given by

$$\mathrm{GainR}(C,F) = \left( \frac{H(C) - H(C \mid F)}{H(F)} \right) \tag{6.14}$$

where $H$ is the information entropy, $C$ is the class label and $F$ is the feature. The amount by which the entropy of the class decreases reflects the additional information about the class provided by the feature. Each feature is assigned a score based on the information gain ratio which generates a ranked feature list. Another feature selection criterion, the mutual information (MI), has been used to evaluate the "information content" of each of the individual feature with regard to the output class (Battiti, 1994). Higher values of mutual information for a feature are indicative of better discriminative capability between classes.

## 6.3 Classifier

In the present work, we use a standard GMM classifier (Boumann) for the evaluation of our features. The basic classification task assumes that each vector belongs to a class and each class is represented by a Probability Distribution Function (PDF), which can be modeled as a mixture of multi-dimensional Gaussians (Arias, Pinquier, & Andre-Obrecht, 2005). There are two phases, which are present in operation of GMM classifiers.

1. During the training phase the PDF parameters of each class are estimated
2. During the classification phase, a decision is taken for each test observation by computing the maximum-likelihood criterion.

From the training data, parameters required for modeling the GMM are first estimated. For Gaussian models the parameters required are the means, variances and weights of each of the GMM components belonging to each class. Since each model has a number of

components, weights are assigned to these components and the final model is built for each class. The Expectation Maximization (EM) algorithm is used for finding the means, variances and weights of the Gaussian components of each class. Fuzzy k-means algorithm is used for initializing the parameters of the classifier. The algorithm is implemented iteratively until the log likelihood of the training data with respect to the model is maximized. While testing, an unknown feature vector is provided to the GMM classifier. The final likelihood for each class is obtained by the weighted sum of likelihoods of each of the individual components belonging to the class. The output class label is that class which provides the maximum value of likelihood.

There are two major advantages of using GMM. The first is the intuitive notion that the individual component densities of a multi-modal density may model some underlying set of acoustic characteristics. The second advantage of using Gaussian mixture densities is that the empirical observation that a linear combination of Gaussian basis functions is capable of representing a large class of sample distributions. One of the powerful attributes of GMM is its ability to form smooth approximations to arbitrarily shaped densities (Arias, Pinquier, & Andre-Obrecht, 2005) GMM not only provides a smooth overall distribution but its components also clearly detail the multi-modal nature of the distribution.

While the straightforward concatenation of features is a common way to integrate the overall information content represented by the individual features or feature sets, a combination of individual classifiers can improve the effectiveness of the full system while offsetting difficulties arising from high dimensionality (Kittler, Hatef, Duin, & Matas, 1998). Combining the likelihood scores of classifiers is particularly beneficial if the corresponding individual feature sets represent complementary information about the underlying signal. Weighted linear combination of likelihoods provides a flexible method of combining multiple classifiers with the provision of varying the weights to optimize performance. The final class likelihood $S$ for vocal class $V$ is given by

$$S(V) = \sum_{n=1}^{N} w_n \left( \frac{p_n\left(\overline{f}_n \mid V\right)}{p_n\left(\overline{f}_n \mid V\right) + p_n\left(\overline{f}_n \mid I\right)} \right) \tag{6.15}$$

where $N$ is the number of classifiers, $\overline{f}_n$ and $w_n$ are the feature vector and weights for the $n^{\text{th}}$ classifier respectively, $p(f|C)$ is the conditional probability of observing $f$ given class $C$.

## 6.4 Boundary Detection for Post-Processing

For post-processing we use the framework described by Li & Wang (2007) in which the frame-level decision labels are combined over automatically detected homogenous segments. In this section we first describe and show the problems with using the spectral change detection function used by Li & Wang (2007) for segment boundary detection in the SVD context and then describe and evaluate a novelty-based frame-work for boundary detection.

### 6.4.1 Spectral Change Detector

Boundary location in music by identifying instances of significant spectral change is predicated by the knowledge that most musical notes will initially have a short unsteady part with a sudden energy increase which will be followed by a longer steady state region. The spectral change function is then expected to have high values during note onset and low values for the steady state. In their implementation of post-processing for SVD Li & Wang (2007) use a system proposed by Duxbury, Bello, Davies & Sandler (2003). Their system was shown to have well formed peaks at note onset locations for polyphonic music (although the nature of the instruments was not stated).

#### *6.4.1.1 Implementation*

This spectral change function (SCF) is computed using the following steps:

Step 1: Compute Euclidian distance $\eta(m)$ between the expected complex spectral value and the observed one in a frame

$$\eta(m) = \sum_k \left( \left| \hat{S}_k(m) - S_k(m) \right| \right) \tag{6.16}$$

where $S_k(m)$ is the observed spectral value at frame $m$ and frequency bin $k$. $\hat{S}_k(m)$ is the expected spectral value of the frame and the same bin, calculated by

$$\hat{S}_k(m) = \left| S_k(m-1) \right| e^{j\hat{\phi}_k(m)} \tag{6.17}$$

where $|S_k(m-1)|$ is the spectral magnitude of the previous frame at bin $k$. $\hat{\phi}_k(m)$ is the expected phase which can be calculated as the sum of the phases of the previous frame and the phase difference between the previous two frames.

$$\hat{\phi}_k(m) = \tilde{\varphi}_k(m-1) + \left( \tilde{\varphi}_k(m-1) - \tilde{\varphi}_k(m-2) \right) \tag{6.18}$$

where $\tilde{\varphi}_k(m-1)$ and $\tilde{\varphi}_k(m-2)$ are the unwrapped phases for frame *m-1* and frame *m-2* resp.

Local peaks in $\eta(m)$ indicates a spectral change.

Step 2: To accommodate the dynamic range of spectral change as well as spectral fluctuations, weighted dynamic thresholding is applied to identify the instances of significant spectral changes. Specifically, a frame *m* will be recognized as an instance of spectral change if $\eta(m)$ is a local peak and $\eta(m)$ is greater than the weighted median value in a window of size *H* i.e.

$$\eta(m) > C \times \text{ median}\left(\eta\left(m-\frac{H}{2}\right),...,\eta\left(m+\frac{H}{2}\right)\right) \tag{6.19}$$

where *C* is the weighting factor.

Step 3: Finally, two instances are merged if the enclosed interval is less than $T_{min}$; specifically if two significant spectral changes occur within $T_{min}$, only the one with the larger spectral change value $\eta(m)$ is retained. The values of *H*, *C* and $T_{min}$ are 10, 1.5 and 100 ms respectively.

#### *6.4.1.2 Evaluation*

The spectral change detector was implemented using the parameter settings as recommended in (Li & Wang, 2007). We only used spectral content upto 5 kHz, an acceptable upper limit for significant voice harmonics. We first tested it on a monophonic sung voice segment, which had three notes of same pitch and utterance /aa/ and plotted its SCF. (See top & bottom left of Figure 6.2). Rather than displaying peaks at the sung note onsets and offsets the SCF goes high at the vowel onset and stays high till the offset. This is contrary to expectation since we are expected to detect peaks in the SCF at note onsets. This can be attributed to voice instability and the fact that the energy of the sung note does not decay as rapidly as an instrument note. To emphasize this further, we computed the SCF for a sequence of synthetic tabla strokes, whose rate of decay is more rapid than that of the voice (See right of Figure 6.2). Here the peaks in the SCF will clearly indicate the onsets of strokes.

As we are interested in the phrase onsets and offsets of sung vocal segments, the peaks from the SCF itself may not be reliable locations of boundaries. In fact in (Li & Wang, 2007) it is stated that this kind of a detector is useful assuming that the voice "more likely joins the accompaniment at beat times in order to conform to the rhythmic structure of the song." So in music where it is not necessary that the voice onsets and a beat location may coincide, using

the peaks from the SCF may not be reliable. Since the SCF is dominated by strong percussive interference, broad changes in the feature from vocal to non-vocal segments may not be visible due to the numerous peaks caused by individual beats. To illustrate this consider Figure 6.3.a. Here there are two sung phrases, whose onsets and offsets are marked by dotted lines. But the SCF shows numerous peaks during the phrases corresponding to percussive stroke onsets. This causes the distinction between vocal and non-vocal segments to become unclear.

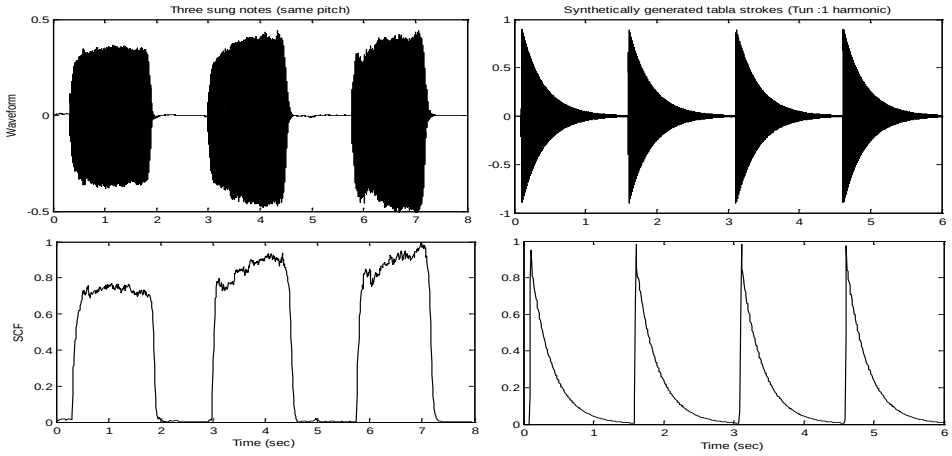


Figure 6.2: Waveforms (above) and SCFs (below) for a three note natural sung signal (left) and a four stroke synthetic *tabla* signal (right)

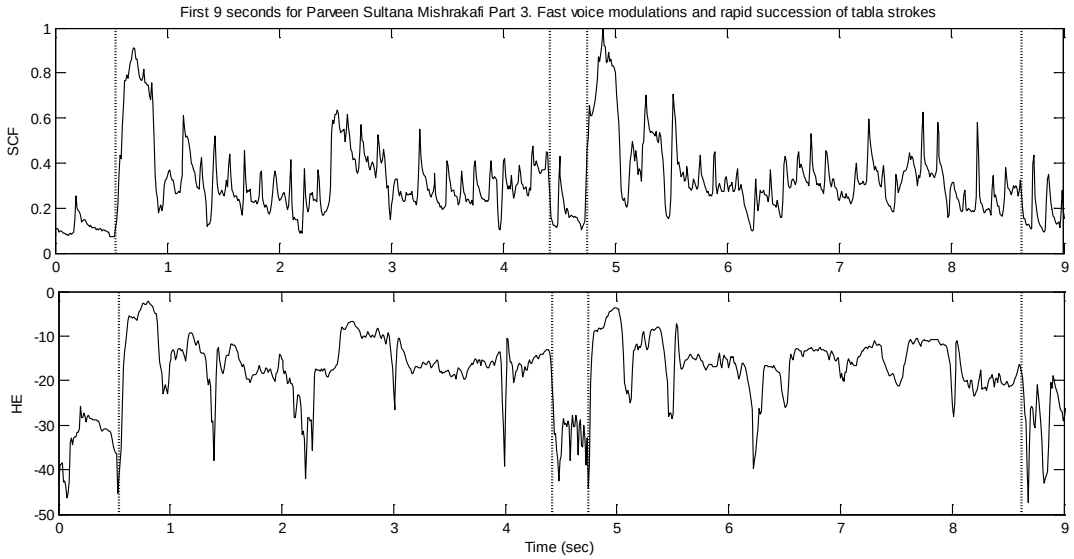


Figure 6.3: (a) SCF and (b) NHE plots for the last 9 seconds of a Hindustani classical vocal performance with fast voice pitch variations and rapid sequence of *tabla* strokes. Dotted lines indicate sung phrase onsets/offsets.

### 6.4.2 Audio Novelty-based Boundary Detection Framework

The boundary detection framework proposed by Foote (2000) examines the audio signal for detecting points of 'novelty'. A novel point in the audio signal is a point of significant change that will have high self-similarity in the past and future and low cross-similarity. The output of the system is a novelty function. In the context of SVD an ideal novelty function should have strong peaks at actual sung-phrase boundary locations and be zero elsewhere. The inputs to the novelty function generator will typically be some feature(s), which show sharp changes at actual boundary locations. A description of the implementation follows.

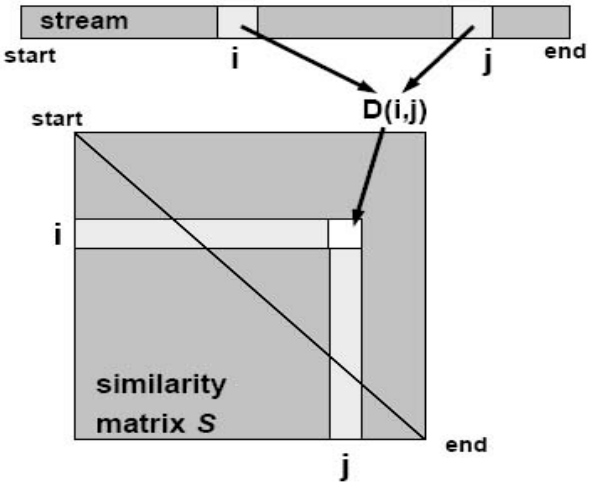


Figure 6.4: Similarity matrix construction from (Foote, 2000). *Used with permission*.

Step 1: Here initially a distance measure *D* is computed between the feature vectors of all possible pairs of frames. Examples of distance measures are Euclidian or cosine distances. Then the distance measure is embedded in a 2-dimensional representation called a similarity matrix *S*. The *i*, *j*th element of *S* is *D*(*i*, *j*) (see Figure 6.4). In general, *S* will have maximum values on the diagonal because every window will be maximally similar to itself.

Step 2: *S* can be visualized as a square image where each pixel *i*, *j* has a gray scale value proportional to the similarity measure *D*(*i*, *j*). For example, for a two note cuckoo call, where pitch is the feature, *S* will look like a 2 x 2 checkerboard. White squares on the diagonal correspond to the notes, which have high self similarity. Black squares on the off diagonal correspond to regions of low cross similarity. Finding the instant of change is done by correlating *S* with a kernel that itself looks like a checkerboard. The simplest kernel is a 2x2 unit kernel (*C*) given by

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6.20}$$

Larger kernels can be computed by forming the Kronecker product of $C$ with matrix of ones. For example to form a 4x4 kernel ($D$),

$$D = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \tag{6.21}$$

In order to avoid edge effects and for better time localization, we smooth these large kernels using a radially symmetric Gaussian function to obtain a Gaussian difference kernel (GDK).

Step 3: The novelty function is generated by continuously shifting the GDK along the diagonal and correlating with the region of $S$ covered by it. When $D$ is positioned exactly at a crux of the checkerboard, the positive regions will multiply with the positive regions of high self similarity and the negative regions will multiply with the negative regions of low cross-similarity, and the overall sum will be large. When $D$ is over a relatively uniform region the positive and negative regions will tend to sum to zero. Peaks in the novelty function will then indicate instants of change.

### 6.4.3 Evaluation of the Boundary Detection System

Here we evaluate the performance of the novelty-based boundary detection algorithm for Indian classical music. The criteria for evaluation are the number of true hits and the number of false alarms. A boundary is said to be correctly found when it is within 150 ms of the manually marked boundary. If multiple boundaries are found within 150 ms of a manually marked boundary only the closest boundary is considered to be a true hit. The data we have used is a subset of the SVD-Hind dataset spanning 23 minutes in total. Three of the recordings are of female artists and four are of male artists. The statistics of the vocal and instrumental segments across the entire data set is shown in Table 6.1. As we can see, the vocal segments comprise nearly 75% of the data. The total number of true boundaries presents in the data, excluding start and ends of excerpts, is 374.

Table 6.1: Statistics of the vocal and non-vocal segments for the audio data set

| Segments | Number | Longest | Shortest | Avg. duration | Total duration |
|---|---|---|---|---|---|
| Vocal | 187 | 15.13 s | 0.53 s | 5.44 s | 1018.01 s |
| Non-vocal | 208 | 13.20 s | 0.11 s | 1.73 s | 359.62 s |

#### *6.4.3.1 Experimental Setup*

The input to the boundary detection framework of the previous section is the NHE feature, as described in Section 6.2.2.2. In Figure 6.3.b, we have plotted the NHE values over time for an excerpt of a female Hindustani vocal performance. We can see that the NHE values are seen to be high for voiced regions and low for instrumental regions, which indicates that it is suitable for use in the present boundary detection framework. The predominant-F0 used for computing the NHE value is computed for each clip using a system described in Section 7.2.1 (MIREX submission).

For use the novelty function generator, we require the NHE to vary between positive and negative. Since, the NHE is normalized over individual performances its values are all negative with the maximum being 0 dB. Scaling the NHE values by the mean may not be appropriate since a large percentage of values will be high because of the presence of more vocal to instrumental frames for normal songs, leading to a high mean value. We choose to scale the NHE values positively by the mid-point of the range of NHE values before sending it to the boundary detector.

We had conducted preliminary experiments with a variety of GDK lengths and novelty thresholds and found that GDK lengths of 400 or 500 ms and a novelty threshold of 0.15 resulted in optimum performance. We also added an additional boundary pruning stage i.e. if two boundaries are within X ms, then the stronger boundary (in terms of novelty function value) will be retained. The two duration thresholds we experiment with are 100 and 150 ms.

#### *6.4.3.2 Results*

The true hits and false alarms values for the novelty-based boundary detection algorithm for GDK lengths of 400 and 500 ms and pruning thresholds of 100 and 150 ms for a novelty threshold of 0.15 are shown in Table 6.2. We can see that for the length of 500 ms, while we have a marginally reduced accuracy, the number of false alarms is much lower than when length of 400 ms is used. Further the pruning stage is found to significantly reduce the number

of false alarms while only marginally reducing the number of correctly detected boundary locations. Figure 6.5 shows a histogram of the offsets from actual boundaries, obtained from annotated data, occurring during boundary marking for the same four cases as Table 6.2. We can see that for the 400 ms GDK length the largest offset in the location of a detected boundary from its actual location does not exceed 300 ms. Since, the application for the boundary detector is the grouping of frame level classification results over homogenous segments, we can tolerate a higher number of false alarms but do not want to compromise on detection accuracy. So for the SVD experiments of Section 7.1.2 we use a GDK length of 400 ms and a duration pruning threshold of 100 ms.

Table 6.2: True hits vs. False alarms for GDK of lengths 400 and 500 ms, prune thresholds of 100 and 150 ms. Fixed Novelty threshold of 0.15.

| | | | **Before Pruning** | **After Pruning** |
|---|---|---|---|---|
| Length 400 ms | Threshold 100 ms | True hits | 350 (93.6 %) | 347 (92.8 %) |
| | | False alarms | 1489 | 1198 |
| | Threshold 150 ms | True hits | 350 (93.6 %) | 343 (91.7 %) |
| | | False alarms | 1489 | 1069 |
| Length 500 ms | Threshold 100 ms | True hits | 345 (92.3 %) | 342 (91.4 %) |
| | | False alarms | 1219 | 960 |
| | Threshold 150 ms | True hits | 345 (92.3 %) | 338 (90.4 %) |
| | | False alarms | 1219 | 885 |

## 6.5 Summary

In this chapter we have presented different aspects of the singing voice detection (SVD) problem. We reviewed previous work in SVD in terms of the features, classifiers and post-processing used in a standard machine-learning train-test framework. We then described several features both static and dynamic for use in the SVD system. Both of these feature groups are further sub-categorized as timbral and F0-harmonic. The dynamic features proposed in this chapter have not been used in the SVD context before. Next we described & evaluated different boundary detection algorithms on Indian classical music data. It was found that the NHE feature input to an audio novelty detector using a similarity matrix resulted in the reliable sung-phrase boundaries.

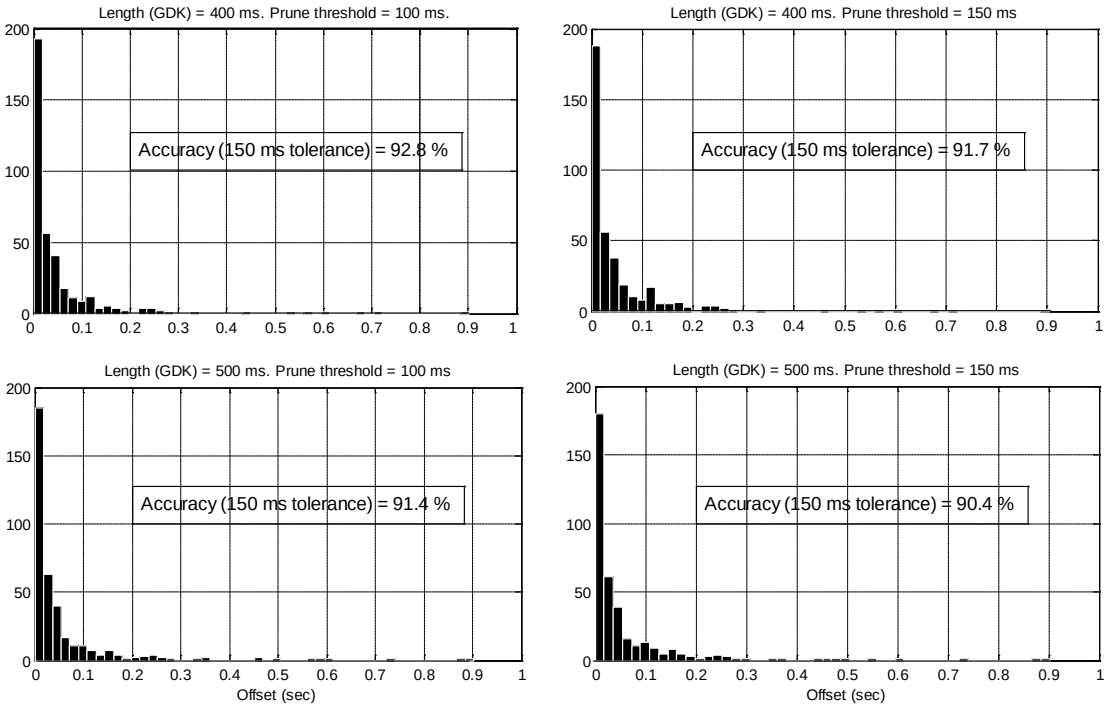


Figure 6.5: Histogram offsets for GDK of lengths 400 and 500 ms and for prune thresholds of 100 and 150 ms. Novelty threshold is fixed at 0.15.

# Chapter 7

## Melody Extraction System Evaluations – Part I

In this chapter we first report some internal evaluations of the predominant-F0 extraction system and the singing voice detection (SVD) system. The predominant-F0 extraction system considered here is the combination of main-lobe matching based fixed-resolution sinusoidal spectral representation, two-way mismatch (TWM) salience function and dynamic programming based single-F0 trajectory extraction, as described in Chapters 3, 4 and 5 respectively. The performance of this predominant-F0 extraction system is compared to some other similar systems on common music audio data. We then present an investigation of the reduction in computation complexity by modification of the multi-F0 analysis module as described in Section 4.3. For the SVD system we compare different feature sets in a standard machine-learning classification framework for common music audio data. These evaluations led to the final design of the melody extraction system submitted to the MIREX 2008 & 2009 contests, described next. We then provide an overview of the automatic melody extraction

contest at MIREX, in terms of the common datasets, evaluation metrics and submitted algorithms, following which we discuss the results of our system at the MIREX evaluations.

## 7.1 Evaluations Leading to MIREX Submission Design

Separate experiments were carried out for the evaluation of the predominant-F0 extraction system and the singing voice detection system.

### 7.1.1 Predominant-F0 Extraction Evaluations

We have seen in Chapters 4 and 5 that a combination of the fixed-resolution sinusoidal representation, two-way mismatch PDA and dynamic programming (DP)-based post-processing shows robustness to tracking the pitch of a target signal in the presence of harmonic interference. In this section we report some evaluations of this combination on real signals. The experiments with the multi-resolution and adaptive signal representations, described in Section 3.3.4 happened after our MIREX submission, and so we have used the fixed single-resolution sinusoidal representation in the subsequent experiments, where sinusoids have been detected using the main-lobe matching method.

#### *7.1.1.1 Experiment to compare TWM & PR PDAs*

Here we review two harmonic matching PDAs that have been developed for musical F0 tracking – the PR (Brown, 1992) and TWM (Maher & Beauchamp, 1994) PDAs. While both methods are based on fitting a harmonic sequence to the measured signal spectrum, they differ on the important aspect of the error, or fitness, criterion. The data used here are the different mixes (V, VT, VTT & VTTH) from the PF0-ICM dataset and only the vocal clips from the PF0-ADC04 & PF0-MIREX05-T datasets. Durations and other information regarding the ICM and MIREX data used in this experiment are given in Table 7.1 and Table 7.2 respectively. The evaluation metric (PA) is computed for the output of the TWM algorithm (RAW) and after post-processing (DP). These are computed separately for the PR and TWM algorithms. For both PDAs, the window-length and F0 search range used are 40 ms and 100-875 Hz respectively. The implementation of the TWM PDA used here computes the TWM error at all possible trial F0s within the F0 search range with very small frequency increments (1 Hz). Note that for the PF0-ADC04 data the ground-truth pitch values are provided every 5.8 ms, so we reduce the standard deviation of the smoothness cost function in the DP algorithm correspondingly to 0.058 (For the 10 ms hop we were using a SD of 0.1).

Table 7.1: Statistics of multi-track audio data PF0-ICM from NCPA

| **Performer** | **Raag** | **Vocal / Total (sec)** | **% Vocal frames** |
|---|---|---|---|
| Female | Jai Javanti | 111.22 / 123.65 | 89.95 % |
| Male | Shankar | 101.22 / 129.52 | 80.38 % |
| | Overall | 212.44 / 249.57 | 85.04 % |

Table 7.2: Statistics of ISMIR 2004 and MIREX 2005 data

| **Dataset** | **Genre label** | **Vocal / Total (sec)** | **% vocal frames** |
|---|---|---|---|
| **ISMIR 2004** | SYNTH | 66.31 / 74.91 | 88.52 % |
| | OPERA | 62.32 / 72.20 | 86.31 % |
| | POP | 65.36 / 83.82 | 77.98 % |
| **MIREX 2005** | VOICE | 173.95 / 265.89 | 65.42 % |

The results of this evaluation for the PF0-ICM and MIREX data are given in Table 7.3 and Table 7.4 respectively. For the former, it can be seen that the TWM PDA is robust to *tabla* interference by the almost similar, and also high, values of PA for the first two rows. The PA values of the TWM PDA on the *tanpura* included signal do not show any significant degradation and are still very high. Even though the *tanpura* signal is spectrally dense, a majority of its partials escape detection during voiced frames because of their very low strength and so are not involved in the TWM error computation. The PA values for the PR PDA show a significant decrease on the addition of the *tabla* but only a marginal decrease thereafter on the addition of the *tanpura*. However, the addition of the loud harmonium reduces the PA values of both PDAs by about 30% though the PA values of the TWM-DP PDA are still higher than that of the PR-DP PDA.

Table 7.3: PA values (%) for different PDAs for ICM data

| **Content** | **PR** | | **TWM** | |
|---|---|---|---|---|
| | **RAW (%)** | **DP (%)** | **RAW (%)** | **DP (%)** |
| V | 90.81 | 98.24 | 98.34 | 99.66 |
| VT | 78.01 | 80.45 | 97.41 | 99.51 |
| VTT | 76.74 | 79.71 | 92.90 | 98.20 |
| VTTH | 47.77 | 50.49 | 67.38 | 73.04 |

Table 7.4: PA values (%) for PF0-MIREX04 & PF0-MIREX05 Vocal data

| DATASET | Content | PR | | TWM | |
|---|---|---|---|---|---|
| | | RAW (%) | DP (%) | RAW (%) | DP (%) |
| ISMIR 2004 | SYNTH | 55.33 | 52.72 | 76.62 | 86.40 |
| | OPERA | 49.63 | 54.36 | 62.09 | 69.34 |
| | POP | 60.24 | 64.35 | 79.42 | 87.78 |
| MIREX 2005 | VOICE | 58.93 | 59.11 | 82.40 | 84.20 |

For the MIREX data again we see that the TWM-DP PDA consistently significantly outperforms the PR-DP PDA. Between the different genres it can be seen that the SYNTH and POP genres give very similar results while the results for the OPERA genre are significantly reduced for the TWM-DP PDA. This is caused because of large octave errors in the female voices in this genre, which contain very high F0s (around 850 Hz). The PA values for TWM-DP for the female-singer files are 40.67 % and 46.07 %. If the lower bound on F0 is increased to 200 Hz for these files, the PA values of TWM-DP jump to 72.60 % and 80.41 %. This is because the overall TWM error is slightly biased towards lower F0s.

#### *7.1.1.2 Comparison of TWM-DP system to another system (Cao, Li, Liu, & Yan, 2007)*

Cao, Li, Liu & Yan (2007) reported PA values for the PF0-MIREX05-T dataset and part (POP**)** of the PF0-ADC04 dataset for their melody extraction system. Their algorithm used a combination of the SHS PDA and post-processing based on harmonic tracking (HT). A comparison of these results with the TWM-DP algorithm is shown in Table 7.5. There is no mention of the F0 search range used by this SHS-HT algorithm. For the TWM-DP algorithm, the window-length and F0 search range were kept fixed as 40 ms, and from 100 to 875 Hz respectively as before.

In Table 7.5 tracks 1 to 13 form the entire 2005 dataset. Pop1 to Pop4 are taken from the PF0-ADC04 dataset. The vocal results indicate the average PA values across tracks 1 to 9 and Pop1 to Pop4. Here $A_p$ and $A_f$ indicate the PA (%) before and after post-processing respectively. The PA for both algorithms is computed using a $1/4^{th}$ tone (50 cent) tolerance. We can see that the TWM-DP algorithm performance is significantly better than the SHS-HT algorithm for the vocal data. Although the performance of the raw TWM & SHS PDAs seems to be on par with each other, the use of the DP post-processing results in PA values that are significantly higher than the HT algorithm.

Table 7.5: Comparison between SHS-HT and TWM-DP algorithms

| File | SHS - HT | | TWM-DP | |
|---|---|---|---|---|
| | $A_f$ (%) | $A_p$ (%) | $A_f$ (%) | $A_p$ (%) |
| track01 | 84.75 | 83.43 | 91.09 | 81.30 |
| track02 | 59.47 | 61.52 | 63.33 | 46.39 |
| track03 | 77.00 | 68.82 | 83.04 | 70.10 |
| track04 | 73.31 | 70.29 | 81.27 | 776.46 |
| track05 | 87.29 | 83.46 | 88.86 | 81.46 |
| track06 | 66.84 | 55.36 | 65.56 | 51.11 |
| track07 | 80.14 | 77.05 | 86.90 | 78.39 |
| track08 | 82.24 | 80.12 | 88.02 | 84.78 |
| track09 | 86.09 | 76.43 | 87.85 | 79.96 |
| track10 | 91.35 | 84.27 | 83.15 | 53.27 |
| track11 | 95.68 | 91.39 | 91.16 | 87.96 |
| track12 | 99.26 | 95.98 | 8.52 | 16.08 |
| track13 | 83.19 | 83.59 | 63.15 | 55.59 |
| pop1 | 78.62 | 75.44 | 84.12 | 72.32 |
| pop2 | 83.32 | 79.34 | 87.94 | 84.49 |
| pop3 | 82.65 | 70.50 | 89.37 | 79.15 |
| pop4 | 88.34 | 75.51 | 90.10 | 83.02 |
| **Overall** | | | | |
| **MIREX 2005** | 82.23 | 78.30 | 75.88 | 67.54 |
| **Vocal** | 79.39 | 74.12 | 84.50 | 75.67 |

However for the instrumental files (tracks 10 to 13) the performance is significantly worse than the SHS-HT algorithm. The PA values are very low e.g. 8 % for track 12. Almost the entire contour is incorrect. The CA value for the same track is 42.17 % indicating that a large number of errors were octave errors. The lead instrument in this particular track is a flute with very high pitch, often exceeding our F0 search range. Such errors are expected since we have designed our algorithm to specifically track vocal pitch contours in polyphonic music.

#### *7.1.1.3 Reducing Computational Complexity of the TWM-DP Algorithm*

A drawback of the above implementation of the TWM-DP PDA is the large processing time required, which stands at about 1.5 to 2 times real-time on a 3.20 GHz Intel Pentium(R) 4 CPU with 1 GB RAM running the Microsoft Windows XP Pro operating system. This is primarily due to the fact that the implementation used so far computes the TWM error at all possible trial F0s ranging from a lower ($F0_{low}$) to upper ($F0_{high}$) value with very small frequency increments (1 Hz). In this section we apply and evaluate the extensions of TWM to

Multi-F0 analysis (as described in Section 4.3). These steps help in reducing computation time as well.

The results after each stage of the modifications are presented in Table 7.6. The first row of the table (Stage 0) presents the PA values and the overall time taken by the current implementation of the TWM-DP algorithm to compute the melody for all the four VTT excerpts of the PF0-ICM dataset, the PF0-ADC04 vocal dataset and the PF0-MIREX05-T vocal dataset. The subsequent rows present the accuracies and computation time for each of the modifications (stages) to the TWM algorithm discussed next. Note that the time taken for the 2004 dataset for the same duration of data will be more because the hop size is smaller (5.8 ms), which results in a larger number of frames for which the pitch is to be estimated.

Table 7.6: Different stages of reducing computation time for the TWM-DP algorithm. Accuracy (%) and time (sec) values are computed for the ICM dataset, the ISMIR 2004 Vocal dataset and the MIREX 2005 Vocal dataset.

| **Stage** | **ICM dataset (250 sec, 10 ms hop)** | | **ISMIR 2004 vocal dataset (231 sec, 5.8 ms hop)** | | **MIREX 2005 dataset (266 sec, 10 ms hop)** | |
|---|---|---|---|---|---|---|
| | **PA (%)** | **Time (sec)** | **PA (%)** | **Time (sec)** | **PA (%)** | **Time (sec)** |
| 0 | 98.2 | 377 | 81.4 | 748 | 82.4 | 510 |
| 1 | 97.9 | 201 | 79.3 | 307 | 80.0 | 210 |
| 2 | 98.1 | 199 | 81.2 | 299 | 82.1 | 205 |
| 3 | 98.1 | 193 | 80.9 | 287 | 82.2 | 198 |
| 4 | 98.2 | 156 | 81.9 | 223 | 82.4 | 153 |

So the first modification made (Stage 1) is to compute possible candidate F0s from the detected spectral peaks and only compute TWM error at these F0s. We include all measured peaks and their sub-multiples (division factors ranging from 2 to 10) that lie within the F0 search range and also distances between consecutive measured peaks that lie within the F0 search range. In all cases (Row 2) there is a large reduction in the processing time after this step but this is also accompanied by a small reduction in PA. On a detailed examination of the F0 candidates computed for erroneous frames for files that showed a marked reduction in PA, it was found that there were a lot of F0 candidates in close vicinity of each other, all having low values of TWM error. However, since the upper limit on the number of candidates used is 20 for all files, this sometimes did not allow candidates with higher TWM error values but that were more likely to be picked by the DP algorithm into the top 20 list. One way to reduce

this error would be to increase the upper limit on the number of candidates but this would again increase the processing time and is avoidable.

Instead, the next modification (Stage 2) involves pruning of the candidates in close vicinity of each other i.e. the candidate with the least error value is retained and all candidates within a 3 % (50 cent) vicinity having higher error values are deleted. The results after this stage (Row 3) show that the PA values indeed increase due to higher selectivity in choosing candidate F0s.

In order to reduce the processing time further, candidate F0s are only computed using sub-multiples of measured component frequencies and not the distance between successive components (Stage 3). This reduces the processing time marginally without a significant reduction in PA (Row 4). Finally, the range of division factors, for sub-multiples of measured peaks, is reduced from 10 to 5 (Stage 4). Now there is a considerable decrease in processing time (Row 5) but also a slight increase in PA. This could be because there are fewer spurious F0 candidates that could be selected by DP on the basis of the smoothness cost alone in frames having weaker predominant F0.

From Stage 0 to Stage 4, the TWM-DP algorithm processing time is reduced by more than a factor of 2 without compromising accuracy. For a 10 ms hop the processing time is well within real-time.

#### *7.1.1.4 Parameter Settings for Male and Female Singers*

Here we investigate the optimal setting of some analysis parameters, such as analysis window-length and the TWM parameter ρ, for male and female singers. For this experiment we use the PF0-Bolly dataset which contains an equal duration of male and female clips. We evaluate the performance of the TWM-DP pitch tracker, in terms of PA and CA, for different parameter settings for the male and female datasets respectively. We experiment with 3 different values of window-length (20, 30 and 40 ms) and ρ (0.1, 0.15 and 0.2). We use a fixed F0 search range [100 1280 Hz] throughout the experiment. The results of these evaluations are given in Table 7.7.

From these results it seems that using values of 40 ms and 0.15 or 0.2 for the window-length and ρ respectively, result in optimal performance for the male dataset, and using values of 20 ms and 0.1 for the window-length and ρ respectively, result in optimal performance for the female dataset. If fully automatic melody extraction is desired then using values of 30 ms and 0.125 for the window-length and ρ respectively lead to an optimal trade-off.

Table 7.7: Pitch tracking performance of TWM-DP system for different parameter setting for male and female sub-sets of PF0-Bolly data

| **Male dataset** | | | | | | |
|---|---|---|---|---|---|---|
| | 20 ms | | 30 ms | | 40 ms | |
| **ρ** | **PA (%)** | **CA (%)** | **PA (%)** | **CA (%)** | **PA (%)** | **CA (%)** |
| 0.1 | 73.1 | 84.9 | 92.1 | 95.2 | 94.6 | 95.8 |
| 0.15 | 82.5 | 89.0 | 94.9 | 95.9 | 95.9 | 96.1 |
| 0.2 | 86.8 | 90.7 | 95.7 | 96.3 | 96.0 | 96.2 |
| **Female dataset** | | | | | | |
| | 20 ms | | 30 ms | | 40 ms | |
| **ρ** | **PA (%)** | **CA (%)** | **PA (%)** | **CA (%)** | **PA (%)** | **CA (%)** |
| 0.1 | 96.0 | 96.2 | 95.9 | 96.8 | 84.8 | 93.8 |
| 0.15 | 96.2 | 96.3 | 88.4 | 95.8 | 59.1 | 88.4 |
| 0.2 | 95.4 | 96.2 | 72.4 | 93.6 | 35.0 | 78.6 |

## 7.1.2 Singing Voice Detection Evaluations[1]

Here we evaluate our singing voice detection design by first comparing different feature sets in a training-testing framework using the SVD-Hind database. Next we compare different parameter settings for the chosen feature set using the PF0-ADC04 vocal and PF0-MIREX05-T vocal datasets.

### *7.1.2.1 Feature Set Comparison*

*Data*

The data used in this evaluation is a subset of the SVD-Hind dataset. This set is divided into training and testing data. For testing we also used one other feature set – the VTTH examples from the PF0-ICM dataset. In this second dataset a loud harmonium is present. The duration information of the vocal and instrumental segments across the training and testing datasets are shown in Table 7.8 and Table 7.9 respectively. The overall duration of the training and the two testing datasets is about 23 min, 7.5 min and 4.5 min respectively. As we can observe the vocal segment comprises nearly 75% of the data.

*Experimental Setup*

First, 10-fold cross validation (CV) is used to evaluate the overall performance of the classifier for different parameters. In *k*-fold cross validation the training set is divided in *k*

[1] These investigations were carried out with the help of S. Ramakrishnan

parts. (*k*-1) parts are used for training, while the remaining one part is used for testing. The process is iterated *k* times, using each of the *k* parts exactly once for testing. The idea behind the iteration is the improvement of the estimate of the empirical error, which becomes the average of errors on each part of the training set. Furthermore this technique ensures that all data is used for training as well as model selection purposes. The major application of this technique is to select the classifier parameters like number of GMM components. A variable number of mixtures per sub-class are used to train the classifier. The cross validation results are used to quantify the performance of the classifier for different classifier parameters. The number of GMM mixtures tested for each sub-class (Vocal, Instrumental) was [4, 4] and [8, 8]. Results for either of these were not found to vary much so the remaining experiments use 4 mixtures per class.

Once the GMM parameters are fixed the classifier is trained with the entire training data and the model was tested on files not used for training (testing data set). This is used to simulate the performance of the model on external unknown data.

For the 10-fold CV and the unseen testing data we compute confusion matrices, which contain the vocal and instrumental recalls individually, to quantify the classifier performance. In the case of 10-fold cross validation means and standard deviation across the individual folds are used as a measure of classifier performance. Means of recalls quantify the performance of the classifier while the standard deviation is used to measure the uniformity of performance across different folds. For the unseen testing data, the classifier performance is measured by the means of accuracies across all the files.

*Feature Sets*

All features described in this section have been extracted using a 40 ms long analysis window and the same hop as used in melody extraction (10 ms). Three sets of features are used. The first (FS1) consists of MFCC coefficients (Logan, 2000). It was observed that there is a very insignificant increase in accuracy, in 10-fold CV, after increasing the number of coefficients above 13. Hence 13 MFCC coefficients were chosen for the rest of the experiments.

Table 7.8: Duration information of training data from SVD-Hind dataset

| **Segments** | **Number** | **Average duration** | **Total duration** |
|---|---|---|---|
| Vocal | 187 | 5.44 s | 1018.01 s |
| Non-Vocal | 208 | 1.73 s | 359.62 s |

Table 7.9: Duration information of testing data from SVD-Hind dataset

| | SVD-Hind (Testing) | | | PF0-ICM (VTTH) | | |
|---|---|---|---|---|---|---|
| **Segments** | **Number** | **Average duration** | **Total duration** | **Number** | **Average duration** | **Total duration** |
| Vocal | 75 | 4.35 s | 326.01 s | 39 | 5.6 s | 219.9 s |
| Non-Vocal | 80 | 1.33 s | 106.54 s | 43 | 1.2 s | 50.4 s |

The second feature set (FS2) consists of acoustic features described in Section 6.2.2.1. A combination of 7 features was found to give the highest 10-fold CV accuracy. These were normalized sub-band energy (SBE), normalized flux (FLUX), audio spectral flatness (ASF), energy ratio (ER), spectral spread (SPR), spectral roll-off (ROLL) and audio spectral centroid (ASC). Table 7.10 shows the mutual information (MI) values of each of the features computed using the training data. Here MI values have been computed assuming the vocal and instrumental class probabilities to be 0.7 and 0.3 respectively. The final set of 7 features was arrived in two stages. First all features were ranked in descending order of their mutual information values for the training data. Then 10-fold CV experiments, again with the training data, were carried out by adding one feature at a time to the feature vector. The set of features that demonstrated the best overall classification accuracy in this experiment was chosen as the final feature set.

Table 7.10: Features and their corresponding MI values

| **Feature set** | **Feature name** | **MI** |
|---|---|---|
| FS2 | Normalized sub-band energy (SBE) | 0.456 |
| | Normalized flux (FLUX) | 0.498 |
| | Audio spectral flatness (ASF) | 0.451 |
| | Energy ratio (5-8/0-1.5) kHz (ER) | 0.350 |
| | Spectral spread (SPR) | 0.319 |
| | Spectral roll-off (ROLL) | 0.132 |
| | Audio spectral centroid (ASC) | 0.157 |
| FS3 | Normalized harmonic energy (NHE) | 0.682 |

The third feature set (FS3) consists of a single pitch-based feature – the normalized harmonic energy (NHE), as described in Section 6.2.2.2. The motivation behind using a pitch-based feature was the fact that voiced utterances, which show harmonic structure, composed 96 % of the sung regions in the ICM database, and any cue to harmonicity, such as pitch, would be useful for singing voice detection. The predominant-F0 used in the computation of NHE is extracted from the TWMDP system evaluated in the previous section for the SVD-Hind datasets. In the case of the PF0-ICM (VTTH) dataset we used the ground-truth voice and instrument F0s extracted from the single-channel monophonic files for the voice and harmonium. The MI value for NHE was found to be 0.682, the highest over any of the features from FS2, as seen in Table 7.10.

*Post-Processing*

Here we use the system evaluated in Section 6.4.3. This involves grouping the frame-level classification labels over automatically detected boundaries. This grouping can either be done by a process of majority voting or by combining frame-level log-likelihoods as output by the classifier. Experiments have shown that both methods of grouping yield very similar results. For the current experiment, the process of majority voting has been used. The boundaries are detected by the use of a novelty detector, which takes the NHE feature as input, and uses a similarity matrix and Gaussian Difference Kernel (GDK) to output a novelty function. Peaks in the novelty function above a threshold are marked as boundary locations. Boundaries closer than some minimum threshold are further pruned. Values of the GDK length, novelty threshold and boundary pruning threshold used are 500 ms, 0.15 and 150 ms respectively.

*Results*

Classification accuracies for the 10-fold CV and the testing data for the different feature sets are given in Table 7.11 and Table 7.12 respectively. For the 10-fold CV both vocal and instrumental accuracies are the highest for FS3. For the experiments using the SVD-Hind testing data, although the vocal accuracy of FS3 is marginally lower than FS2 or FS1, the instrumental accuracy is significantly higher. For the PF0-ICM (VTTH) dataset all the features perform poorly in terms of instrument classification because of the presence of the loud, timbrally similar harmonium. We have proposed one solution to this in terms of pre-processing for instrument suppression (presented in the Appendix in Section A.2). However such an approach is restricted to flat-note instrument suppression. The use of timbral & F0-harmonic dynamics for dealing with such cases is evaluated in the next chapter.

Since the NHE feature is non-timbral in nature, we also attempted to append it with individual timbral features from FS2. Each of these features were added individually to NHE to compare the classifier performance. The 10-fold CV results are shown in Table 7.13. From the table it can be observed that the accuracy does not increase much with addition of any feature to with NHE.

Table 7.14 shows the result of grouping at the segment level over automatic boundaries and ideal boundaries with majority voting for testing database. Ideal boundaries refer to the vocal and instrumental phrase boundaries, which have been obtained as the output of the manual labeling process. Automatic boundaries refer to those boundaries as output by the audio novelty based boundary detector. It can be seen that there is an improvement in classifier performance after grouping if the underlying classification is good. The performance of the segment grouping over automatic boundaries is in-between the frame level classification and the ideal boundary detector results. Ideal boundaries can give an idea of the maximum possible improvement in classifier performance with ideal grouping. Hence improvement in performance of the boundary detector can result in improvement in accuracy.

Table 7.11: Comparison of 10-fold CV accuracies for different feature sets

| **Feature set** | **Vocal recall (%)** | **Instrumental recall (%)** | **Overall recall (%)** |
|---|---|---|---|
| FS1 | 77.16 ± 11.83 | 76.88 ± 13.50 | 77.09 ± 6.46 |
| FS2 | 83.38 ± 11.67 | 77.57 ± 13.09 | 81.86 ± 6.97 |
| FS3 | 87.46 ± 5.00 | 88.43 ± 7.50 | 87.71 ± 3.69 |

Table 7.12: Comparison of testing database results for different feature sets

| | **SVD-Hind (Testing)** | | **PF0-ICM (VTTH)** | |
|---|---|---|---|---|
| **Feature set** | **Vocal recall (%)** | **Instrumental recall (%)** | **Vocal recall (%)** | **Instrumental recall (%)** |
| FS1 | 92.17 | 66.43 | 91.61 | 40.91 |
| FS2 | 92.38 | 66.29 | 87.53 | 57.40 |
| FS3 | 89.05 | 92.10 | 86.60 | 45.22 |

Table 7.13: Combination of NHE with other acoustic features

| Feature List | V/UV Confusion Matrix | | Overall accuracy (%) |
|---|---|---|---|
| [NHE] | 87.46 ± 5 | 12.54 | **87.71 ± 3.70** |
| | 11.57 | 88.43 ± 7.51 | |
| [NHE SBE] | 85.16 ±6.01 | 14.84 | 86.25 ± 4.46 |
| | 10.66 | 89.34±7.76 | |
| [NHE FLUX] | 86.73 ± 6.64 | 13.27 | 87.16 ± 5.3 |
| | 11.62 | 88.38 ± 7.33 | |
| [NHE ASF] | 87.3 ± 6.29 | 12.7 | 87.44 ± 4.46 |
| | 12.18 | 87.82 ± 6.13 | |
| [NHE ER] | 87.22 ± 5.34 | 12.78 | 87.74 ± 4.37 |
| | 10.78 | 89.22 ± 7.01 | |
| [NHE SPR] | 86.86 ± 5.34 | 13.14 | 87.33 ± 3.86 |
| | 11.35 | 88.65 ± 7.51 | |
| [NHE ROLL] | 85.03 ± 5.25 | 14.97 | 86.55 ± 4.1 |
| | 9.17 | 90.83 ± 5.85 | |
| [NHE ASC] | 86.61 ± 5.47 | 13.39 | 87.24 ± 4.31 |
| | 10.99 | 89.01 ± 6.95 | |

Table 7.14: Comparison of classifier performance with automatic and ideal boundaries for testing database for SVD-Hind dataset

| Feature set | After grouping (Automatic boundaries) | | After grouping (ideal boundaries) | |
|---|---|---|---|---|
| | Vocal accuracy (%) | Instrumental accuracy (%) | Vocal accuracy (%) | Instrumental accuracy (%) |
| FS1 | 97.86 | 61.61 | 99.00 | 68.27 |
| FS2 | 96.28 | 68.98 | 99.00 | 69.06 |
| FS3 | 93.15 | 96.58 | 100 | 99.14 |

### *7.1.2.2 Experiment for setting NHE Threshold*

From the experiments with Hindustani music in the previous section it appears that using a single feature results in the best singing voice detection performance. The GMM classifier operates as a threshold based classifier when using a single feature. This threshold was found to be -18.3048 dB for the classifier trained with the SVD-Hind training data. Here different values of this threshold have been experimented with from -15 to -19 dB in steps of 1 dB. Singing voice detection results using these different voicing thresholds for the PF0-ADC04 & PF0-MIREX05 datasets are shown in Table 7.15 and Table 7.16 respectively. The evaluation metrics used to evaluate singing voice detection mechanisms at MIREX were voicing detection rate (Recall) and voicing false alarm rate (FalseAlm). The voicing detection rate is the proportion of frames labeled voiced in the reference transcription that are estimated to be

Table 7.15: Comparison of Voicing detection Recall and False alarm for vocal, non-vocal and overall ISMIR 2004 dataset for different voicing thresholds

| Voicing Threshold (dB) | Vocal | | Non-Vocal | | Overall | |
|---|---|---|---|---|---|---|
| | Recall (%) | FalseAlm (%) | Recall (%) | FalseAlm (%) | Recall (%) | FalseAlm (%) |
| -15 | 79.88 | 24.52 | 82.17 | 22.25 | 80.76 | 23.84 |
| -16 | 82.51 | 30.76 | 85.01 | 26.55 | 83.47 | 29.50 |
| -17 | 85.31 | 36.98 | 87.56 | 31.03 | 86.17 | 35.20 |
| -18 | 87.40 | 42.55 | 89.54 | 34.85 | 88.23 | 40.24 |
| -19 | 89.34 | 47.81 | 90.94 | 38.45 | 89.96 | 45.00 |

Table 7.16: Comparison of Voicing detection Recall and False alarm for vocal, non-vocal and overall MIREX 2005 dataset for different voicing thresholds

| Voicing Threshold (dB) | Vocal | | Non-Vocal | | Overall | |
|---|---|---|---|---|---|---|
| | Recall (%) | FalseAlm (%) | Recall (%) | FalseAlm (%) | Recall (%) | FalseAlm (%) |
| -15 | 80.25 | 24.88 | 96.37 | 45.52 | 85.04 | 32.38 |
| -16 | 83.47 | 29.56 | 97.42 | 50.76 | 87.62 | 37.27 |
| -17 | 86.29 | 34.55 | 98.12 | 56.60 | 89.81 | 42.56 |
| -18 | 88.38 | 39.43 | 98.74 | 62.54 | 91.46 | 47.83 |
| -19 | 89.99 | 44.50 | 99.17 | 68.26 | 92.72 | 53.14 |

voiced by the algorithm. The voicing false alarm rate is the proportion of frames that are not voiced (melody silent) according to the reference transcription that are estimated to be voiced by the algorithm. It can be seen that raising this threshold reduces the false alarms at the expense of recall.

## 7.2 MIREX Evaluations

The problem of melody extraction from polyphonic audio, involving the detection and extraction of the pitch contour of the lead melodic instrument, has received considerable attention from researchers in the past; as reflected by the large number of entries for audio melody extraction task in the MIREX 2005 & 2006 evaluations. Recently however several researchers have shifted focus to the multi-F0 detection problem i.e. the estimation of the melodic contours of multiple instruments that sound simultaneously; as reflected by the lack of a melody extraction task in the MIREX 2007 evaluations. The difference in the above two tasks lies primarily in the nature of the signals operated upon. While the Multi-F0 data can have two or more instruments playing different melodies simultaneously, the melody

extraction data has a more clear melody vs. background distinction with the possibility of the background being more complex and rich than that for the multi-F0 problem. However melody extraction is still not a solved problem and in order to assess the performance of current systems this task was resurrected and raised considerable interest at the 2008, 2009 and 2010 MIREX evaluations.

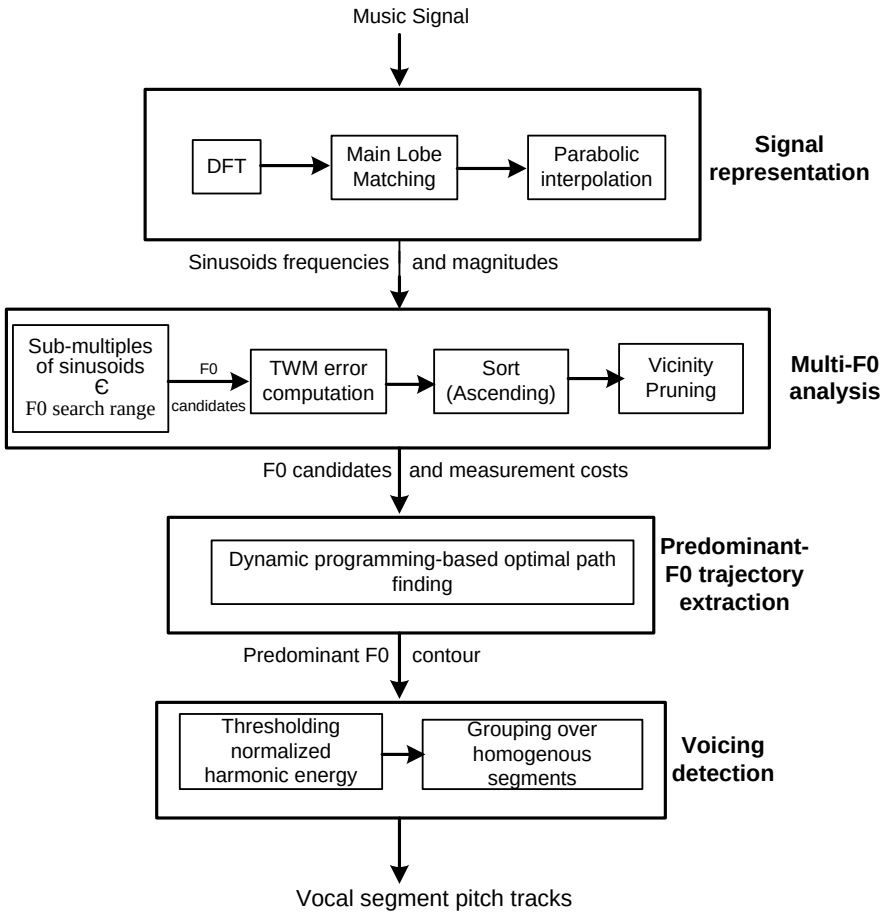


Figure 7.1: Block diagram of submission to MIREX'08 &'09

### 7.2.1 Overview of System Submitted to MIREX

Figure 7.1 shows the block diagram of our system. Table 7.17 shows the parameter values for each of the modules. Briefly, the signal representation module transforms short time frames of the signal into the frequency domain via a DFT (using a frame and hop length of 30 ms and 10 ms resp.), and then identifies sinusoids by applying a main-lobe matching technique. *S* (0.6) is the threshold for sinusoidality. The upper limit on spectral content is 5 kHz. The sinusoid parameters (frequencies and amplitudes) are refined using parabolic interpolation. The Multi-F0 extraction first identifies potential F0 candidates as sub-multiples of well

formed sinusoids ($S$>0.8). For each of these candidates a normalized TWM error value is computed. p, q, r & ρ are TWM parameters. The F0 candidates are then sorted in ascending order and weaker candidates that are close (25 cents) to stronger candidates are pruned. The maximum allowed number of candidates for a single frame (MaxCand) is 20. The final list of candidates and their associated TWM error values is input to a DP –based path finding stage, which uses a Gaussian log smoothness cost (σ=0.1). The output of this stage is the predominant-F0 contour and associated NHE values. The singing voice detection stage then applies a fixed threshold to the NHE values followed by smoothing over boundaries detected by an audio novelty based boundary detector to output the final voice-pitch contour. For the 2008 submission we had used an NHE threshold of -15 dB but we found that our performance improved when we increased this threshold to -18 dB, which was the value used in the MIREX 2009 submission. The Signal representation & Multi-F0 analysis blocks were parallelized for execution on multi-core systems so as to reduce run-time.

Table 7.17: Parameter values for the melody extraction system submission to MIREX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Signal representation | Window | Hop | Fmax | S | | | |
| | 30 ms | 10 ms | 5 kHz | 0.6/0.8 | | | |
| Multi-F0 analysis | F0 low | F0 high | p | q | r | ρ | MaxCand |
| | 100 Hz | 1280 Hz | 0.5 | 1.4 | 0.5 | 0.1 | 20 |
| Predominant-F0 extraction | σ | | | | | | |
| | 0.1 | | | | | | |
| Voicing detection | NHE Thresh | GDK length | Novelty threshold | Duration threshold | | | |
| | -15/-20 | 300 ms | 0.15 | 100 ms | | | |

## 7.2.2 Datasets, Evaluation Metrics and Approaches

### *7.2.2.1 Datasets*

The MIREX evaluations use the PF0-ADC04, PF0-MIREX05-S, PF0-MIREX08 and PF0-MIREX09 datasets for evaluations. These datasets have been described in Chapter 2. The last two datasets are completely vocal in nature. However the first two datasets also have audio clips in which the lead instrument is not the singing voice. Since our system is designed specifically for singing voice processing we are only concerned with the results for the vocal part of these two datasets.

The Audio Melody Extraction task was run at MIREX 2010, 2009, 2008 & 2005. In 2006 & 2007 there was not enough interest in the task; the evaluations for a given task are

only run if 3 or more participants express interest in that task. The PF0-ADC04, PF0-MIREX05-S datasets were used in all four evaluations. The PF0-MIREX08 dataset was used 2008 onwards and the PF0-MIREX09 dataset was used 2009 onwards.

#### *7.2.2.2 Evaluation Metrics*

For the predominant-F0 evaluation the metrics computed are pitch accuracy (PA) & chroma accuracy (CA). For singing voice detection the metrics computed are vocal recall and vocal false alarm rate. The evaluation of the predominant-F0 extraction stage is made independent of the voicing stage by the use of positive and negative pitch values to indicate vocal and non-vocal frames respectively. The overall accuracy is computed as the percentage of actually voiced frames which have been detected as voice and whose F0 is correctly detected. Additionally the run-times for each algorithm are also recorded and made available.

#### *7.2.2.3 Approaches*

Almost all the submissions to the MIREX AME task for all years follow the paradigm described in Figure 1.1 except for the submissions of Brossier (2006), who used an extension of the YIN algorithm (de Cheveigne & Kawahara, 2002), and Durrieu, Richard & David (2008; 2009), who used generative models for the singing voice and accompaniment signals. Tachibana, Ono, Ono & Sagayama (2009; 2010) and Hsu & Jang (2010) are the only ones to use a harmonic/percussive source-separation driven pre-processing stage to first enhance melodic content in the signal. All submissions that incorporate a spectral analysis stage use either a fixed- or multi-resolution STFT based signal representation stage. Multi-resolution analysis is carried out by means of a constant-Q transform (Tachibana, Ono, Ono, & Sagayama, 2009; 2010; Cancela, 2008; 2009) multi-resolution FFT (Dressler, 2006; 2010; Hsu & Jang, 2010) or the optimal-QSTFT (Wendelboe, 2009). The only submission to use an auditory-model front-end was Rynnanen (2006). The only submission to use an adaptive window-length scheme, where the window adaptation is driven by the autocorrelation coefficient is by Joo, Jo & Yoo (2009; 2010). For multi-F0 analysis almost all approaches, directly or indirectly, implement a “harmonic-sieve” salience function (Poliner et. al., 2007) except for Hsu & Jang (2010), who use a vibrato/tremolo driven salience function. For predominant-F0 trajectory estimation either a path finding (DP or viterbi) or partial tracking/auditory streaming approach is adopted by all submissions of which the latter is more popular. Submissions that adopt partial tracking/auditory streaming use a cumulative salience measure of each source-track for melody line identification. Other submissions use a gate on

cumulative harmonic energy (Tachibana, Ono, Ono, & Sagayama, 2010) or a machine-learning approach (Hsu, Jang, & Chen, 2009; Hsu & Jang, 2010). Not all algorithms apply a melody identification stage (Tachibana, Ono, Ono, & Sagayama, 2009; Wendelboe, 2009).

Table 7.18: Audio Melody Extraction Results Summary- ADC 2004 dataset – Vocal. **vr** and **rr** indicate our submission in 2008 and 2009 respectively.

| **2006** | | | | | | |
|---|---|---|---|---|---|---|
| | Vx Recall (%) | Vx False Alm (%) | Raw pitch (%) | Raw Chroma (%) | Overall Accuracy (%) | RunTime (seconds) |
| dressler | 89.8 | 10.9 | 77.1 | 78.0 | 77.3 | 27 |
| ryynanen | 85.9 | 11.5 | 78.3 | 79.3 | 76.2 | 440 |
| poliner | 88.4 | 34.5 | 65.4 | 69.0 | 64.7 | N.A. |
| sutton | 90.8 | 32.0 | 67.5 | 68.0 | 64.2 | 5014 |
| brossier | 99.8 | 93.9 | 56.3 | 63.5 | 46.7 | 30 |
| **2008** | | | | | | |
| pc | 91.4 | 11.0 | 89.5 | 89.8 | 88.3 | 33974 |
| drd2 | 92.7 | 30.9 | 87.4 | 87.6 | 83.2 | 2082 |
| rk | 89.2 | 29.3 | 84.0 | 84.2 | 79.9 | 73 |
| **vr** | **80.2** | **20.9** | **81.6** | **88.1** | **72.0** | **110** |
| drd1 | 95.4 | 59.8 | 77.2 | 83.4 | 69.6 | 5 days |
| clly2 | 87.3 | 53.1 | 71.8 | 72.3 | 64.9 | 39 |
| clly1 | 62.5 | 32.2 | 71.8 | 72.3 | 52.3 | 36 |
| **2009** | | | | | | |
| cl1 | 95.84 | 79.36 | 85.63 | 86.21 | 75.16 | |
| cl2 | 88.08 | 51.91 | 85.63 | 86.21 | 75.32 | |
| dr1 | 92.36 | 49.25 | 86.96 | 87.40 | 79.99 | |
| dr2 | 86.62 | 28.47 | 83.26 | 85.19 | 78.07 | |
| hjc1 | 50.02 | 25.38 | 63.11 | 74.10 | 48.77 | |
| hjc2 | 50.02 | 25.38 | 46.52 | 64.47 | 44.92 | |
| jjy | 85.83 | 39.05 | 81.96 | 85.80 | 74.68 | |
| kd | 91.63 | 14.98 | 85.97 | 86.42 | 85.87 | |
| mw | 99.94 | 98.44 | 83.14 | 86.59 | 70.96 | |
| pc | 88.78 | 19.36 | 86.96 | 87.55 | 85.95 | |
| **rr** | **92.83** | **59.18** | **81.45** | **88.04** | **73.77** | |
| toos | 99.91 | 97.79 | 59.77 | 72.13 | 50.75 | |
| **2010** | | | | | | (dd:hh:mm) |
| HJ1 | 73.43 | 20.99 | 76.83 | 79.76 | 61.33 | 00:30:53 |
| TOOS1 | 79.58 | 32.77 | 62.85 | 73.38 | 53.68 | 00:26:06 |
| JJY2 | 93.65 | 50.11 | 79.64 | 85.27 | 71.94 | 00:36:17 |
| JJY1 | 93.65 | 50.15 | 76.87 | 83.79 | 69.64 | 01:05:58 |
| SG1 | 80.55 | 23.18 | 75.22 | 78.22 | 69.93 | 00:04:26 |

### 7.2.3 MIREX Results

Results of the MIREX evaluations for all years along with the extended abstracts of individual submissions can be accessed via the MIREX wiki at http://www.music-ir.org/mirex/wiki/MIREX_HOME. Table 7.18 and Table 7.19 present the results on the PF0-ADC'04 and PF0-MIREX05-S vocal datasets over all four MIREX AME evaluations. Table 7.20 presents the results for the PF0-MIREX08 dataset over three MIREXs starting from 2008 and Table 7.21 presents the results for the PF0-MIREX09 dataset over the two MIREXs starting from 2009. In each table with bold values indicate the performance of our submission.

Table 7.19: Audio Melody Extraction Results Summary - MIREX 2005 dataset – Vocal. **vr** and **rr** indicate our submission in 2008 and 2009 respectively.

| **2006** | | | | | | |
|---|---|---|---|---|---|---|
| | Vx Recall (%) | Vx False Alm (%) | Raw pitch (%) | Raw Chroma (%) | Overall Accuracy (%) | Timing (seconds) (%) |
| dressler | 85.5 | 28.7 | 78.5 | 81.6 | 73.7 | 48 |
| ryynanen | 77.0 | 15.6 | 75.7 | 76.9 | 72.5 | 773 |
| poliner | 93.7 | 44.3 | 69.1 | 70.6 | 65.0 | N.A. |
| sutton | 71.8 | 12.3 | 70.7 | 71.6 | 67.3 | 8195 |
| brossier | 99.6 | 97.9 | 42.7 | 53.5 | 30.7 | 58 |
| **2008** | | | | | | |
| pc | 83.1 | 19.7 | 77.6 | 78.1 | 76.8 | 61470 |
| **vr** | **81.5** | **20.5** | **76.1** | **79.2** | **71.8** | **200** |
| rk | 89.5 | 45.0 | 74.6 | 76.6 | 67.6 | 115 |
| drd2 | 92.6 | 51.1 | 75.0 | 77.9 | 66.7 | 3294 |
| clly2 | 81.5 | 46.0 | 70.1 | 72.8 | 63.7 | 57 |
| clly1 | 75.1 | 37.5 | 70.1 | 72.8 | 62.7 | 56 |
| drd1 | 91.7 | 63.6 | 54.7 | 62.4 | 48.0 | 3 days |
| **2009** | | | | | | |
| cl1 | 91.23 | 67.71 | 70.81 | 73.92 | 59.71 | |
| cl2 | 80.95 | 44.81 | 70.81 | 73.92 | 64.46 | |
| dr1 | 93.75 | 53.53 | 76.11 | 77.71 | 66.96 | |
| dr2 | 88.06 | 38.02 | 70.93 | 75.92 | 66.52 | |
| hjc1 | 65.84 | 19.82 | 62.66 | 73.49 | 54.85 | |
| hjc2 | 65.84 | 19.82 | 54.13 | 69.42 | 52.29 | |
| jjy | 88.85 | 41.98 | 76.27 | 79.32 | 66.31 | |
| kd | 82.60 | 15.26 | 77.46 | 80.82 | 76.96 | |
| mw | 99.93 | 99.79 | 75.74 | 80.38 | 53.73 | |
| pc | 75.64 | 21.09 | 71.71 | 72.51 | 70.47 | |
| **rr** | **92.91** | **56.36** | **75.95** | **79.11** | **65.77** | |
| toos | 99.88 | 99.54 | 73.43 | 77.81 | 52.10 | |
| **2010** | | | | | | (dd:hh:mm) |
| HJ1 | 70.80 | 44.90 | 71.70 | 74.93 | 53.89 | 00:59:31 |
| TOOS1 | 84.65 | 41.93 | 68.87 | 74.62 | 60.84 | 00:50:31 |
| JJY2 | 96.86 | 69.62 | 70.17 | 78.32 | 60.81 | 01:09:30 |
| JJY1 | 97.33 | 70.16 | 71.58 | 78.61 | 61.54 | 03:48:02 |
| SG1 | 76.39 | 22.78 | 61.80 | 73.70 | 62.13 | 00:08:15 |

Table 7.20: Audio Melody Extraction Results Summary - MIREX 2008 dataset. **vr** and **rr** indicate our submission in 2008 and 2009 respectively.

| **2008** | | | | | |
|---|---|---|---|---|---|
| | Vx Recall (%) | Vx False Alm (%) | Raw Pitch (%) | Raw Chroma (%) | Overall Accuracy (%) |
| drd1 | 98.7 | 68.9 | 85.8 | 88.3 | 76.0 |
| rk | 90.4 | 48.5 | 83.5 | 83.8 | 75.3 |
| drd2 | 96.6 | 57.1 | 81.8 | 82.6 | 75.0 |
| pc | 95.7 | 65.4 | 83.9 | 84.0 | 73.3 |
| **vr** | **68.3** | **21.1** | **88.2** | **88.6** | **66.7** |
| clly1 | 77.6 | 43.4 | 54.7 | 55.3 | 51.4 |
| clly2 | 86.1 | 67.8 | 54.7 | 55.3 | 49.7 |
| **2009** | | | | | |
| cl1 | 93.39 | 90.54 | 50.81 | 51.33 | 45.34 |
| cl2 | 84.42 | 66.15 | 50.81 | 51.33 | 46.81 |
| dr1 | 97.00 | 51.09 | 88.01 | 88.17 | 81.18 |
| dr2 | 94.19 | 39.30 | 86.58 | 86.81 | 80.03 |
| hjc1 | 55.93 | 7.73 | 67.56 | 74.87 | 48.07 |
| hjc2 | 55.93 | 7.73 | 60.84 | 74.81 | 46.51 |
| jjy | 85.16 | 38.91 | 68.30 | 81.88 | 61.16 |
| kd | 95.07 | 53.31 | 87.82 | 88.82 | 80.65 |
| mw | 100.00 | 99.87 | 85.99 | 88.87 | 73.50 |
| pc | 94.05 | 66.10 | 81.83 | 81.98 | 73.64 |
| **rr** | **94.82** | **48.20** | **86.16** | **86.67** | **78.97** |
| toos | 100.00 | 99.26 | 79.76 | 83.66 | 68.50 |
| **2010** | | | | | |
| HJ1 | 89.81 | 22.82 | 86.00 | 86.76 | 76.76 |
| TOOS1 | 85.46 | 23.08 | 82.40 | 86.23 | 72.04 |
| JJY2 | 95.05 | 44.64 | 88.55 | 90.35 | 79.63 |
| JJY1 | 95.05 | 44.64 | 88.45 | 90.01 | 79.54 |
| SG1 | 86.16 | 23.16 | 85.66 | 86.94 | 77.70 |

### 7.2.4 Analysis of MIREX Results

The consistently top-performing algorithms across the ADC'04, PF0-MIREX05-S and PF0-MIREX08 datasets in terms of predominant-F0 extraction, voicing detection and overall accuracy are that of Dressler (2009) and Cancela (2008). The chroma accuracy values for *our submission* (indicated by **vr** and **rr** for 2008 and 2009 respectively) for the ADC'04, PF0-MIREX05-S and PF0-MIREX08 datasets indicate that the predominant F0 extraction performance of our system is *on-par with*, if not, *the best performing system*, although our system does make some octave errors. The CA values of our system for the PF0-MIREX09 dataset are lower than the best-performing algorithms. However it seems that these latter

submissions were particularly tuned for this (publicly available) dataset as they exhibit significantly lower CA values for the other datasets.

The low values of overall accuracy for our submission in 2008 are not particularly disturbing because the threshold for the NHE feature chosen in the system results in significantly lesser number of false alarms than all other systems, but also lower voicing recall. The recall can easily be increased, at the expense of increasing the false alarms, by lowering the threshold for the NHE feature. In our 2009 submission we reduced the NHE threshold parameter from -15 to -18 dB. This resulted in higher overall accuracy for the same datasets in the 2009 evaluations due to a larger vocal bias in the data.

Table 7.21: Audio Melody Extraction Results Summary - MIREX 2009 dataset. **rr** indicates our submission.

| **2009** | | | | | | |
|---|---|---|---|---|---|---|
| Participant | Vx Recall (%) | Vx False Alm (%) | Raw Pitch (%) | Raw Chroma (%) | Overall Accuracy (%) | Runtime (seconds) |
| cl1 | 92.49 | 83.57 | 59.14 | 62.95 | 43.97 | 00:00:28 |
| cl2 | 77.21 | 59.74 | 59.14 | 62.95 | 49.23 | 00:00:33 |
| dr1 | 91.87 | 55.36 | 69.88 | 72.51 | 60.13 | 16:00:00 |
| dr2 | 87.40 | 47.34 | 66.55 | 70.79 | 59.51 | 00:08:44 |
| hjc1 | 34.17 | 1.79 | 72.66 | 75.29 | 53.18 | 00:05:44 |
| hjc2 | 34.17 | 1.79 | 51.69 | 70.00 | 51.75 | 00:09:38 |
| jjy | 38.91 | 19.41 | 75.94 | 80.25 | 49.69 | 02:14:06 |
| kd | 91.18 | 47.78 | 80.46 | 81.88 | 68.22 | 00:00:24 |
| mw | 99.99 | 99.47 | 67.29 | 71.00 | 43.64 | 00:02:12 |
| pc | 73.12 | 43.48 | 50.89 | 53.37 | 51.50 | 03:05:57 |
| **rr** | **88.81** | **50.76** | **68.62** | **71.37** | **60.77** | **00:00:26** |
| toos | 99.98 | 99.42 | 82.29 | 85.75 | 53.56 | 01:00:28 |
| **2010** | | | | | | |
| HJ1 | 82.06 | 14.27 | 83.15 | 84.23 | 76.17 | 14:39:16 |
| TOOS1 | 94.17 | 38.58 | 82.59 | 86.18 | 72.23 | 12:07:21 |
| JJY2 | 98.33 | 70.62 | 81.29 | 83.83 | 62.55 | 14:06:20 |
| JJY1 | 98.34 | 70.65 | 82.20 | 84.57 | 62.90 | 65:21:11 |
| SG1 | 89.65 | 30.22 | 80.05 | 85.50 | 73.59 | 01:56:27 |

## 7.3 Summary

In this chapter we first internally evaluated the predominant-F0 extraction and singing voice detection systems separately using some Hindustani classical music data and also some publicly available music data. These evaluations resulted in the design and analysis parameter

selection of our melody extraction system (Figure 7.1), which was then submitted to the MIREX 2008 & 2009 evaluations. At these evaluations it was found that our system performed on-par with the best performing systems, both in terms of predominant-F0 extraction as well as singing voice detection.

However, as described in Section 5.2, our system is not robust to music with pitched accompaniment whose volume is comparable to that of the singing voice. On close analysis of the performance of our system for the available MIREX audio data it was found that reduced pitch accuracies occurred for instances of vocal music with loud pitched accompaniment. This observation is consistent with Dressler's analysis of the MIREX melody extraction results in which she states that performance of vocal melody extraction systems degrade when "the instrumental accompaniment naturally reaches the volume of the softer human melody parts."(Dressler, 2010). Consequently, in the next chapter we evaluate specific enhancements made to the predominant-F0 trajectory extraction and singing voice detection modules that were designed to address the loud pitched accompaniment problem.

# Chapter 8

## Melody Extraction System Evaluations – Part II

In the previous chapter, the performance of our vocal melody extraction algorithm was shown to be on-par with state-of-the-art systems at the MIREX evaluations. However like most other systems, our system too faces problems when presented with polyphonic signals with loud pitched accompaniment that compete for local dominance with the singing voice. In this chapter we evaluate enhancements to our system for addressing this specific problem. First we evaluate the performance of enhancements to the predominant-F0 extraction system, namely the extension of the system to simultaneously tracking a pair of F0 trajectories instead of a single trajectory and the subsequent identification of the voice pitch contour (described in Chapter 5), in comparison to another state-of-the art system that has also been designed to deal with competing pitched sounds. For this comparison we use western and Hindustani music data which is representative of the loud accompaniment problem. Next we evaluate the performance of enhancements to the singing voice detection system, namely the use of a predominant-F0 based source spectral isolation stage prior to feature extraction and the

combination of static and dynamic feature sets (described in Chapter 6). The performance of the system with these enhancements is compared to the performance of the system with a baseline feature set, previously shown in the literature to be superior to other features for singing voice detection, for distinct cross-cultural music datasets with loud accompaniment.

## 8.1 Evaluations of Enhancements to Predominant-F0 Trajectory Extraction for Loud Pitched Accompaniment

Here we present an experimental evaluation of our dual-F0 tracking enhancements to the predominant F0 extraction stage, hereafter referred to as the TWMDP system, as compared to another state-of-the-art singing voice melody extraction system on three different sets of polyphonic vocal music. This other algorithm is the one proposed by Li & Wang (2007), hereafter referred to as the LIWANG system, who have made their program code available on the internet.

### 8.1.1 Systems Overview

#### *8.1.1.1 TWMDP System*

Figure 8.1 shows a detailed block diagram of the proposed system of which the three main modules are the computation of a suitable signal representation, multi-F0 analysis and predominant-F0 trajectory extraction. The signal representation and multi-F0 analysis stages are the same as the system submitted to MIREX. The predominant-F0 trajectory extraction stage contains novel enhancements to an existing dynamic programming-based framework that involves harmonically constrained F0 candidate pairing and the simultaneous tracking of these F0 candidate pairs. The subsequent selection of the voice pitch is based on short-duration voice-harmonic frequency instability. These enhancements have been described in Chapter 5.

#### *8.1.1.2 LIWANG System*

The LIWANG system initially processes the signal using an auditory model and correlation-based periodicity analysis, following which different observation likelihoods are defined for the cases of 0, 1 and 2 (jointly estimated) F0s. A hidden Markov model (HMM) is then employed to model, both, the continuity of F0 tracks and also the jump probabilities between the state spaces of 0, 1 or 2 F0s. The 2-pitch hypothesis is introduced to deal with the interference from concurrent pitched sounds. When 2 pitches are output the first of these is

labeled as the predominant (voice) pitch. The LIWANG system has been previously shown in (Li & Wang, 2005) to be superior to those of Rynnanen & Klapuri (2006), Klapuri (2003) and Wu, Wang & Brown (2003), for detecting the pitch of the singing voice in polyphonic audio. It should be noted that, unlike the TWMDP system, the LIWANG system also makes a voicing decision.

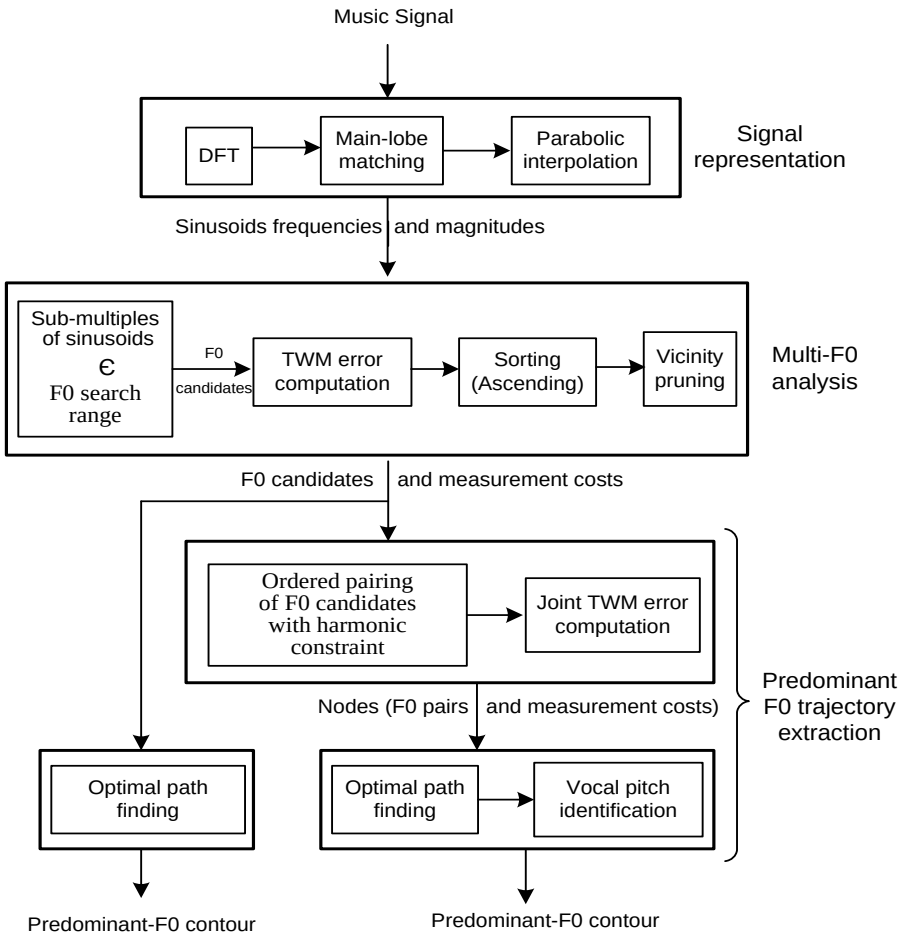


Figure 8.1: Block diagram of TWMDP system

### 8.1.2 Data Description

Three separate datasets are used for these evaluations. Each of the datasets is a subset of one of the datasets for predominant-F0 extraction described in Chapter 2. The durations of each of the three datasets used are shown in Table 8.1. Here total duration refers to the length of the entire audio, and vocal duration refers to the duration for which voiced utterances are present. All the audio clips in each of the datasets are sampled at 16 kHz with 16-bit resolution.

The first dataset (I) consists of the same audio examples as used by Li & Wang for the evaluation of their predominant F0 extraction system in (Li & Wang, 2007) and is referred to

as PF0-LiWang. The clean vocal and accompaniment tracks of these songs were extracted by the original authors from karaoke CDs using de-multiplexing software. The second dataset (II) consists of a subset (13 clips) from the MIR-1k database (PF0-MIREX09). These 13 clips have been selected based on the presence of strong pitched accompanying instruments such as acoustic guitar, piano, harmonica and accordion. This dataset includes both male and female singers. The third dataset (III) consists of a subset of the MIREX'08 dataset (PF0-MIREX08), which has examples of loud harmonium accompaniment.

As mentioned before, the ground-truth voice pitch for all datasets was computed from the clean vocal tracks using the YIN PDA (de Cheveigne & Kawahara, 2002), known to be very accurate for monophonic signals, followed by DP-based post-processing and manual correction of octave and voicing errors.

Table 8.1: Description and durations of each of the testing datasets for evaluation of the predominant-F0 trajectory extraction enhancements

| Dataset | Description | Vocal (sec) | Total (sec) |
|---|---|---|---|
| 1 | Li & Wang data | 55.4 | 97.5 |
| 2 | Examples from MIR-1k dataset with loud pitched accompaniment | 61.8 | 98.1 |
| 3 | Examples from MIREX'08 data (Indian classical music) | 91.2 | 99.2 |
| **TOTAL** | | **208.4** | **294.8** |

### 8.1.3 Experimental Setup

In the following evaluation, we first compare the performance of the TWMDP single-F0 tracking melody extraction system with the LIWANG system using the first dataset. The mixed voice and accompaniment tracks used in this experiment, at Signal-to-Accompaniment Ratios (SARs) of 10, 5, 0 & -5 dB, were obtained directly from Li & Wang. Next we compare the performance of the TWMDP single and dual-F0 tracking systems for all three datasets. The second and third datasets however are particularly representative of the kind of polyphonic scenario where the TWMDP dual-F0 tracker is expected to show significant improvement. The voice and accompaniment in these cases are mixed at SARs where both the voice melody and instrument pitch are clearly audible. This results in SARs of 10dB and 0 dB

for the second and third datasets respectively. Almost none of the clips in first dataset contain strong pitched accompaniment.

A fixed set of parameters for the TWMDP system is used for the entire experiment (shown in Table 8.2). Also, code provided to us by Li & Wang for the LIWANG system is compiled without making any modifications. In the interest of fairness the same F0 search range (80-500 Hz) as used by the LIWANG system is also used by the TWMDP system. Both systems provide a pitch estimate every 10 ms.

The multi-F0 analysis module of the TWMDP system is separately evaluated in terms of percentage presence of the ground-truth voice pitches in the F0 output candidate list. Percentage presence is defined as the percentage of voiced frames that an F0 candidate is found within 50 cents of the ground truth voice-F0.

Table 8.2: TWMDP System parameters

| **Parameter** | **Value** |
|---|---|
| Frame length | 40 ms |
| Hop | 10 ms |
| F0 search range | 80 – 500 Hz |
| Upper limit on spectral content | 5000 Hz |
| Single-F0 TWM param. ($p$, $q$, $r$ & $\rho$) | 0.5, 1.4, 0.5 & 0.1 |
| Dual-F0 TWM param. ($p$, $q$, $r$ & $\rho$) | 0.5, 1.4, 0.5 & 0.25 |
| Std. dev. of smoothness cost ($\sigma$) | 0.1 |
| Harmonic relationship threshold | 5 cents |

For the evaluation of the complete single- and dual-F0 melody extraction systems, the metrics used are pitch accuracy (PA) and chroma accuracy (CA). PA is defined as the percentage of voiced frames for which the pitch has been correctly detected i.e. within 50 cents of a ground-truth pitch. CA is the same as PA except that octave errors are forgiven. Only valid ground-truth values i.e. frames in which a voiced utterance is present, are used for evaluation. These evaluation metrics are computed with the output of the TWMDP single- and dual-F0 tracking systems as well as with the output of the LIWANG system.

For dual-F0 tracking evaluation two sets of metrics are computed. The first is a measure of whether the correct (vocal) pitch at a given instant is tracked by *at least* one of the two contours, and is called the Either-Or accuracy. This metric is an indicator of melodic recovery

by the dual-F0 tracking system. The second set of metrics is computed on the final single contour output after vocal pitch identification. Comparison between these two sets of metrics will be indicative of the reliability of the system for vocal pitch identification.

### 8.1.4 Results

Results for the evaluation of the multi-F0 extraction part of the TWMDP system for all three datasets appear in Table 8.3. For dataset I, we have used the 0 dB mix. The percentage presence of the voice-F0 is computed in the top 5 and top 10 candidates respectively, as output by the multi-F0 extraction system. It can be seen that the voice-F0 is present in the top 10 candidates about 95 % of the time thus supporting the design of the multi-F0 analysis module.

Figure 8.2 (a) & (b) compare the performance of the LIWANG system with the TWMDP single-F0 tracking system for different SAR mixes of dataset I in terms of pitch and chroma accuracy respectively. The TWMDP system is clearly superior to the LIWANG system. The relative difference in accuracies increases as the SARs worsen.

Finally, Table 8.4 compares the performance of the LIWANG, TWMDP single- and dual-F0 tracking systems. Here too we have used the 0 dB mix for dataset I. The percentage improvements of the TWMDP single- and dual-F0 tracking systems over the LIWANG system (treated as a baseline) are provided in parentheses. It should be noted that the accuracies of the LIWANG system under the ‘single-F0’ and ‘dual-F0 final’ headings are the same, since their vocal pitch identification mechanism just labels the first F0 of the two output F0s (if any) as the predominant F0. Again here we can see that the TWMDP single-F0 accuracies are significantly higher than the LIWANG accuracies. For datasets II & III, in which a strong pitched accompaniment was often present, the use of the dual-F0 approach in the TWMDP system results in further significant improvement over the single-F0 system.

Table 8.3: Percentage presence of ground-truth voice- F0 in F0 candidate list output by multi-F0 extraction module for each of the three datasets.

| **Dataset** | **Percentage presence of voice-F0 (%)** | |
|---|---|---|
| | **Top 5 Candidates** | **Top 10 Candidates** |
| 1 | 92.9 | 95.4 |
| 2 | 88.5 | 95.1 |
| 3 | 90.0 | 94.1 |

Table 8.4: Pitch accuracies (PA & CA) of TWMDP single- and dual-F0 tracking systems for all datasets. The percentage improvement over the LIWANG system is given in parentheses.

| | | **TWMDP (% improvement over LiWang)** | | |
|---|---|---|---|---|
| **Dataset** | | **Single-F0** | **Dual-F0** | |
| | | | **Either-Or** | **Final** |
| 1 | PA (%) | 88.5 (8.3) | 89.3 (0.9) | 84.1 (2.9) |
| | CA (%) | 90.2 (6.4) | 92.0 (1.1) | 88.8 (3.9) |
| 2 | PA (%) | 57.0 (24.5) | 74.2 (-6.8) | 69.1 (50.9) |
| | CA (%) | 61.1 (14.2) | 81.2 (-5.3) | 74.1 (38.5) |
| 3 | PA (%) | 66.0 (11.3) | 85.7 (30.2) | 73.9 (24.6) |
| | CA (%) | 66.5 (9.7) | 87.1 (18.0) | 76.3 (25.9) |

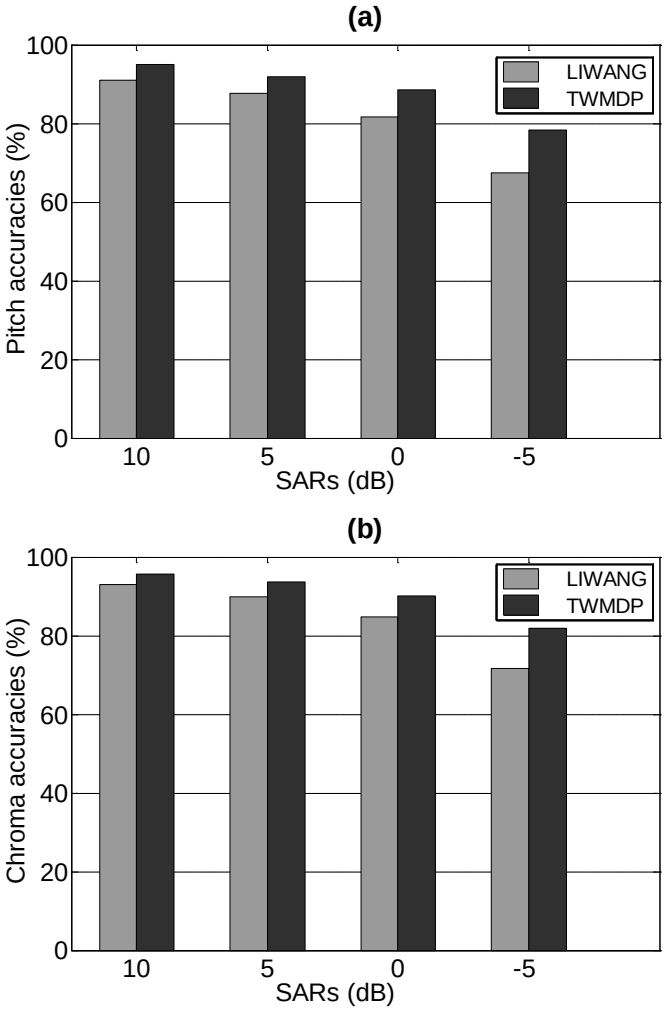


Figure 8.2: (a) Pitch and (b) Chroma accuracies for LIWANG and TWMDP Single-F0 tracking systems for Dataset 1 at SARs of 10, 5, 0 & - 5 dB.

### 8.1.5 Discussion

#### *8.1.5.1 Melodic F0 Recovery for TWMDP*

From the results in Table 8.4 it is observed that for all datasets the Either-Or pitch accuracy of the TWMDP dual-F0 tracking system is higher than that of the single-F0 system indicating that some of the melodic contour information, lost by the latter, has been recovered. Errors in the output of the single-F0 tracking system were observed when some pitched accompanying instrument in the polyphony is of comparable strength to the singing voice. At these locations the single-F0 pitch contour very often tracks the pitch of the accompanying instrument rather than the singing voice. The dual-F0 tracking approach alleviates the bias in the single-F0 system measurement cost towards such locally dominant pitched accompaniment by including another pitch trajectory in the tracking framework, which deals with the instrument F0, thereby allowing the continuous tracking of the voice-F0. The dual-F0 tracking approach also aids melodic recovery around F0 collisions between the voice-F0 and an instrument-F0 because of the faster resumption of tracking the voice-F0 around the collision by any one of the two contours in the dual-F0 system output

The Either-Or accuracy for datasets II and III is significantly higher than the single-F0 tracking accuracies but this is not the case for dataset I where the difference is much smaller. As mentioned before the presence of strong pitched accompaniment in dataset I was rare. This indicates that the dual-F0 tracking approach is particularly beneficial for music in which strong, pitched accompaniment is present but may not provide much added benefit otherwise.

An example of melodic recovery by the dual-F0 tracking approach is shown in Figure 8.3. This figure shows the ground truth voice-pitch contour (thin) along with the F0-contours output by the single-F0 (thick), in Figure 8.3.a, and dual-F0 (thick and dashed), in Figure 8.3.b, tracking systems for an excerpt of an audio clip from dataset II. The F0s are plotted in an octave scale using a reference frequency of 110 Hz. The ground truth pitch is offset vertically by –0.2 octaves for clarity. It can be seen that the single-F0 contour switches over from tracking the voice pitch to an instrument pitch (here acoustic guitar) around 6 sec. However one of the contours output by the dual-F0 tracking is able to track the voice-pitch in this region since the other contour is actively tracking the guitar pitch in this region.

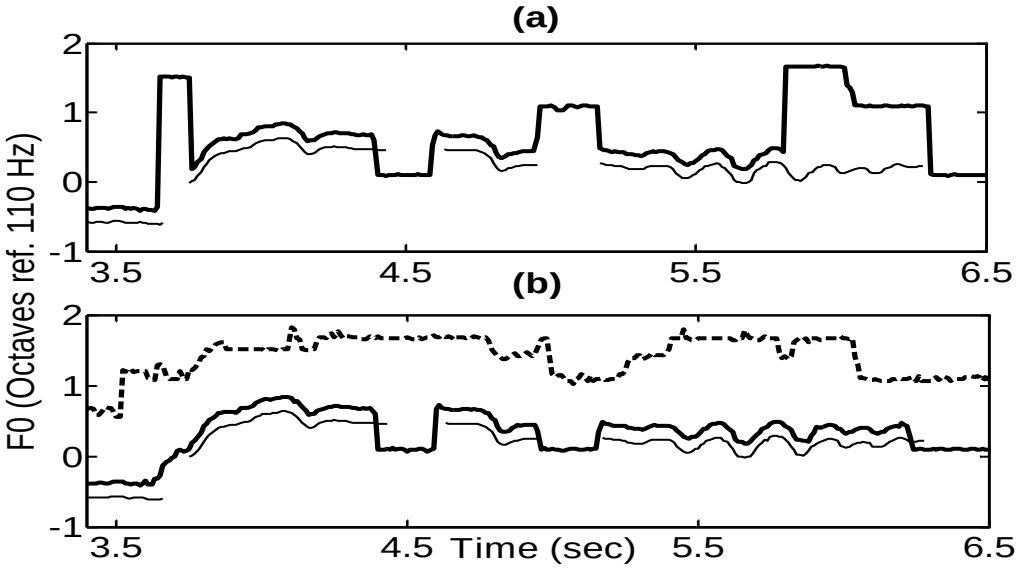


Figure 8.3: Example of melodic recovery using the dual-F0 tracking approach for an excerpt of an audio clip from dataset 2. Ground truth voice-pitch (thin) are offset vertically for clarity by –0.2 octave, (a) single-F0 output (thick) and (b) dual-F0 output (thick and dashed). Single-F0 output switches from tracking voice to instrument pitch a little before 6 sec. Dual-F0 contours track both, the voice and instrument pitch in this region.

It is possible that the simultaneous tracking of more than 2 F0s may lead to even better melodic recovery. However such an approach is not expected to result in as significant an improvement in voice-pitch tracking accuracy as the improvement resulting in the transition from single- to dual-F0 tracking. This hypothesis is based on our premise that in vocal music the voice is already the 'dominant' sound source. On occasion, an accompanying instrument may be more locally dominant than the voice however we feel that the chances that two pitched instruments are simultaneously of higher salience than the voice are relatively small.

#### *8.1.5.2 Comparison of TWMDP and LIWANG Algorithms*

From Figure 8.2 and Table 8.3 it is seen that the TWMDP algorithm consistently, and in most cases significantly, outperforms the LIWANG algorithm. The relatively lower performance of the LIWANG system could be for various reasons. One of these could be their multi-F0 extraction module, which applies a correlation-based periodicity analysis on an auditory model-based signal representation. Such multi-F0 extraction methods require that voice harmonics are dominant in at least one channel to ensure reliable voice-F0 detection, though not necessarily high salience. Previous studies indicate that such multi-F0 extraction algorithms often get confused in two-sound mixtures, especially if both sounds have several strong partials in the pass-band (Klapuri, 2008; Rao & Shandilya, 2004), and may not even

detect a weaker sound F0 (Tolonen & Karjalainen, 2000). Another cause of inaccurate pitch output of the LIWANG algorithm is the limited frequency resolution, especially at higher pitches, caused by the use of integer-valued lags.

Although the LIWANG system incorporates a 2-pitch hypothesis in its implementation (as described previously) and therefore has potential for increased robustness to pitched interference, its final performance for datasets II and III, which are representative of such accompaniment, is significantly lower than that of the TWMDP dual-F0 tracking system. This is due to multiple reasons. For dataset II the lower final accuracy of this system is due to a lack of a sophisticated vocal pitch identification stage. The Either-Or accuracies for this dataset are higher than those of the TWMDP system indicating that the voice pitch is indeed present in one of the two output pitches but is not the dominant pitch and so is not the final output. For dataset III it was observed that the LIWANG system tracks an F0 and its multiple rather than F0s from separate sources, which leads to lower Either-Or and final accuracies.

#### *8.1.5.3 Voice-pitch Identification*

The voice-pitch identification method used in the TWMDP dual-F0 tracking system does lead to increased accuracies when compared to the single-F0 tracking system. However, the final accuracies are still below the Either-Or accuracies. This indicates that some errors are being made and there is potential for further improvement in voice pitch identification. Currently we are using only a single temporal feature for voice pitch identification. We could, in the future, additionally exploit the temporal smoothness of timbral features such as MFCCs.

#### *8.1.5.4 Errors due to F0 Collisions*

Collisions between voice and instrument pitches often causes the dual-F0 tracking output contours to switch between tracking the voice and instrument pitch contours. This is explained as follows. Around the collision, one of the contours tracks a spurious F0 candidate. If this contour is the one that was previously tracking the instrument pitch then a contour that was tracking the voice pitch may now switch over to tracking the smoother instrument pitch. This will cause discontinuities in both the contours which will result in non-homogenous fragment formation during the voice-pitch identification process, which in turn degrades the voice-pitch identification performance. This is indicated by the larger differences between the Either-Or and final accuracies of the dual-F0 tracking system for dataset III, which is replete with F0 collisions, as compared to dataset II. Further, even melodic recovery may be

negatively affected since the resumption of voice-pitch tracking may be delayed after a collision.

The use of predictive models of F0 contours, similar to those used for sinusoidal modeling in polyphony (Lagrange, Marchand, & Rault, 2007), may be investigated to ensure F0 continuity of the contours output by the dual-F0 tracking system across F0 collisions. To avoid the negative effects of spurious candidate tracking at the exact F0 collision location care would have to be taken to ensure that both contours be assigned the same F0 value at that location.

### 8.1.6 Conclusions

In this experiment we have evaluated enhancements to a predominant-F0 trajectory extraction system, previously shown to be on par with state-of-the-art, with a focus on improving pitch accuracy in the presence of strong pitched accompaniment. These novel enhancements involve the harmonically constrained pairing and subsequent joint tracking of F0-Candidate pairs by the DP algorithm and the final identification of the voice pitch contour from the dual-F0 tracking output utilizing the temporal instability (in frequency) of voice harmonics.

On evaluation using music datasets with strong pitched accompaniment, it was found that the single-F0 tracking system made pitch tracking errors caused by the output pitch contour switching between tracking the voice and instrument pitches. The dual-F0 tracking approach, which dynamically tracks F0-candidate pairs generated by imposing specific harmonic relation-related constraints and then identifies the voice-pitch from these pairs, retrieves significant quantities of voice pitches for the same data. It is also shown that the performance of the proposed single- and dual-F0 tracking algorithms is significantly better than another contemporary system specifically designed for detecting the pitch of the singing voice in polyphonic music, using the same music datasets.

## 8.2 Evaluations of Enhancements to Singing Voice Detection for Loud Pitched Accompaniment[1]

In the previous chapter we saw that the use of a predominant-F0 driven energy feature outperformed multiple static timbral feature sets for the singing voice detection task on a Hindustani music dataset (SVD-Hind). However the presence of a loud pitched accompaniment will naturally degrade the performance of such an energy-feature. Although

[1] This work was done with the help of Chitralekha Gupta

pre-processing for flat-note instrument suppression is one possible solution (see Appendix A), this will not have any effect when the loud pitched accompanying instruments are also capable of continuous pitch variation similar to the voice. In this section we perform classification experiments in order to evaluate the incremental contributions, if any, of different proposed enhancements to a baseline SVD system specifically targeted at audio data that contains loud, pitched accompaniment, which may or may not be capable of continuous pitch variations.

These enhancements include the use of a predominant-F0 based isolated source spectrum using harmonic sinusoidal model (described in Section 6.2.1), a new static feature set (described in Section 6.2.2.2), the combination of static features with the different dynamic feature categories (described in Section 6.2.3) in various modes and the grouping of decision labels over automatically detected homogenous segments. To put the evaluation in perspective we compare the above performance with a baseline feature set of the first 13 MFCCs extracted from the original frame-level magnitude spectrum. As mentioned before Rocamora & Herrera (2007) had found the performance of these to be superior to several other features for SVD. A GMM classifier with 4 mixtures per class with full covariance matrices is used. Vocal/Non-vocal decision labels are generated for every 200 ms texture window. The newly proposed features of the present work are applied to the same classier framework in order to evaluate the performance improvement with respect to the baseline feature set and to derive a system based on possible feature combinations that performs best for a specific genre and across genres.

### 8.2.1 Database Description

In a previous study on cross-cultural singing voice detection, one of the categories that was badly classified, and also negatively influenced the training set effectiveness, corresponded indeed to songs with predominant melodic instruments and singing co-occurring with instruments apart from singing voice with extreme characteristics in pitch, voice quality and accentedness (Proutskova & Casey, 2009). Paralleling this observation, are studies on predominant musical instrument identification in polyphony which state that pitched instruments are particularly difficult to classify due to their sparse spectra (Fuhrmann, Haro, & Herrera, 2009). Thus our choice of evaluation datasets is guided by the known difficulty of the musical context as well as the wide availability of such a category of music cross-culturally. We consider the effective extraction and evaluation of static and dynamic features

on a dataset of vocal music drawn from Western popular, Greek Rembetiko and three distinct Indian genres: north Indian classical (Hindustani), south Indian classical (Carnatic) and popular or film music (Bollywood).

All the audio excerpts in our database contain polyphonic music with lead vocals and dominant pitched melodic accompaniment, and are in 22.05 kHz 16-bit Mono format. Vocal sections, with loud pitched accompaniment, and purely instrumental sections of songs have been selected from each of the 5 genres. The Western and Greek clips are subsets of the SVD-West and SVD-Greek datasets described in Chapter 2, previously used in (Ramona, Richard, & David, 2008) and (Markaki, Holzapfel, & Stylianou, 2008) respectively. The Bollywood, Hindustani and Carnatic datasets are sub-sets of the SVD-Bolly, PF0-ICM and SVD-Carn datasets described in Chapter 2. The total size of the database is about 65 minutes which is divided into 13 min. from each genre on average. Information pertaining to the number of songs, vocal and instrumental durations for each genre is given in Table 8.5. In a given genre a particular artist is represented by only one song.

Table 8.5: Duration information of SVD Test Audio Datasets

| **Genre** | **Number of songs** | **Vocal duration** | **Instrumental duration** | **Overall duration** |
|---|---|---|---|---|
| I. Western | 11 | 7m 19s | 7m 02s | 14m 21s |
| II. Greek | 10 | 6m 30s | 6m 29s | 12m 59s |
| III. Bollywood | 13 | 6m 10s | 6m 26s | 12m 36s |
| IV. Hindustani | 8 | 7m 10s | 5m 24s | 12m 54s |
| V. Carnatic | 12 | 6m 15s | 5m 58s | 12m 13s |
| Total | 45 | 33m 44s | 31m 19s | 65m 03s |

The selected genres contain distinctly different singing styles and instrumentation. A noticeable difference between the singing styles of the Western and non-western genres is the extensive use of pitch-modulation (other than vibrato) in the latter. Pitch modulations further show large variations across non-western genres in the nature, shape, extents, rates and frequency of use of specific pitch ornaments. Further, whereas Western, Greek and Bollywood songs use syllabic singing with meaningful lyrics, the Hindustani and Carnatic music data is dominated by melismatic singing (several notes on a single syllable in the form of continuous pitch variation). The instruments in Indian popular and Carnatic genres are

typically pitch-continuous such as the violin, saxophone, flute, *shehnai*, and *been*, whose expressiveness resembles that of the singing voice in terms of similar large and continuous pitch movements. Although there are instances of pitch-continuous instruments such as electric guitar and violin in the Western and Greek genres as well, these, and the Hindustani genre, are largely dominated by discrete-pitch instruments such as the piano and guitar, accordion and the *harmonium*. A summary of genre-specific singing voice and instrumental characteristics appears in Table 8.6.

Table 8.6: Description of genre-specific singing and instrumental characteristics

| Genre | Singing | Dominant Instrument |
|---|---|---|
| I Western | Syllabic. No large pitch modulations. Voice often softer than instrument. | Mainly flat-note (piano, guitar). Pitch range overlapping with voice. |
| II Greek | Syllabic. Replete with fast, pitch modulations. | Equal occurrence of flat-note plucked-string /accordion and of pitch-modulated violin. |
| III Bollywood | Syllabic. More pitch modulations than western but lesser than other Indian genres. | Mainly pitch-modulated wood-wind & bowed instruments. Pitches often much higher than voice. |
| IV Hindustani | Syllabic and melismatic. Varies from long, pitch-flat, vowel-only notes to large & rapid pitch modulations | Mainly flat-note harmonium (woodwind). Pitch range overlapping with voice. |
| V Carnatic | Syllabic and melismatic. Replete with fast, pitch modulations | Mainly pitch-modulated violin. F0 range generally higher than voice but has some overlap in pitch range. |

## 8.2.2 Features and Feature Selection

### *8.2.2.1 Features*

Three different sets of features are extracted. These three feature sets are subsets of the static timbral, dynamic timbral and dynamic F0-Harmonic features described in Chapter 6 respectively. The list of features under consideration in this experiment is given in Table 8.7. All features are computed from a predominant-F0 based isolated source spectral representation using the harmonic sinusoidal model of Section 6.2.1. In order to study the comparative performance of features unobscured by possible pitch detection errors, we carry out feature extraction in both fully-automatic and semi-automatic modes of predominant F0 detection for the dominant source spectrum isolation. In the latter mode, the analysis is carried

out using the semi-automatic interface, described in Chapter 9, and analysis parameters are selected considering a priori information on the pitch range of the voice in the given piece of music. In order to avoid feature extraction in frames with no singing or pitched instrument playing and silence frames we do not process those frames whose energy is lower than a threshold of 30 dB below the global maximum energy for a particular song. The values of features in such frames are interpolated from valid feature values in adjacent frames.

Table 8.7: List of features in each category. Bold indicates finally selected feature.

| **C1**<br>**Static timbral** | **C2**<br>**Dynamic timbral** | **C3**<br>**Dynamic F0-Harmonic** |
|---|---|---|
| **F0** | Δ 10 Harmonic powers | Mean & **median** of **ΔF0** |
| **10 Harmonic powers** | Δ SC & Δ SE | Mean, median & Std.Dev. of ΔHarmonics in the range [0 2 kHz] |
| **Spectral centroid (SE)** | **Std. Dev.** of **SC** for 0.5, **1** & 2 sec | Mean, median & Std.Dev. of ΔHarmonics in the range [2 5 kHz**]** |
| **Sub-band energy (SE)** | **MER** of **SC** for 0.5, 1 & **2** sec | Mean, median & Std.Dev. of ΔHarmonics 1 to 5 |
| | **Std. Dev.** of **SE** for **0.5**, 1 & 2 sec | Mean, **median** & Std.Dev. of **ΔHarmonics 6 to10** |
| | MER of SE for 0.5, 1 & 2 sec | Mean, **median** & Std.Dev. of **ΔHarmonics 1 to10** |
| | | Ratio of mean, median & Std.dev. of ΔHarmonics 1 to 5 : ΔHarmonics 6 to 10 |

. The first feature set (C1) contains the following static timbral features: F0, first 10 harmonic powers, Spectral centroid (SC) and Sub-band energy (SE). These are extracted every 10 ms. The second feature set (C2) contains the following dynamic timbral features: Δ values for each of the first 10 harmonic powers, SC and SE, and standard deviations and modulation energy ratios (MER) of the SC and SE computed over 0.5, 1 & 2 second windows. The third feature set (C3) contains the following dynamic F0-harmonic features: 1) Mean & median of ΔF0, 2) mean, median and std.dev of ΔHarmonics in the ranges [0 2 kHz] and [2 5 kHz], 3) mean, median and std. dev. of ΔHarmonics 1-5, 6-10 and 1-10 and finally, 4) the ratio of the mean, median and std. dev. of ΔHarmonics 1-5 to ΔHarmonics 6-10.

All features are brought to the time-scale of 200 ms long decision windows. The frame-level static timbral features, generated every 10 ms, are averaged over this time-scale and the timbral dynamic features, generated over larger windows: 0.5, 1 and 2 sec, are repeated within

200 ms intervals. The F0-harmonic dynamic features were generated at 200 ms non-overlapping windows in the first place and do not need to be adjusted.

#### *8.2.2.2 Feature Selection*

Feature subset selection is applied to identify a small number of highly predictive features and remove as much redundant information as possible. Reducing the dimensionality of the data allows machine learning algorithms to operate more effectively from available training data. Each of the feature sets (C1, C2 and C3) is fed to the feature selection system using information gain ratio, described in Sec. 6.2.4, to generate a ranked list for each individual genre. A feature vector comprising the top-N features common across genres was tested for SVD in a cross-validation classification experiment to select N best features. For C1 it was observed that using all the features in this category consistently maximized the intra-genre classification accuracies and so we did not discard any of these features. For C2 and C3 we observed that the top six selected features for each of the genres consistently maximized their respective classification accuracies. Features that were common across the genres were finally selected for these two feature categories. The finally selected features in each of the categories appear in bold in Table 8.7.

In the dynamic timbral feature set the Δ values of the static features are completely ignored by the feature selection algorithm in favour of the std. dev. and *MER* values of the *SC* and *SE*. The feature selection algorithm took into account the expected high degree of correlation between the same dynamic features at different time-scales and only selected at-most one time-scale for each dynamic feature. For the F0-harmonic dynamic feature set, the final selected features (C3) are the medians of ΔF0 and ΔHarmonic-tracks rather than their means or std. dev. The choice of medians was seen to be driven by the common occurrence of intra-window flat-pitched instruments note-transitions where the F0/Harmonic tracks make a discontinuous jump. In such cases, the means and standard deviations of the Δs exhibit large values as opposed to the relatively unaffected median values, which remain low.

### 8.2.3 Boundary Detection for Post-Processing

We use the framework for audio novelty detection, originally proposed by Foote (2000), as described in Section 6.4.2. Briefly, the inputs to the novelty function generator will typically be features, which show sharp, but relatively stable, changes at boundary locations. From these a similarity matrix, a 2-dimensional representation of how similar each frame is to

every other frame, is computed. The novelty function is generated by convolving the similarity matrix with a 2-d Gaussian difference kernel along the diagonal. Peaks in the novelty function above a global threshold correspond to significant changes in the audio content and are picked as potential segment boundaries. We then prune detected boundaries using a minimum segment duration criterion i.e. if two boundaries are closer than the duration threshold then the one with the lower novelty score is discarded.

For the input to the boundary detector, we consider the NHE feature. The optimal values i.e. ones that give the best trade-off between true boundaries and false alarms, of the difference kernel duration, the novelty function threshold and the minimum segment duration were empirically found to be 500 ms, 0.15 and 200 ms respectively.

### 8.2.4 Evaluation

Two types of classification experiments are performed. The first is an N-fold cross-validation experiment carried out within each genre. Since the durations of different songs within a particular genre are unequal we consider each song to be a fold so as to avoid the presence of tokens of the same song in the training and testing data to achieve a 'Leave 1 Song out' cross-validation. The other experiment is a 'Leave 1 Genre out' cross-validation designed to evaluate the robustness of different feature sets under cross-training across genres. Here we consider each genre to be a single fold for testing while the corresponding training set includes all the datasets from the remaining genres.

For each of the experiments, we first evaluate the performance of the baseline features (MFCCs), before and after applying dominant source spectrum isolation. We next evaluate the performance of the different categories of feature sets individually (C1, C2 & C3). Further we evaluate the performance of different feature set combinations: C1+C2, C1+C3 and C1+C2+C3. In each case we evaluate both combination options - the feature concatenation approach with a single classifier (A), and a linear combination of the log-likelihood outputs per class of separate classifiers for each feature set (B). In the latter case we have used a linear combination of the log-likelihood outputs per class of separate classifiers for each feature set i.e. set all classifier weights $w_n$ in Equation (6.15) to 1.

Vocal/non-vocal decision labels are generated for every 200 ms texture window. The ground-truth sung-phrase annotations for the Western and Greek genres were provided along with the datasets. The ground-truth annotations for the remaining genres were manually

marked using PRAAT (Boersma & Weenink, 2005). In all cases classifier performance is given by the percentage of decision windows that are correctly classified.

Table 8.8: % correct classification for different genres in 'leave 1 song out' cross-validation using *semi-automatic* predominant-F0 extraction. A – feature concatenation, B – classifier combination. Bold indicates best achieved in each genre.

| **Genre** | **I** | **II** | **III** | **IV** | **V** | **Total / Grouped Total** |
|---|---|---|---|---|---|---|
| Baseline | 77.2 | 66.0 | 65.6 | 82.6 | 83.2 | 74.9 / 75.8 |
| MFCCs (after source isolation) | 78.9 | 77.8 | 78.0 | 85.9 | 85.9 | 81.2 / 82.1 |
| C1 | 79.6 | 77.4 | 79.3 | 82.3 | 87.1 | 81.0 / 82.5 |
| C2 | 72.0 | 77.9 | 80.0 | 70.1 | 65.3 | 73.2 / 74.4 |
| C3 | 64.3 | 77.0 | 68.3 | 83.7 | 70.2 | 72.6 / 78.9 |
| C1+C2 (A) | 79.1 | 77.7 | 79.9 | 83.3 | 87.2 | 81.3 / 82.8 |
| C1+C2 (B) | **82.3** | 83.6 | 85.4 | 83.3 | 86.8 | 84.2 / 84.8 |
| C1+C3 (A) | 75.9 | 83.6 | 80.3 | 88.4 | 87.5 | 82.9 / 84.7 |
| C1+C3 (B) | 80.2 | 83.4 | 81.7 | **89.7** | **88.2** | 84.5 / 85.9 |
| C1+C2+C3 (A) | 75.7 | 83.1 | 80.8 | 88.3 | 88.1 | 82.9 / 84.8 |
| C1+C2+C3 (B) | 81.1 | **86.9** | **86.4** | 88.5 | 87.3 | **85.9 / 86.5** |

Table 8.9: % correct classification in 'leave 1 genre' out cross validation using *semi-automatic* predominant-F0 extraction. A – feature concatenation, B – classifier combination. Bold indicates best achieved for each genre.

| **Genre** | **I** | **II** | **III** | **IV** | **V** | **Total / Grouped Total** |
|---|---|---|---|---|---|---|
| Baseline | 71.8 | 68.0 | 65.2 | 65.6 | 54.3 | 65.2 / 68.2 |
| MFCCs (after source isolation) | **78.1** | 76.7 | 73.5 | 65.5 | 74.3 | 73.7 / 74.2 |
| C1 | 72.7 | 77.4 | 77.8 | **72.4** | 73.7 | 74.7 / 75.6 |
| C2 | 72.1 | 71.5 | 72.0 | 47.1 | 57.7 | 54.3 / 63.1 |
| C3 | 59.0 | 70.9 | 65.7 | 62.8 | 68.1 | 65.2 / 68.4 |
| C1+C2 (A) | 73.3 | 77.4 | 78.3 | 70.6 | 74.1 | 74.7 / 75.5 |
| C1+C2 (B) | 76.4 | 80.8 | 81.2 | 64.5 | 72.7 | 75.1 / 76.3 |
| C1+C3 (A) | 72.9 | 81.9 | 80.2 | 71.4 | 74.6 | 76.1 / 78.8 |
| C1+C3 (B) | 71.4 | 82.7 | 78.9 | 70.6 | **78.8** | 76.3 / 77.7 |
| C1+C2+C3 (A) | 72.8 | 82.4 | 80.7 | 66.9 | 74.6 | 76.4 / 77.6 |
| C1+C2+C3 (B) | 75.7 | **84.0** | **82.6** | 71.8 | 76.2 | **77.1 / 78.5** |

### 8.2.5 Results and Discussion

The results of the 'Leave 1 song out' and 'Leave 1 genre out' experiments are given in Table 8.8 and Table 8.9 respectively. The best overall performance for both experiments is achieved for the combination of all three feature sets (C1, C2 and C3) and is significantly (10-12%) higher than the baseline performance. For the static feature comparison it can be seen that the feature sets C1 and MFCCs after source isolation for both experiments show similar performance and are, in general, significantly superior ($p<0.05$) to the baseline features (non-source-isolated MFCCs). This is indicative of the effectiveness of the predominant-F0 based source spectrum isolation stage. We can also see that feature combination by linear combination of classifier likelihoods is, by and large, significantly superior to feature concatenation within a single classifier ($p<0.05$). In all further discussion mention of feature combination only refers to linear combination of classifier likelihoods. The clear superiority of the C1+C2+C3 feature combination over the static feature set C1 and over the baseline MFCC feature set can also be observed by the across genre average vocal precision v/s recall curves in Figure 8.4 for the 'leave 1 song out' experiment. Further, for each feature set grouping of window-level decisions over boundaries further enhances performance as can be seen by the final columns in the result tables. Although this improvement seems marginal, it is statistically significant ($p<0.05$) for both the cross-validation experiments. A detailed analysis of the genre-specific performance of each feature set follows.

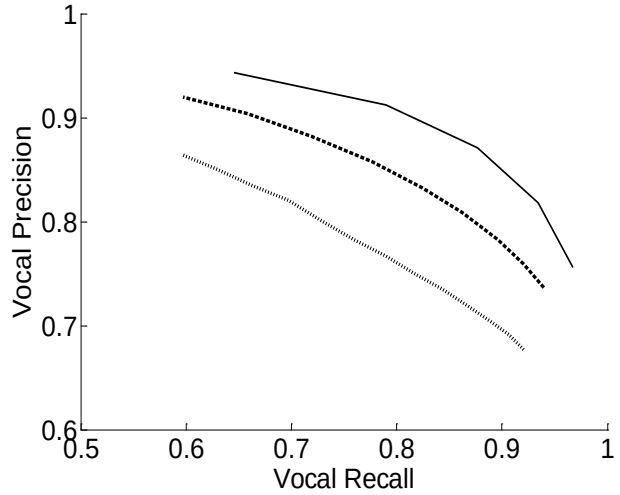


Figure 8.4: Avg. Vocal Recall v/s Precision curves for different feature sets (baseline (dotted), C1 (dashed) and C1+C2+C3 classifier combination (solid)) across genres in the 'Leave 1 song out' classification experiment.

#### *8.2.5.1 Leave 1 song out*

The feature set C2 shows relatively high performance for the Western, Greek and Bollywood genres as compared to the Hindustani and Carnatic genres. This can be attributed to the presence of normal syllabic singing in the former and long duration vowel and melismatic singing in the latter. The relatively high performance of this feature set in the Bollywood genres where the instruments are mainly pitch-continuous corroborates with the static timbral characteristics of these instruments despite their continuously changing pitch.

The feature set C3 shows relatively lower performance for the Bollywood, Carnatic and Western genres than for the Greek and Hindustani genres. The lack of contrast in the F0-harmonic dynamics between the voice and instruments for Bollywood and Carnatic, which have mainly pitch-continuous instruments, and Western, which has low voice-pitch modulation occurrences can explain this. In both Greek and Hindustani there are several clips that are replete with voice pitch modulations while the instrument pitch is relatively flat.

The combinations of C1+C2 and C1+C3 show trends similar to C2 and C3 respectively. The final combination of all three categories of feature sets shows the most superior results as compared to only C1, since, except for the Carnatic genre for which neither C2 nor C3 was able to add any value, each of the dynamic feature categories contributes positively to the specific genres for which they are individually suited.

The suitability of C2 and C3 to specific signal conditions can be understood from Figure 8.5 (a) and (b), which show spectrograms of excerpts from the Bollywood and Hindustani genres respectively. For the Bollywood excerpt the left half contains a dominant melodic instrument and right half contains vocals, and vice versa for the Hindustani excerpt. In the Bollywood case the instrument is replete with large pitch modulations but the vocal part has mainly flatter note-pitches. However the instrumental timbre is largely invariant while the vocal part contains several phonemic transitions which give a patchy look to the vocal spectrogram. In this case the timbral dynamic feature set (C2) is able to discriminate between the voice and instrument but the F0-harmonic dynamics feature set fails. The situation is reversed for the Hindustani excerpt since, although the instrumental part still displays timbral invariance, this is also exhibited by the vocal part, which consists of a long single utterance i.e. rapid pitch modulations on the held-out vowel /a/. C2 is ineffective in this case due to the absence of phonetic transitions. However the relative flatness of the instrument harmonics as compared with the vocal harmonics leads to good performance for C3.

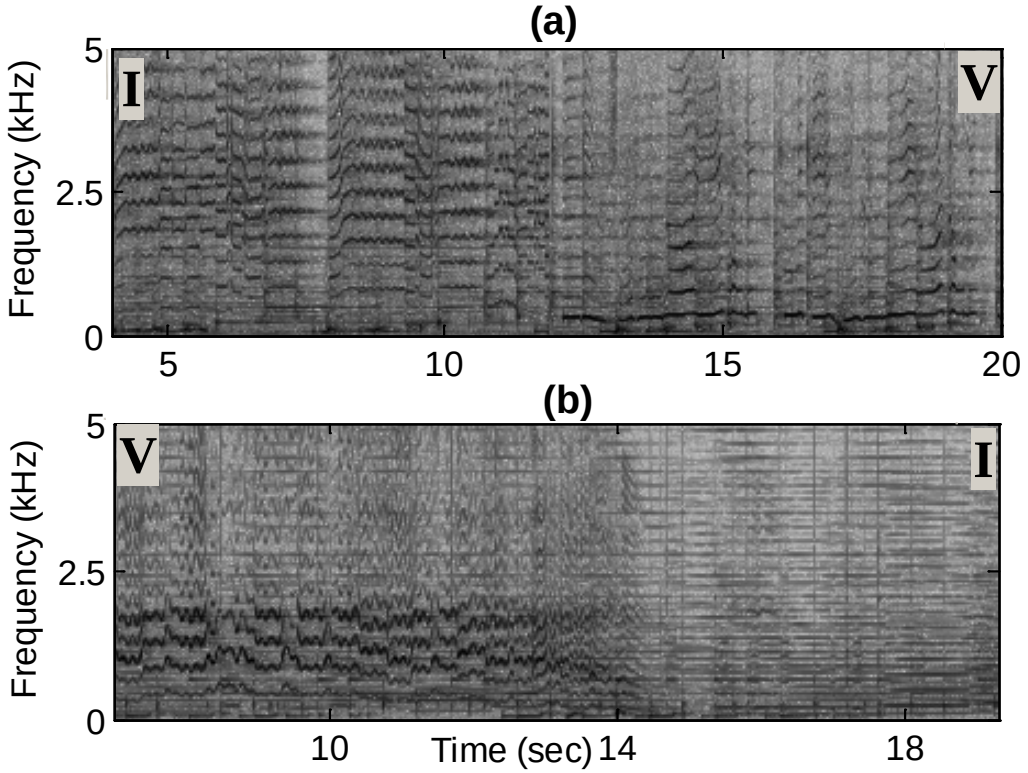


Figure 8.5: Spectrograms of excerpts from (a) Bollywood (left section instrument I and right section vocal V) and (b) Hindustani (left section vocal V and right section instrument I) genres.

#### *8.2.5.2 Leave 1 Genre out*

As compared to its individual performance in the previous experiment the performance of C1, for all genres except Greek, exhibits a drop for the 'Leave 1 genre out' experiment. This indicates that there was some genre-specific training that aided enhanced the vocal/non-vocal discrimination. The invariance of the performance of C1 for the Greek genre across experiments can be attributed to the presence of the family of instruments (bowed, woodwind, plucked-string) used in Greek music in at least one of the other genres.

For C2 and C3 we observe their performance, individually and in combination with C1, only for the genres in which they performed well in the previous experiment. We can observe that similar trends occur in the present experiment except the case in which the addition of C3 to C1 results in a drop in performance for Hindustani. This is in contrast to the positive effect of C3 for the same genre in the previous experiment. On further investigation we found that this drop was caused by a drop in vocal recall due to misclassification of extremely flat long sung notes, mostly absent in other genres and hence in the training set.

#### *8.2.5.3 Comparison with fully-automatic predominant-F0 extraction*

Although the main objective of these experiments is a comparison of the cases where various features are used, predominantF0 extraction is a part of primal difficulty for the F0-dependent approaches, and should not be totally ignored when the accuracy of these feature sets is discussed and compared to the performance of a baseline, non-F0 dependant feature set. So we compute the results of the 'leave 1 song out' cross-validation using fully-automatic predominant F0 extraction based source spectrum isolation for baseline and individual feature sets and for the linear combination of classifiers only, since we previously showed that linear combination of classifier likelihoods is, by and large, superior to feature concatenation within a single classifier.

Table 8.10: % correct classification for different genres in 'leave 1 song out' cross-validation using *fully-automatic* predominant-F0 extraction for individual feature sets and classifier combinations. Bold indicates best achieved in each genre.

| **Genre** | **I** | **II** | **III** | **IV** | **V** | **Total / Grouped Total** |
|---|---|---|---|---|---|---|
| Baseline | 77.2 | 66.0 | 65.6 | 82.6 | 83.2 | 74.9 / 75.8 |
| MFCCs (after source isolation) | 76.9 | 72.3 | 70.0 | 78.9 | 83.0 | 76.2 / 76.4 |
| C1 | 81.1 | 67.8 | 74.8 | 78.9 | 84.5 | 77.4 / 77.9 |
| C2 | 72.0 | 75.9 | 77.1 | 67.5 | 65.8 | 71.7 / 72.2 |
| C3 | 65.6 | 69.1 | 62.1 | 77.1 | 66.0 | 68.0 / 74.6 |
| C1+C2 | **82.9** | 78.5 | 82.8 | 79.5 | **85.0** | 81.7 / 83.1 |
| C1+C3 | 81.5 | 72.9 | 77.6 | **83.9** | 84.9 | 80.2 / 81.5 |
| C1+C2+C3 | 82.1 | **81.1** | **83.5** | 83.0 | 84.7 | **82.8 / 84.2** |

We use the system submitted to the MIREX evaluations (described in Section 7.2.1) as a fully automatic predominant F0 extractor with voicing detection disabled, since we are require a predominant-F0 estimate for every analysis time-instant. While all the other analysis parameters are the same as before, we had to increase the upper limit on F0-search range to 1500 Hz, since the F0s some of the instruments in the Greek and Bollywood genres sometimes reached these values.

The results of the above evaluation are shown in Table 8.10. The fully-automatic F0 extraction based results shows reduced performance for different feature sets and their combinations as compared to the corresponding semi-automatic F0 extraction based results of

Table 8.8. This is due to the pitch detection errors inherent in the fully-automatic F0 extraction system. However the general trends across different feature sets hold for this case as well, with the results of Table 8.8 providing an upper limit on performance achievable with a better pitch detector. Additionally here we see that the performance of the MFCCs (after source isolation) is significantly ($p<0.05$) better than the baseline and the performance of C1 is significantly ($p<0.05$) better than that of the MFCCs (after source isolation).

### 8.2.6 Conclusions

In this section we have evaluated the use of a combination of static and dynamic features for effective detection of lead vocal segments within polyphonic music in a cross-cultural context. Several of the features are novel and have been motivated by considering the distinctive characteristics of singing voice across genres. The polyphonic scenario we have chosen to focus upon is one in which a dominant melodic instrument other than the voice exists simultaneously, which is known to be one of the harder signal conditions for this problem. The introduction of an isolated dominant source spectral representation (harmonic sinusoidal model) resulted in a significant increase in the performance of static timbral features over a popularly used baseline feature set (13 MFCC coefficients). The dynamic features have been segregated into two categories – timbral dynamics and F0-harmonic dynamics. These two categories were found to provide complementary information for different underlying signal conditions related to singing styles and instrumentation specific to individual genres. While the overall combination of the static and dynamic features was found to result in the highest overall classification performance, individual genre accuracies clearly indicate the value of adapting feature sets to genre-specific acoustic characteristics. Thus commonly available metadata, such as genre, may be effectively utilized in the front-end of an MIR system. Further it was found that using a linear combination of log likelihoods of individual classifiers per feature set category resulted in higher averaged cross-genre performance as compared to concatenating all the features within a single classifier. The combination weights can possibly be tuned for further gains in performance. Finally, a grouping algorithm using majority vote across automatically extracted sung-phrase boundaries was found to enhance performance marginally, but consistently, for all feature sets across genres.

# Chapter 9

## Interactive Tool for Melody Extraction

Melody extraction from polyphony is still a young research problem. Although there exists a considerable body of work in pitch extraction from monophonic (single-source) audio, advances in research that enable melodic pitch extraction from polyphonic audio (a harder problem because of increased signal complexity due to polyphony) have only recently been made (in the last decade).

As mentioned before the melody extraction problem comprises of two sub-problems viz. predominant-F0 contour extraction and vocal (sung) segment detection. Both these sub-problems continue to be far from solved for use in practical automatic music transcription systems for operation on large datasets of polyphonic music across various genres and styles. However the applicability of available algorithms can be extended considerably by employing semi-automatic approaches tailored for specific applications (Wang & Zhang, 2008).

This chapter describes a graphical user interface (GUI) for semi-automatic melody extraction from polyphonic music[1]. There were several motivating factors behind the design of the proposed GUI. Primarily we needed to subjectively evaluate the performance of our melody extraction system for different polyphonic signals using different parameters since we had found that for signals of varying characteristics, it was possible to obtain accurate melodic contours by intelligently varying a few of the system parameters. With this possibility in mind, such an interface also has practical utility to the MIR community. For example, melody based reference templates required for the searchable database in query-by-humming systems must be extracted from polyphonic soundtracks. Another example where high accuracy voice-pitch tracks from available commercial recordings of classical music performances can be useful is for musicological analyses as well as for music tutoring applications.

The final objective in the design of the interface is to facilitate the extraction and validation of the voice pitch contour from polyphonic music with minimal human intervention. Since the manual marking of vocal segment (sung phrase) boundaries is much easier than automatic detection of the frame-by-frame voice pitch, the focus on the design of the back-end melody extraction program has been on automatic and high accuracy vocal pitch extraction.

In our interface we have included some features from some previously available related interfaces and have also added some novel features, which we feel will further ease the melody extraction process. We next describe some of the previous interfaces available for similar tasks. Following that we describe the design layout and operation of our melody extraction interface. Section 9.3 brings out the salient features of interface with respect to facilitating melody extraction and refinement. Section 9.4 lists the development details. Section 9.5 presents a summary and some future enhancements to be made to the interface.

## 9.1 Available GUIs for Pitch Processing of Audio

Publicly available pitch detection interfaces, such as PRAAT (Boersma & Weenink, 2005) and the Aubio pitch detector VAMP plugin (Brossier P. ) for the Sonic Visualizer program (Cannam), have been designed for monophonic audio and cannot be expected to perform acceptably well on polyphonic audio. However, a few polyphonic transcription tools also exist. The polyphonic transcription VAMP plugin (Zhou) has been designed exclusively for guitar and piano music and outputs MIDI notes. Melodyne's Direct Note Access (DNA)

[1] This work was done together with Sachin Pant

(Celemony) attempts to isolate simultaneously played musical notes but is not freely available for evaluation. Neither of these programs attempts to extract a single, high-resolution melodic contour (i.e. pitch track of the lead voice) from polyphonic audio.

To best of our knowledge, the only interface specifically designed to facilitate melodic pitch extraction from polyphonic audio is MiruSinger (Nakano, Goto, & Hiraga, 2007). MiruSinger was primarily developed for improving singing skills by comparison between the user's sung pitch contour and the melody extracted from original CD polyphonic recordings. It uses another contemporary melody extraction algorithm as a back-end (Goto, 2004). The MiruSinger melody extraction interface offers several helpful features such as re-synthesis of the extracted melodic contour for validation, user correction of detected vocal segments and user correction of the extracted melodic pitch by choosing from different local salient F0 candidates. However there is no flexibility in terms of parametric control of the back-end executable.

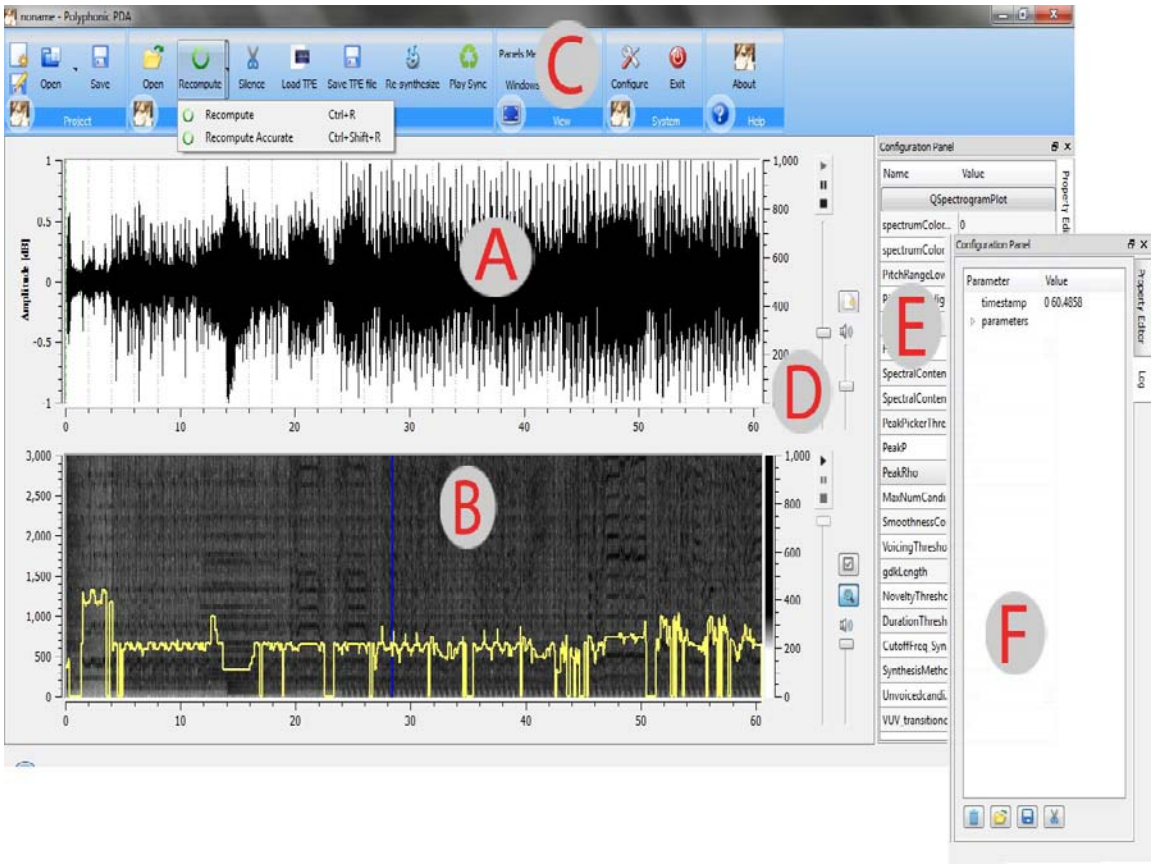


Figure 9.1: Snapshot of melody extraction interface

## 9.2 Interface: Description and Operation

The basic features that are expected from any interface intended for audio analysis/editing comprise of waveform and spectrogram displays, selection and zooming features, and audio playback. In this section we describe the layout of the interface that, in addition to these basic features, also has features designed for the melody extraction task. The operation of the interface is also described.

### 9.2.1 Description

A snapshot of the interface is provided in Figure 9.1. It consists of a waveform viewer (A), a spectrogram viewer (B), a menu bar (C), controls for audio viewing, scrolling and playback (D), a parameter window (E), a log viewer (F). The waveform and spectrogram of the audio are displayed in A & B respectively. The horizontal axis corresponds to the timeline, with data moving from left to right. The right vertical axis in B corresponds to F0 frequency. The vertical bar at the center is the "present time". Controls (D) are provided for playing back the audio, controlling the volume, and the progress of the audio. Moving markers provide timing information. The menu bar (C) and the parameter window (E) control the use of the melody extractor. A log viewer (F) is provided to save and load the analysis parameters for the different segments.

### 9.2.2 Audio Analysis

The audio example to be analyzed, in .wav format, is loaded into the interface. The waveform and spectrogram of the music are automatically displayed. The melody extractor is invoked by pressing the  button in the menu. By default, the melody extractor is called in single-F0 tracking mode, which is found to perform quite accurately on most audio examples. Alternatively the user may also select to use a more accurate, but slower, melody extraction algorithm (See Section 9.3.6) by checking the dual-F0 option in the drop-down menu under the  button. This function is especially useful when an accompanying pitched instrument is of comparable, or greater, loudness than the voice. The resulting pitch contour is displayed as a yellow curve in B. The estimated F0 contour is plotted on top of the spectrogram, which helps visually validate the estimated melody by observing the shape and/or the extent of overlap between the pitch contour and any of the voice harmonics. Voice harmonics are typically characterized by their jittery/unsteady nature. Audio feedback is also provided by

pressing the ► button on the right of the spectrogram. This plays back a natural re-synthesis of the estimated F0 contour. The extracted pitch contour of the entire audio clip can be synthesized using the button from the menu (C).

### 9.2.3 Saving and Loading Sessions

The interface provides an option of saving the final complete melody as well as the parameters used for computing melody for different selected regions by using the button. A user can save the pitch extracted in a specific file format (TPE), which has three columns containing the Time stamp (in sec), the Pitch (in Hz), and the frame-level signal Energy respectively. This amounts to saving a session. This TPE file can be loaded later for further analysis and processing. Also, the parameters of the melody extractor used during the analysis can be saved in an XML file.

## 9.3 Interface: Salient Features

As mentioned before we have attempted to incorporate some features from the MiruSinger melody extraction interface, with further enhancements, and also have incorporated some new features that increase functionality of the interface. The salient features of our melody extraction interface are described below.

### 9.3.1 Novel Melody Extractor

The melody extraction back-end system used by our interface is one of the outcomes of the investigations in this thesis. It has been extensively evaluated on polyphonic vocal music and has demonstrated very accurate voice pitch extraction performance. This is the same melody extraction algorithm that we have submitted to the MIREX 2008 and 2009 evaluations, and is described in Section 7.2.1. The system utilizes a spectral harmonic-matching pitch detection algorithm (PDA) followed by a computationally-efficient, optimal-path finding technique that tracks the melody within musically-related melodic smoothness constraints. An independent vocal segment detection system then identifies audio segments in which the melodic line is active/silent by the use of a melodic pitch-based energy feature.

Further our melody extraction system uses non-training-based algorithmic modules i.e. is completely parametric, unlike those that incorporate pattern classification or machine learning techniques (Fujihara, Kitahara, Goto, Komatani, Ogata, & Okuno, 2006; Poliner & Ellis, 2005). The performance of such systems is highly dependent on the diversity and

characteristics of the training data available. In polyphonic music the range of accompanying instruments and playing (particularly singing) styles across genres are far too varied for such techniques to be generally applicable. When using our interface users, with a little experience and training, can easily develop an intuitive feel for parameter selections that result in accurate voice-pitch contours.

### 9.3.2 Validation

The user can validate the extracted melodic contour by a combination of audio (re-synthesis of extracted pitch) and visual (spectrogram) feedback. We have found that by-and-large the audio feedback is sufficient for melody validation except in the case of rapid pitch modulations, where matching the extracted pitch trajectory with that of a clearly visible harmonic in the spectrogram serves as a more reliable validation mechanism.

Currently there are two options for re-synthesis of the extracted voice-pitch contour. The default option is for a natural synthesis of the pitch contour. This utilizes the harmonic amplitudes as detected from the polyphonic audio resulting in an almost monophonic playback of the captured pitch source. This type of re-synthesis captures the phonemic content of the underlying singing voice that serves as an additional cue for validation of the extracted pitch. However, in the case of low-energy voiced utterances especially in the presence of rich polyphonic orchestration it was found that harmonics from other instruments also get synthesized, which may confuse the user.

In such cases, an alternate option of complex-tone synthesis with equal amplitude harmonics also exists. Here the user will have to use only the pitch of the audio feedback for validation since the nature of the complex tone is nothing like the singing voice. In complex tone synthesis the frame-level signal energy may be used but we have found that this leads to audible bursts especially if the audio has a lot of percussion. Alternatively we have also provided a constant-energy synthesis option which allows the user to focus on purely the pitch content of the synthesis and not be distracted by sudden changes in energy. This option can be selected from the parameter list (E). An additional feature that comes in handy during melodic contour validation is the simultaneous, time-synchronized playback of the original recording and the synthesized output. This can be initiated by clicking the ♻ button on the menu (C). A separate volume control is provided for the original audio and synthesized playback. By controlling these volumes separately, we found that users were able to make better judgments on the accuracy of the extracted voice-pitch.

### 9.3.3 Inter-Segment Parameter Variation

Typical parameters that affect the performance of our melody extraction system are the F0 search range, analysis frame-length, lower-octave bias (ρ of the TWM PDA) and melodic smoothness tolerance (σ of the DP smoothness cost). An intelligent user will be able to tune these parameters, by observing certain signal characteristics, to obtain a correct output melody. For example, in the case of male singers, who usually have lower pitch than females, lowering the F0 search range and increasing the window-length and lower-octave bias results in an accurate output. In the case of large and rapid pitch modulations, increasing the melodic smoothness tolerance is advisable.

It may sometimes be possible to get accurate voice-pitch contours by using a fixed-set of analysis parameters for the whole audio file. But many cases were observed, especially of male-female duet songs and excerpts containing variations in rates of pitch modulation, where the same parameter settings did not result in an accurate pitch contour for the whole file. In order to alleviate such a problem the interface allows different parameters to be used for different segments of audio. This allows for easy manipulation of parameters to obtain a more accurate F0 contour. The parameter window (E) provides a facility to vary the parameters used during analysis. Here we also provide different pre-sets of parameters that have been previously shown to result in optimal predominant-F0 extraction performance for polyphonic audio with male or female singers respectively. This evaluation of the predominant-F0 extraction performance for different gender singers using different parameter presets was presented in Sec. 7.1.1.4.

To emphasize the use of this feature i.e. parameter variation for different audio segments, we consider the analysis of a specific excerpt from a duet Hindi film song. The audio clip has two phrases, the first sung by a male singer and the latter by a female singer. Figure 9.2 shows a snapshot of our interface when a single set of parameters is used on the entire audio clip. It can be seen that the pitch has been correctly estimated for the part (first half) sung by the male singer, but there are errors for the female part (second half). This becomes more evident by observing the spectrogram display closely or also by listening to the re-synthesis of the extracted pitch contour. Selecting the female portion from the song and computing its pitch using a slightly modified set of parameters (reduced analysis frame-length and lower octave bias) leads to much better estimate of female voice-pitch contour (as shown in Figure 9.3).

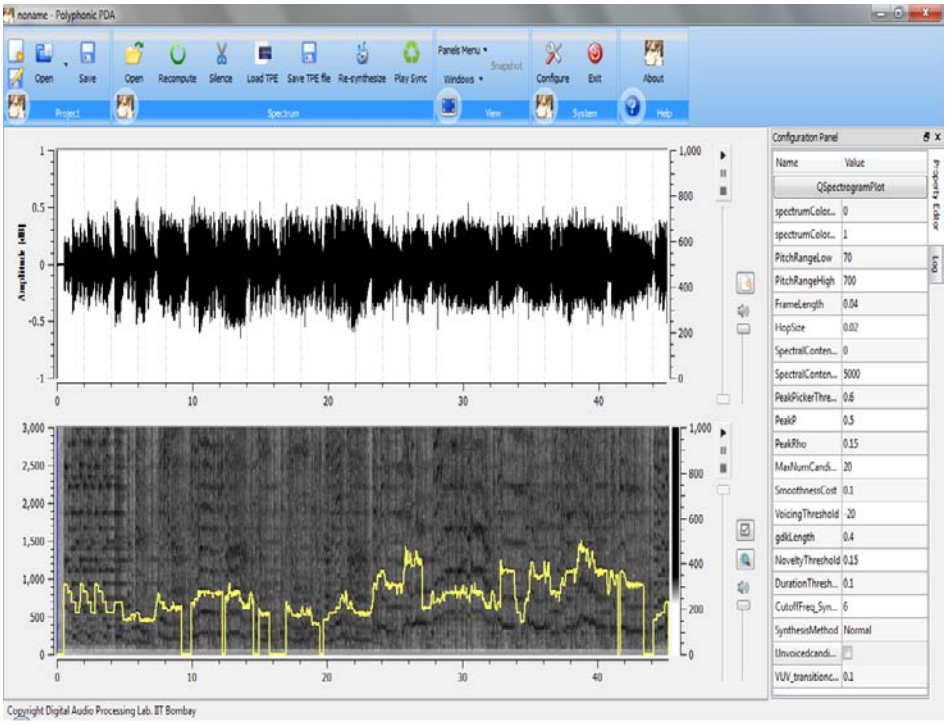


Figure 9.2: Analysis of Hindi film duet song clip showing incorrect pitch computation i.e. octave errors, in the downward direction, in the extracted pitch contour (yellow) are visible towards the second half (female part)

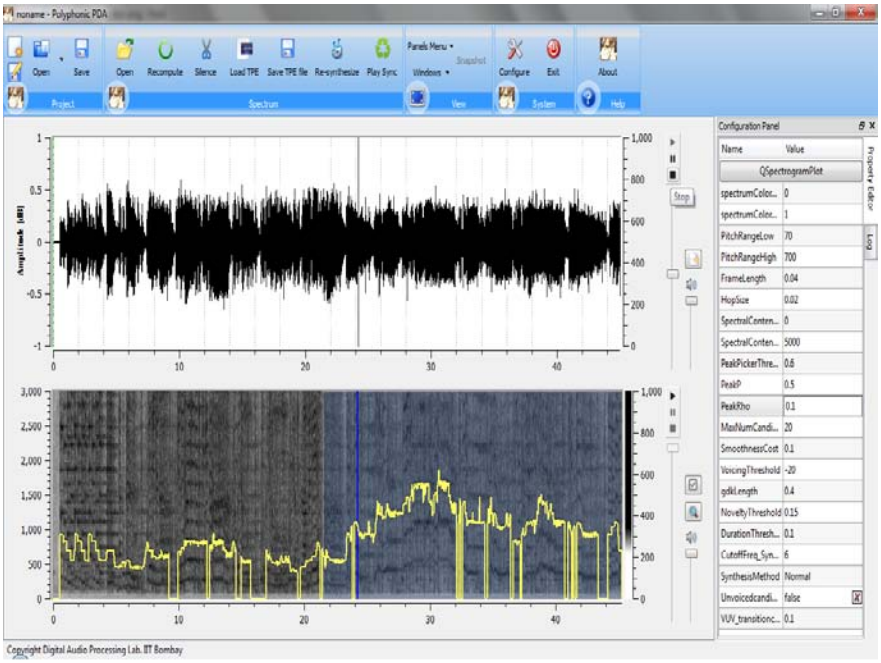


Figure 9.3: Analysis of Hindi film duet song clip showing correct pitch computation. The pitch contour (yellow) of the selected segment was recomputed after modifying some parameters.

### 9.3.4 Non-Vocal Labeling

Even after processing, there may be regions in the audio which do not contain any vocal segments but for which melody has been computed. This occurs when an accompanying, pitched instrument has comparable strength as the voice because the vocal segment detection algorithm is not very robust to such accompaniment. In order to correct such errors we have provided a user-friendly method to zero-out the pitch contour in a non-vocal segment by using the tool from the menu (C).

### 9.3.5 Saving Final Melody and Parameters

The melody computed can be saved and later used for comparison or any MIR tasks. The log component (F) records parameters used for analysis with time-stamps representing the selected regions. By studying these log files for different audio clips we gain insight into optimal parameter settings for different signal conditions. For example, one observation made was that larger analysis frame-lengths resulted in more accurate pitch contours for audio examples in which a lot of instrumentation (polyphony) was present but was found to be detrimental to performance when rapid pitch modulations were present. This motivated us to investigate the use of a signal-driven adaptive time-frequency representation (Chapter 3).

### 9.3.6 Error Correction by Selective Use of Dual-F0 Back-end

State-of-the-art melody extraction algorithms have been known to incorrectly detect the pitches of loud, pitched accompanying instruments as the final melody, in spite of the voice being simultaneously present. In Chapters 5 and 8, however, we have shown that attempting to track two, instead of a single, pitch contours can result in a significant improvement in system performance. Specifically, the path finding technique in the melody extraction algorithm is modified to track the path of an ordered pair of possible pitches through time. Pairing of pitches is done under harmonic constraints i.e. two pitches that are integer (sub) multiples of each other cannot be paired.

The use of the above ‘dual-F0 tracking’ approach presently results in a considerable increase in computation time and may not be practically viable for long audio segments. However, we have provided the option for the user to selectively apply such an analysis approach i.e. track 2 F0s. On selecting this option (by selecting the dual-F0 option in the drop-down menu under the button) the system will output 2 possible, melodic contours.

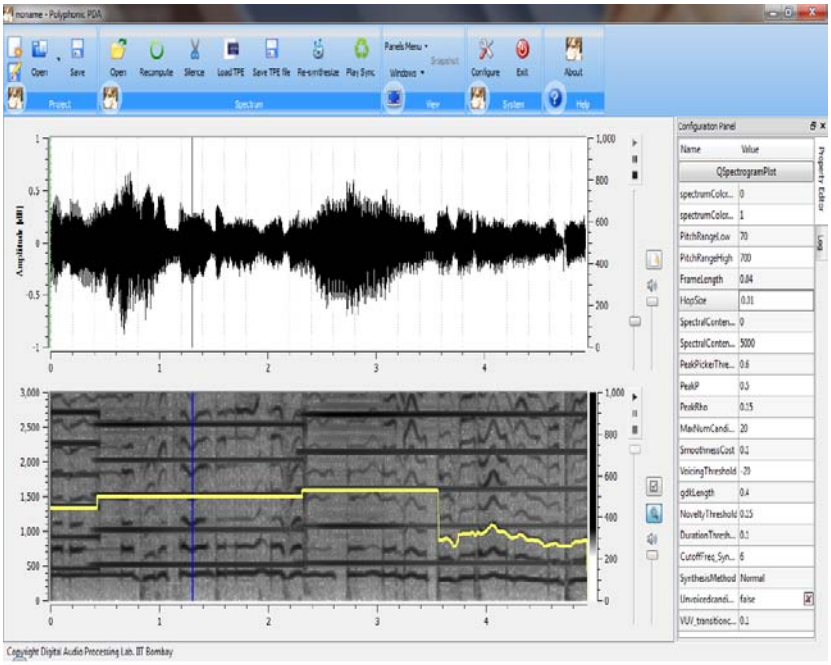

Figure 9.4: Analysis of an audio clip containing voice and loud harmonium using the single-F0 option. The extracted pitch contour (yellow) mainly tracks the harmonium pitch and only switches to the voice pitch towards the end of the clip.

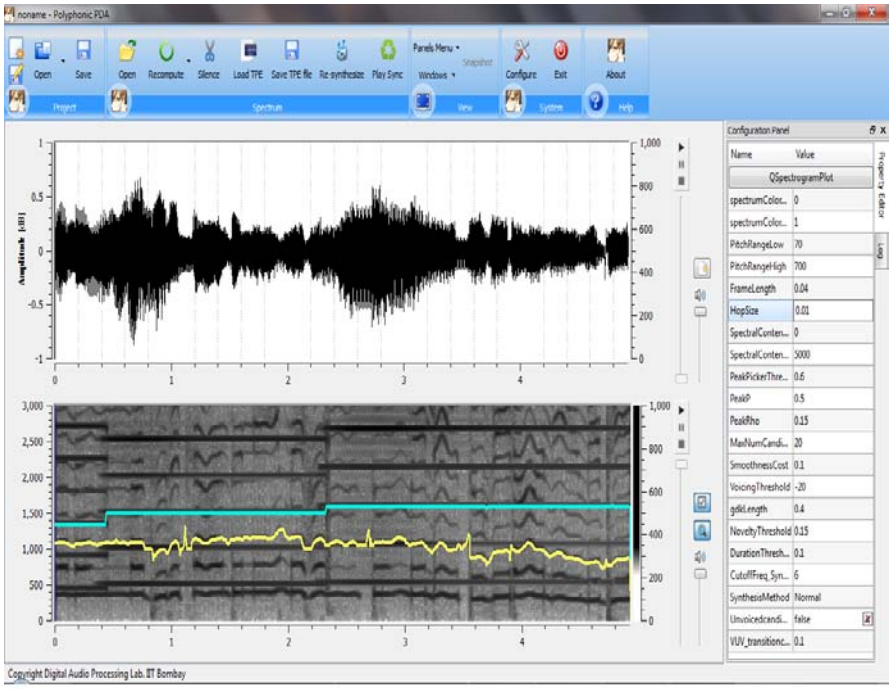

Figure 9.5: Analysis of an audio clip containing voice and loud harmonium using the dual-F0 option. The system outputs two pitch contours (yellow and blue). The yellow contour in this case represents the voice pitch.

This is much cleaner than presenting the user with multiple locally-salient F0 candidates, as this will clutter up the visual display. The user can listen to the re-synthesis of each of these contours and select any one of them as the final melody. Typically we expect users to use this option on segments on which the single-F0 melody extractor always outputs some instrument pitch contour despite trying various parameter settings.

To emphasize the performance improvement on using the dual-F0 tracking option we consider the analysis of an audio clip in which the voice is accompanied by a loud harmonium. Figure 9.4 displays the result of using the melody extractor in single-F0 mode. The resulting pitch contour can be seen to track the harmonium, not the voice, pitch for a major portion of the file. It is not possible to make any parameter changes in order to correct the output. By using the dual-F0 tracking option (Figure 9.5) we can see that now two contours are output: the yellow represents the voice pitch and the light blue represents the harmonium pitch. The user can now select one of these two contours as the final output melody.

## 9.4 Development Details

The graphical interface has been developed using Qt (Nokia) and Qwt (SourceForge) toolkit. The interface is written in C++. Qt because of its cross-compilation capabilities enables deployment of our system on a variety of Platforms. The interface uses generic component framework (GCF) (VCreateLogic) developed by VCreateLogic, which provides component based architecture making development and deployment easier.

## 9.5 Summary and Future Work

### 9.5.1 Summary

In this chapter we have presented a graphical user interface for semi-automatic melody extraction based on a recently proposed algorithm for voice-pitch extraction from polyphonic music. This interface is shown to display several novel features that facilitate the easy extraction of melodic contours from polyphonic music with minimal human intervention. It has been effectively used for melody extraction from large durations of Indian classical music to facilitate studies on *Raga* identification (Belle, Rao, & Joshi, 2009) and also on Hindi film music for extraction of reference templates to be used in a query-by-humming system (Raju,

Sundaram, & Rao, 2003). We are working towards making this interface available to fellow researchers who are interested in analyses of polyphonic music signals.

## 9.5.2 Future Work

### *9.5.2.1 Spectrogram Display*

The spectrogram display (B) is useful for validation of the extracted melodic contour, as described in Section 9.3.2. However, the 'melodic range spectrogram' (MRS) as used in the 'Sonic Visualiser' program (Cannam) would be much more appropriate. The MRS is the same as the spectrogram except for different parameter settings. It only displays output in the range from 40 Hz to 1.5 kHz (5.5 octaves), which most usually contains the melodic content. The window sizes are larger with heavy overlap for better frequency resolution. The vertical frequency scale is logarithmic i.e. linear in perceived musical pitch. Finally the color scale is linear making noise and low-level content invisible but making it easier to identify salient musical entities. The integration of the MRS into our interface will be taken up in the future.

### *9.5.2.2 Signal-driven Window-length Adaptation*

In Section 9.3.3 we have mentioned that different parameter presets may be selected for different underlying signal characteristics. The analysis frame length is one parameter that is often changed to obtain more accurate pitch tracks. In Chapter 3 we demonstrated that the use of a sparsity-driven frame-length adaptation in a multi-resolution analysis improved frame-level sinusoid detection performance. However this framework was not integrated into our final melody extraction system. Here we present a preliminary experiment that indicates that the using a signal-driven adaptive frame-length framework may alleviate the need for selection of explicit presets for the analysis frame-length parameter.

We consider one excerpt each, of 30 sec duration, from the beginning and end of a male and female North Indian vocal performance from the SVD-Hind dataset (16-bit, Mono, sampled at 22.05 kHz). The beginning and end segments contain more stable pitches and rapid pitch modulations respectively. For each excerpt we compute the pitch and chroma accuracies (PA and CA) with respect to known ground-truth pitches for fixed frame-lengths of 20, 30 and 40 ms. We then compute the PA and CA for a kurtosis-driven adaptive frame-length scheme in which, at each analysis time instant (spaced 10 ms apart), that frame-length, out of 20, 30 and 40 ms, is selected that maximizes the value of the normalized kurtosis (Section 3.1.2.1). The normalized kurtosis is computed from the 2.5 to 4 kHz region of the

Table 9.1: Performance (pitch accuracy (PA %), chroma accuracy (CA %)) of the different fixed (20, 30 and 40 ms) and adaptive frame-lengths for excerpts from the beginning (slow) and end (fast) of a male and female North Indian vocal performance. WIN (%) is the percentage of the time a given frame-length was selected in the adaptive scheme.

| **Excepts** | **20 ms window** | | | **30 ms window** | | | **40 ms window** | | | **Adaptive** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **PA** | **CA** | **WIN** | **PA** | **CA** | **WIN** | **PA** | **CA** | **WIN** | **PA** | **CA** |
| Female slow | 97.4 | 97.5 | 0.4 | 99.5 | 99.5 | 1.6 | 96.8 | 97.3 | 98.0 | 97.5 | 98.1 |
| Female fast | 97.7 | 97.7 | 34.9 | 91.3 | 92.5 | 15.1 | 81.4 | 83.2 | 50.0 | 96.1 | 96.3 |
| Male slow | 47.4 | 85.4 | 2.1 | 90.1 | 92.3 | 1.2 | 97.1 | 97.1 | 96.7 | 96.2 | 96.8 |
| Male fast | 78.3 | 81.5 | 18.6 | 90.0 | 91.3 | 23.2 | 85.5 | 85.7 | 58.1 | 88.2 | 88.5 |
| **Average** | **80.2** | **90.5** | **-** | **92.7** | **93.9** | **-** | **90.2** | **90.8** | **-** | **94.5** | **94.9** |

magnitude spectrum which, for all frame-lengths, is computed using a 2048 point zero-padded DFT. We also note the % of the time (WIN) each of the different frame-lengths is selected for each audio clip.

The results of the above experiment are given in Table 9.1. The results indicate that the adaptive frame-length scheme leads to the best performance on average. However for individual files different fixed frame-lengths perform best. For the slow female excerpt the selection of frame-length is not very critical since the voice harmonics are well resolved for all frame-lengths, because of higher pitch. This is in contrast to the results for the slow male excerpt, with lower pitch, for which the best results are obtained for the 40 ms frame-length, and these are significantly better than that for the 20 ms frame-length. For the fast female and fast male excerpts the best performance was obtained for the fixed 20 and 30 ms frame-lengths respectively. In case of the latter it appears that the 30 ms frame-length is a trade-off between reducing intra-frame signal non-stationarity and resolving low-pitch voice harmonics. It also appears that the percentage selection of each window in the adaptive scheme (WIN) may be used as an indicator of which fixed frame-length is suitable for each excerpt. The distribution of this measure is more spread for the fast excerpts and skewed towards the 40 ms frame-length for the slow excerpts.

This preliminary experiment indicates that a measure of sparsity may be used as an indicator of which preset will lead to favorable results, at least in the case of extreme cases of signal non-stationarity i.e. stable and rapidly modulated voice pitch. Further experimentation using a multi-resolution adaptive representation will be taken up in the future.

# Chapter 10

# Conclusions and Future Work

## 10.1 Summary and Conclusions

The main aim of this thesis was to investigate the problem of melody extraction from polyphonic music in which the lead instrument is the human singing voice. Despite a significant volume of research in this area worldwide in the last decade, a practically applicable, general-purpose melody extraction system is not presently available. In a review of the related literature it was found that the presence of competing pitched accompaniment was often mentioned as one of the causes of errors in contemporary melody extraction algorithms. Since the use of pitched accompaniment is particularly pervasive in Indian music, as also in other non-Western music traditions such as Greek Rembetiko, this problem was chosen as the focus of the research leading to this thesis. We propose a melody extraction system that specifically addresses the pitched accompaniment problem and demonstrates robustness to such accompaniment. An intermediate version of our system was submitted to the MIREX Audio Melody Extraction task evaluations in 2008 and 2009 and was shown to be

on-par with state-of-the-art melody extraction systems. The block diagram of the final system developed as a result of this thesis is shown in Figure 10.1. A summary of the main results of the investigations for each sub-module of the system follows.

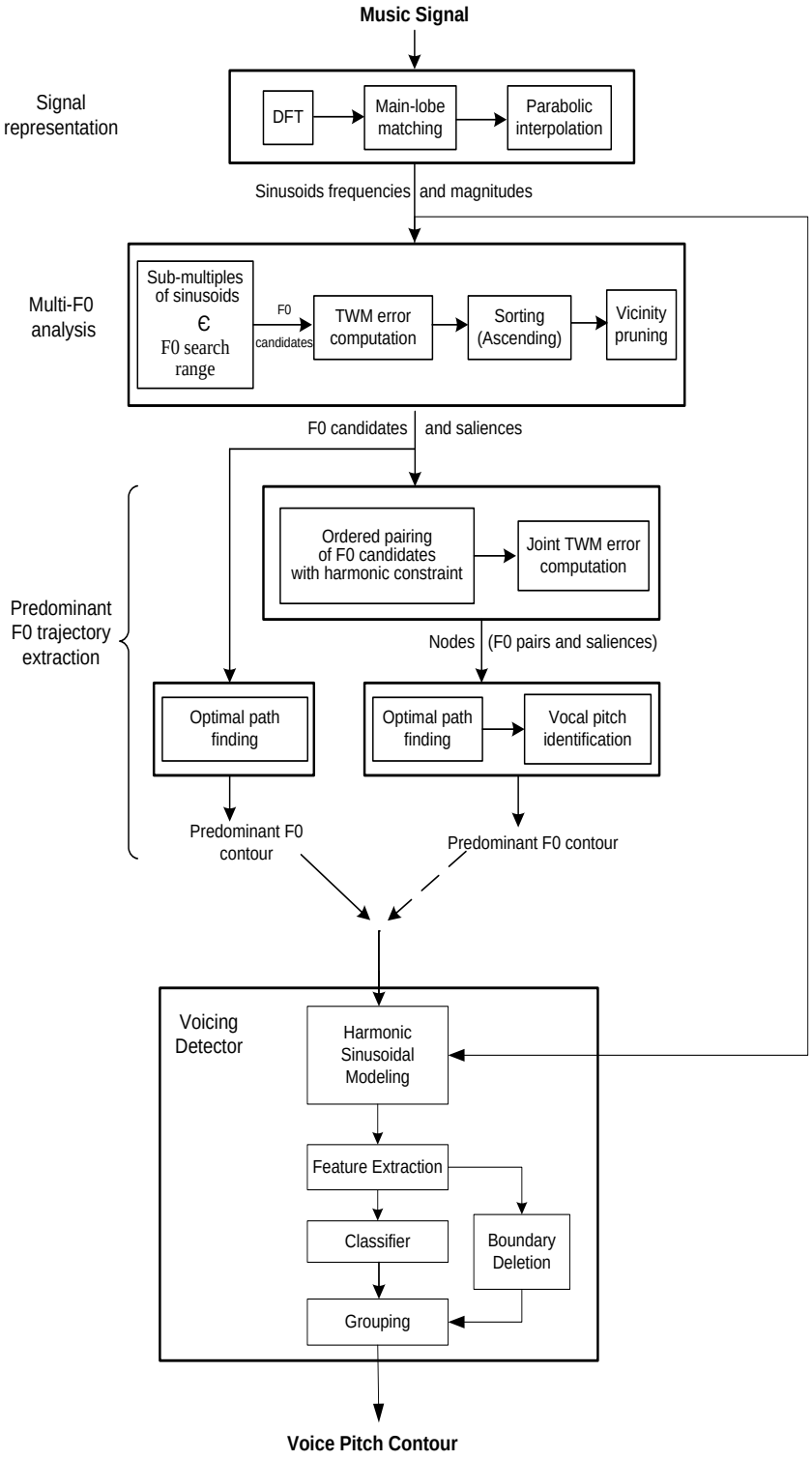


Figure 10.1: Final melody extraction system

In investigating the signal representation block we found that a main-lobe matching technique consistently outperforms other sinusoid identification techniques in terms of robustness to polyphony and non-stationarity of the target signal. Additionally a signal-sparsity driven adaptive multi-resolution approach to signal analysis was found to be superior to a fixed single- or multi-resolution signal analysis. A measure of normalized kurtosis was found to result in superior sinusoid detection performance than other measures of signal sparsity for simulated and real signals. The investigations on window-length adaptation were conducted towards the end of this thesis and are yet to be incorporated in our final melody extraction system.

In the multi-F0 analysis module we found that it was possible to detect the voice-F0 consistently by applying some modifications to a known monophonic PDA (TWM). This was done by explicitly separating the F0-candidate detection and salience computation steps. The use of well-formed sinusoids in the F0 candidate identification stage and the TWM error as the salience function led to robustness to pitched accompaniment.

In the predominant-F0 trajectory formation stage we favored the use of a dynamic programming-based optimal path finding algorithm. We modified this algorithm to track ordered pairs of F0s. These pairs were formed under harmonic constraints so as to avoid the tracking of an F0 candidate and it's multiple. The final predominant-F0 contour was identified by use of a voice-harmonic frequency instability based feature. This enhancement resulted in a significant increase in the robustness of the system when presented with polyphonic music with loud pitched accompaniment.

The singing voice detection module was considered as an independent module. Here we used a machine learning approach to singing voice detection. It was found that using an isolated spectral representation of the predominant source, using the previously extracted predominant-F0, resulted in a significant improvement in the classification performance of static timbral features. This performance was further improved by combining three classifiers' outputs, each trained on static timbral, dynamic timbral and dynamic F0-harmonic features respectively. These improvements in classification performance were consistently seen across five distinctly different cross-cultural music genres. The use of a predominant-F0 based energy feature for boundary detection was seen to result in homogenous segments that, when used to post-process the classifier labels, improved singing voice detection performance further.

For the purpose of testing our system with different parameter settings for different polyphonic signals, we developed a graphical user interface that enabled the semi-automatic usage of our melody extractor. We incorporated several features that enabled the easy subjective evaluation of the extracted voice-pitch contours via audio re-synthesis and visual display. In addition to evaluating our system, we found that users, with a little training, could quickly extract high-accuracy vocal pitch contours from polyphonic music, by intelligently varying some intuitive parameters or choosing between some pre-set parameter settings. Consequently this interface was found to be of practical utility for melody extraction for use in pedagogy or in MIR applications such as QBSH.

## 10.2 Future Work in Melody Extraction

Although the semi-automatic GUI running our back-end melody extraction system results in high-accuracy pitch contours for a variety of polyphonic music, a fully automatic high-accuracy melody extraction algorithm is as yet unavailable. The future work described here involves steps towards automating the melody extraction tool as far as possible.

### 10.2.1 Signal Representation

Using a fixed window-length is unadvisable for the melody extraction problem given the diversity in the dynamic nature of the underlying voice and accompaniment signals. It was seen in Chapter 3 that adapting the window-lengths using a signal derived sparsity measures improved the sinusoid detection performance consistently in audio containing polyphony and vocal pitch dynamics. Further the consistently top performing submission to MIREX (Dressler, 2006) uses a multi-resolution signal representation while we have used fixed resolution in our system. However, as mentioned before, a signal-adaptive multi-resolution signal representation has yet to be integrated into the front-end of our melody extraction algorithm.

### 10.2.2 Predominant-F0 Tracking

Currently we are using only a single temporal feature for voice pitch identification in our Dual-F0 framework. We could, in the future, additionally utilize the timbral and F0 temporal dynamics, as described in Section 6.2.2 for identifying a single predominant-F0 contour.

Collisions between voice and instrument pitches often causes the dual-F0 tracking output contours to switch between tracking the voice and instrument pitch contours. The use of

predictive models of F0 contours, similar to those used for sinusoidal modeling in polyphony (Lagrange, Marchand, & Rault, 2007), may be investigated to ensure F0 continuity of the contours output by the dual-F0 tracking system across F0 collisions. To avoid the negative effects of spurious candidate tracking at the exact F0 collision location care would have to be taken to ensure that both contours be assigned the same F0 value at that location.

It would also be interesting to experiment with the proposed melody extraction algorithms for music with lead melodic instruments other than the human singing voice. Preliminary tests have shown satisfactory results for a variety of Indian music performances of solo instruments, with percussive and drone accompaniment, such as the flute, violin, saxophone, *sitar*, *shehnai*, *sarod*, *sarangi* and *nadaswaram*. However, competing melodic instruments pose a problem since the predominant F0 extraction system output may switch between the pitches contours of individual instruments. Further, melody extraction in instruments like the sitar, which have a set of drone strings plucked frequently throughout the performance, can be problematic when the drone strings overpower the main melody line.

### 10.2.3 Singing Voice Detection

The present dataset used for validating cross-cultural robustness of the selected features is restricted to five genres, three of them being sub-genres of Indian music, albeit with very different signal characteristics in terms of singing and instrumentation. In order to validate our work further we need to extend the performance evaluation to larger datasets for each of the existing genres and also incorporate new culturally distinct genres such as Chinese music which has a fair share of vocal music with concurrent melodic instrumentation.

Use of the boundaries provided by the boundary detection algorithm in post-processing the classifier output did not result in as large an increase on the multi-cultural datasets as it did on the SVD-Hind dataset. This is probably due to the additional presence of the loud pitched accompaniment in the former, which may have missed sung-phrase boundaries. We would like to investigate the use of dynamic features in marking sung-phrase boundaries in order to further enhance the performance of the class label grouping algorithm.

On the machine learning front the use of different classifiers such as the commonly used Support Vector Machines (SVM) and the automatic adjustment of classifier weighting for maximization of classification performance merit investigation. Machine learning-based approaches to SVD require that the training data be as representative as possible. In the present context it is near impossible to capture all the possible inter- and intra-cultural

diversity in the underlying signals. A bootstrapping approach to SVD, such as the approach proposed by Tzanetakis (2004) may be considered.

## 10.3 Future Work on the Use of Melody Extraction

One of the applications we would like to investigate is the use of our melody extractor in extracting reference melodic templates from polyphonic songs in a QBSH tool designed for Indian music. Further these extracted melodies can also be used in a singing evaluation tool designed for musical pedagogy.

Extracted melodies from Indian classical vocal performances can be used for musicological analyses on the use of different pitch-ornaments. These can also be used to extract raga information, which can be used for indexing audio or for musicological analysis.

Duets are songs in which two singers sing separate melodic lines. Often in harmonized western duets, the melodic line of each singer is composed so as to sound pleasant when both are sung together. In this context, it is sometimes difficult to transcribe the melodic line of each singer. The dual-F0 tracking mechanism could be applied to duet melody tracking.

# Appendix

# Pre-Processing for Instrument Suppression

In this section we present two techniques for suppressing flat-note instruments. The first of these is for music in which there is a perpetually present drone, such as Hindustani or Carnatic music. It relies on the relative non-stationarity of the drone signal. The second technique relies on the extracted predominant-F0 contour to suppress the relatively stable harmonics of flat-note accompanying instruments. This will not be useful for suppressing instruments that are capable of continuous pitch modulations such as the violin.

## A.1. Drone (Tanpura) Suppression

When the predominant-F0 tracker (used in our MIREX submission) was applied to Hindustani vocal music, it was found that in some segments where the voice energy was very low, the pitch tracker was tracking the steady tanpura F0 rather than the voice F0. An illustration of such occurrences is given in Figure A. 1, which displays the estimated melodic contours for the start of the first phrase and the end of the second phrase of a recording of a Hindustani classical female vocal performance. The tonic, indicated by the dotted line, is at 245 Hz. For both phrases, the only utterance sung is /a/. In Figure A. 1.a. the pitch tracker is

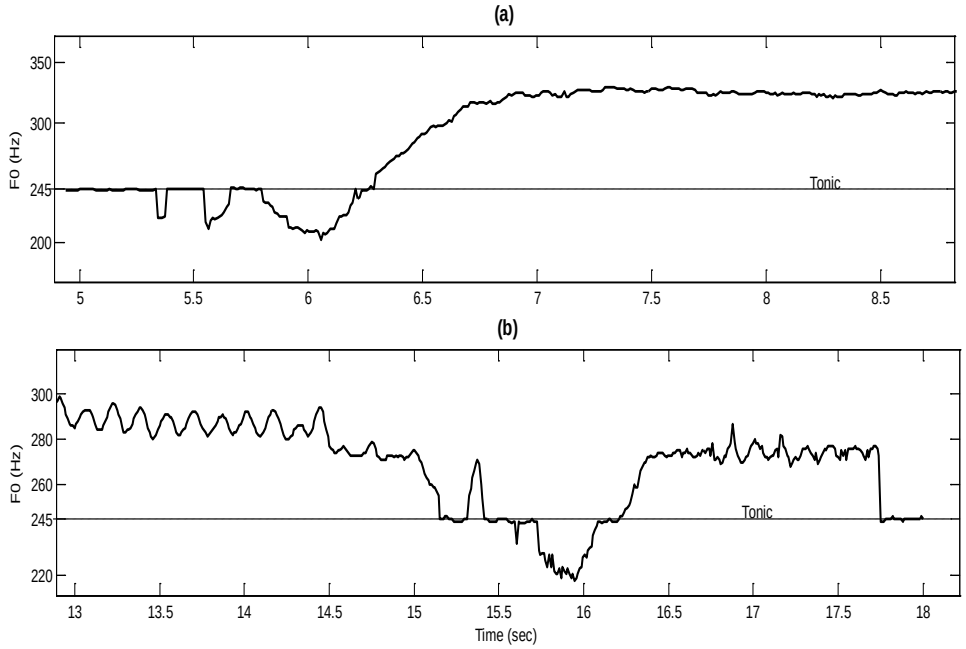


Figure A. 1: Estimated voice pitch contour of (a) the start of the first phrase and (b) the end of the second phrase of a Hindustani female vocal performance. The dotted line indicates the tonic at 245 Hz.

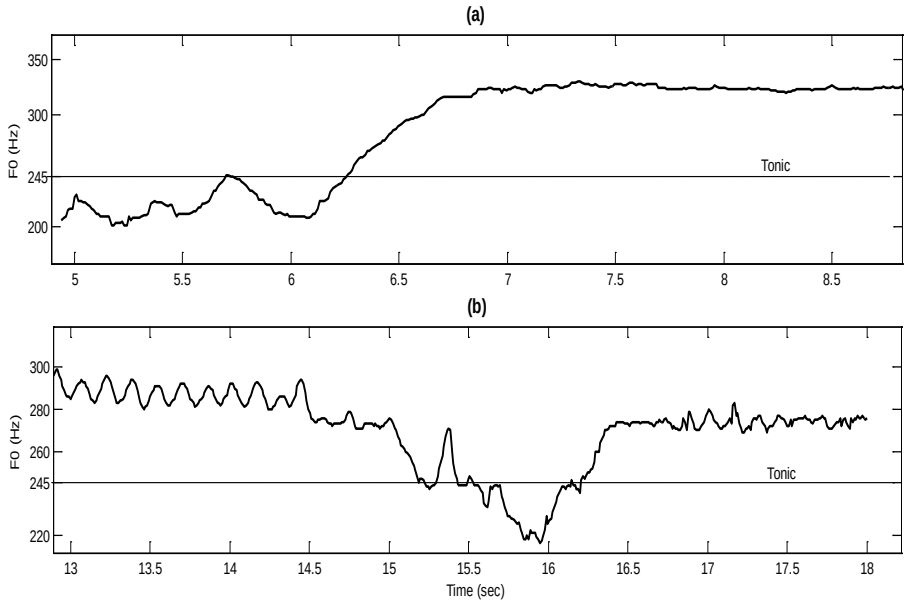


Figure A. 2: Estimated voice pitch contour of (a) the start of the first phrase and (b) the end of the second phrase of a Hindustani female vocal performance after pre-processing using spectral subtraction ($\alpha$=1.0). The dotted line indicates the tonic at 245 Hz.

incorrectly latching onto the tonic thrice between 5 and 6 seconds, and some clearly audible modulations in the melody of the original audio are lost. In Figure A. 1.b. the pitch tracker is incorrectly latching onto the tonic at the end of the phrase i.e. between 17.75 an 18 seconds, whereas no such pitch is heard from the original audio for the same time location.

In both of the above cases, the voice energy is very low during the incorrectly estimated segments of the melody. In addition, there is no tabla present in these segments as well. Since the only other tonal presence is the tanpura which repeatedly plucks a string(s) pitched at the tonic, the inference is that the pitch tracker is tracking the tanpura F0 during these segments. This indicates the need for pre-processing for tanpura suppression prior to the application of the TWM + DP melody extractor.

### A.1.1 Spectral Subtraction (SS)

Spectral subtraction is a well known technique for noise suppression and has been used extensively in speech processing (Boll, 1979). As its name implies, it involves the subtraction of an average noise magnitude spectrum, computed during non-speech activity, from the magnitude spectrum of a noisy signal. The reconstructed signal, using the updated magnitude spectrum and phase spectrum of the noisy signal, should have a lower noise floor. The assumptions made are that (1) the noise is additive, (2) the noise is stationary to the degree that its spectral magnitude value just prior to speech activity equals its expected value during speech activity. Details of the operation of spectral subtraction are given below.

*Operation*

Let *s(k)* be speech signal to which the noise *n(k)* is added and their sum is denoted by *x(k).* Their respective short time Fourier transforms are given by $S(e^{j\omega})$, $N(e^{j\omega})$ and $X(e^{j\omega})$. A voice activity detector is used to indicate the presence/absence of speech activity. Whenever, speech activity is absent, a noise bias $\mu(e^{j\omega})$ is computed by the average value of $|N(e^{j\omega})|$ during non-speech activity. Once speech activity restarts, the most recent value of the noise bias is used to suppress the noise in the reconstructed speech signal using the following equation.

$$\hat{s}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{S}(e^{j\omega})e^{j\omega k}d\omega \quad \text{where } \hat{S}(e^{j\omega}) = [|X(e^{j\omega})| - \alpha.\mu(e^{j\omega})]e^{j\theta_x(e^{j\omega})} \tag{A.1}$$

Here $\hat{s}(k)$ is the resulting noise-suppressed speech signal, $\theta_x(e^{j\omega})$ is the phase component of $X(e^{j\omega})$ and $\alpha$ is a factor that controls the amount of subtraction that can take place. Also, after magnitude spectral subtraction, the resulting magnitude spectrum is half wave rectified to avoid any negative values that might occur.

*Adaptation to Tanpura Suppression*

Most Indian classical music recordings have an initial segment, which may last around 2 to 10 seconds, of only tanpura. Since multiple strings with differing F0s are plucked while playing the tanpura, a long-term average of the tanpura magnitude spectrum over this initial segment will have peaks at harmonics of the different F0. But because of averaging over multiple plucks of different strings these peaks will have reduced strength as compared to peaks in the short time magnitude spectrum of a single string pluck. The subtraction of such a long term average magnitude spectrum from the magnitude spectra of all the voice frames should suppress the tanpura harmonics in that frame without causing any significant degradation to the voice harmonics.

For each of the four audio files considered the initial tanpura segment was manually located, an average magnitude spectrum computed over this segment and subsequently subtracted from all subsequent voiced frames i.e. all frames within each singing burst. The tanpura suppressed signal was reconstructed from the modified magnitude spectra and original phase spectra using the overlap and add (OLA) method, which has been shown to provide perfect signal reconstruction in the absence of any spectral modification and proper choice and spacing of the analysis windows (Allen, 1977).

*Results*

Figure A. 2 shows the results of applying spectral subtraction ($\alpha$ =1.0) to the same segments that were shown in Figure A. 1. We can see that the pitch tracker does not latch on to the tonic at the beginning of phrase 1 (Figure A. 2.a) nor at the end of phrase 2 (Figure A. 2.b) as was happening earlier. On comparing the audio synthesized using the estimated melody after pre-processing it appears that the voice modulations in the beginning of phrase 1, that were missing from the re-synthesis of the previously estimated melody, are now present. The melody at the end of phrase 2 is now a closer match to the melody of the original audio.

### A.1.2 Other Approaches to Tanpura Suppression Considered

*Inverse comb filtering*

Prior to the successful application of spectral subtraction, two other approaches to tanpura suppression were considered. One of these involved applying an inverse comb filter with notches at harmonics of the tanpura strings' F0s. Typically the tanpura has four strings. Two of the strings are always pitched at the tonic and one is always pitched at an octave above the tonic. The fourth string can be pitched at Pa (fifth), Ma (fourth) and Ni (Pandya, 2005). So two inverse comb filters would have to be applied with notches at harmonics of the tonic and whichever F0 the fourth string is pitched at. This requires apriori knowledge of the location of the tonic and the fourth string F0. While it is possible to estimate the tonic F0 by applying a pitch tracker to the tanpura signal, the F0 of the fourth string is difficult to estimate since the tanpura signal is dominated by the tonic harmonics.

On applying the inverse comb filter with notches at harmonics of the tonic to some segments of the female Hindustani vocal recording used in the previous section, where the tonic was being incorrectly tracked, it was found that instead of tracking the voice F0, the pitch tracker latched on to Ma (fourth).

*Relative Spectra (Rasta) pre-processing*

Rasta processing (Hermansky & Morgan, 1994) has been successfully applied to noise suppression in speech. It employs band-pass filtering of time trajectories of speech feature vectors. The assumption here again is that the trajectories of the feature vectors for the noise will change more slowly or quickly than the typical range of change in speech, which is known to have a syllabic rate of around 4 Hz. Rasta processing of speech, using a fifth-order elliptic band-pass filter with lower and upper cutoff frequencies of 1 and 15 Hz respectively, was found to have results that were comparable to spectral subtraction noise suppression, while removing the need for a voice activity detector.

In the present context, we know that the time trajectories of the feature vectors (FFT bin magnitudes centered at harmonic locations) for a single tanpura string pluck should be near stationary, since the harmonics decay very slowly. Rasta processing would successfully be able to suppress the tanpura without affecting the voice spectral content if the time trajectories of the FFT Bin magnitudes centered at harmonic bin locations for the voice vary at rates much higher than those for the tanpura i.e. the spectral content of the time trajectories of the corresponding FFT Bin magnitudes have non-overlapping content. Unfortunately, the

rate of variation in singing is much slower than in speech. For a typical classical vocal performance, the rate of variation can vary widely within the performance with very slow variation during the initial portions of the song and faster variations during *taans* (rapid passages). So while rasta may be successfully able to suppress the tanpura without affecting the voice during rapid variations it may also degrade the voice during steady notes.

To illustrate this consider Figure A. 3, which shows the spectral content in the time-trajectory of (a) a single FFT bin centered at a voice harmonic over a single steady held note of duration 1.99 seconds and (b) a single FFT bin centered at a tanpura harmonic over a single string pluck, after the onset, for a duration of 1.67 seconds. Both the spectra exhibit the maximum energy at DC, from which we can infer that rasta processing will severely attenuate the content in both the above bins.

Indeed, when we applied the filters described in (Hermansky & Morgan, 1994) we found that while the tanpura partials are attenuated, voice partials during steady notes were also attenuated. Interestingly, voice partials during steady notes held with vibrato were left unaffected.

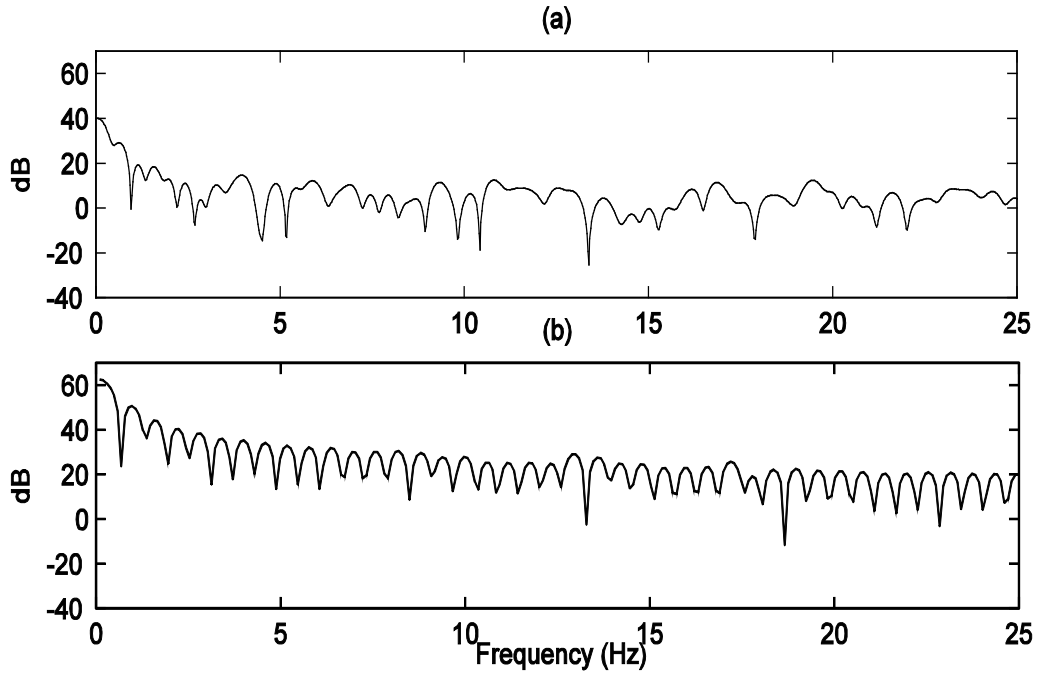


Figure A. 3: Spectra of the time trajectories of (a) FFT bin centered at a voice harmonic during a steady note held for 1.99 seconds (199 frames) for (b) FFT bin centered at a tanpura harmonic for a duration of 1.67 seconds (167 frames) for a single tanpura pluck.

### A.1.3 Effect of Spectral Subtraction on Singing voice Detection

In Section A.1.1 it was subjectively shown that *tanpura* suppression using a scheme based on spectral subtraction (SS) (Boll, 1979) was able to correct pitch estimation errors at the starts and ends of sung phrases where the voice was soft. We applied SS to the data used in the singing voice detection experiment described in Section 7.1.2.1. The 10-fold CV accuracies for all three feature sets after pre-processing is now reported in Table A 1. It was found that running the experiment after pre-processing improved the performance for all feature sets. The drawback of such a scheme is that it requires a *tanpura* segment of 4 seconds or more at the beginning of the song for this method to be possible. This restricts the use of the SS-based pre-processing to specific genres of music in which there is an initial drone-only segment of audio present, such as the classical Indian genres of Hindustani or Carnatic music.

Table A 1: Comparison of 10-fold cross validation accuracies for SVD experiment using different feature sets after SS

| Feature set | Vocal accuracy (%) | Instrumental accuracy (%) | Overall accuracy (%) |
|---|---|---|---|
| FS1 | 81.63 ± 9.30 | 81.08 ± 6.52 | 81.49 ± 6.38 |
| FS2 | 86.46 ± 8.9 | 81.41 ± 10.19 | 85.13 ± 5.77 |
| FS3 | 90.32 ± 3.54 | 90.12 ± 7.95 | 90.26 ± 2.61 |

## A.2 Flat-Note Instrument Suppression

In section 7.1.2.1 it was observed that the singing voice detection system showed low instrumental recall accuracy (Table 7.12) in the presence of a loud secondary melodic instrument (*harmonium).* Rather than investigate methods of increasing robustness of the features to such harmonically rich, loud accompaniment, an alternative is to investigate pre-processing techniques to suppress such loud pitched accompaniment. Standard methods of pre-processing for noise suppression of speech such as spectral subtraction (SS) (Boll, 1979) and RASTA (Hermansky & Morgan, 1994) are not expected to be effective in this context. SS was previously shown to be effective for *tanpura* suppression but is too restrictive in that it requires that the secondary melodic instrument be relatively stationary, a condition only fulfilled by the *tanpura* but not by other instruments. RASTA involves the attenuation of the content of a particular frequency bin based on the time-evolution of the magnitude values in that bin. However rates of decay of amplitude of *harmonium* harmonics are highly variable

since the loudness is governed by the rate of air release from the bellows (controlled by the human hand), which play a similar role to the human lungs in speech/singing production.

On comparing the spectrograms of a harmonium signal to a voice signal (Figure A. 4) we can see that the harmonics of the harmonium appear to be very stable (in frequency) till high frequencies (5 kHz) but the voice harmonics show increasing variation at higher frequencies. From a production perspective, the harmonium pitch is governed by the length and thickness of freely vibrating, metal reeds, which are fixed physical constraints for a given harmonium. So the harmonics for a given *harmonium* pitch are expected to be very stable in frequency. For singing however, it has been observed that there is a natural instability in the voice harmonics that increases with increase in frequency. Even for sung notes held without vibrato there will be period-to-period variations in glottal pulse duration, called jitter, which lead to the naturalness in the perception of the voice. The amount of jitter is usually about 0.5 – 1.0 % of the pitch period and so we can expect this jitter to have increasingly larger values at higher harmonics.

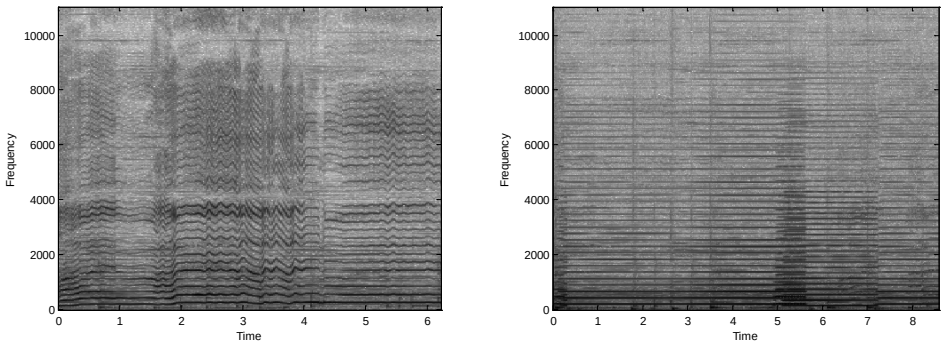


Figure A. 4: Spectrograms of voice (left) and harmonium (right) signals

This section investigates whether this difference in the frequency stability of harmonics for the voice and the *harmonium* can be exploited to suppress the latter. In order to capture this instability it would be necessary to first derive an intermediate representation for the time evolution of each harmonic. One such representation that can be used is the Harmonic Sinusoidal Model (described in Section 6.2.1). Here individual harmonics are represented by tracks whose amplitude, frequency and phase values could vary over time. Isolating tracks based on their instability also finds application in source segregation.

Next we describe a method to prune stable instrument harmonic tracks in the harmonic sinusoidal model based on short-duration track variance. Singing voice detection results for

experiments using the surviving-track energy as a feature in the voicing detector are then presented.

### A.2.1 Standard Deviation Pruning and Feature Definition

In order to attenuate stable instrument tracks, we use a novel track pruning criteria based on standard deviation (SD). We would like to prune tracks whose SDs are below some threshold (indicating that they belong to a stable-pitch instrument). Computing the SD for individual tracks, however, assumes that the entire track belongs to a single sound source. The intersection of harmonics of different sound sources may result in different segments of the same sinusoidal track belonging to different sources. In such cases, regions of tracks that actually belong to stable-pitch instruments may escape pruning since the SD of the entire track may be greater than the threshold. To avoid such an occurrence, the SD is computed over short non-overlapping track segments of length 200 ms and only track segments whose SD is below a particular threshold (here 2 Hz) are pruned. The track pruning procedure is summarized in Figure A. 5.

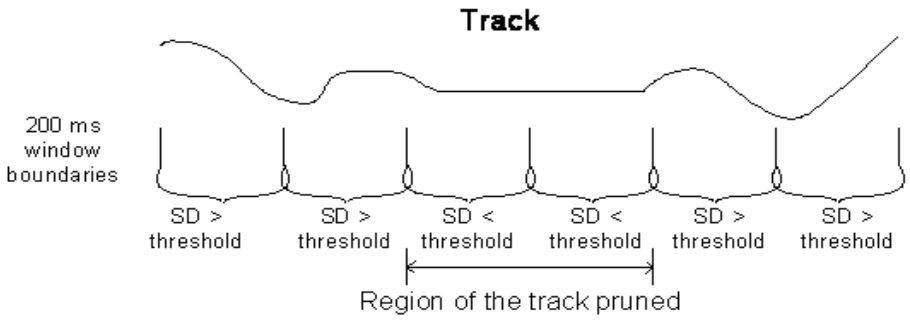


Figure A. 5: Moving window based standard deviation pruning of tracks

To visualize the effect of track pruning, consider Figure A. 6.a, which displays the spectrogram of a mixture of *harmonium* (present throughout) and a Hindustani classical vocal phrase (starting at 1.2 sec). Figure A. 6.b. displays the result of harmonic sinusoidal model for the *harmonium*-voice mixture. Both the (clearly stable) *harmonium* and voice tracks are well formed. However, some tracks formed at collisions between the voice and *harmonium* harmonics can be observed to approximately follow the latter. Figure A. 6.c. shows the result of track pruning based on frequency instability. The *harmonium* tracks have been erased but the majority of the voice tracks survive.

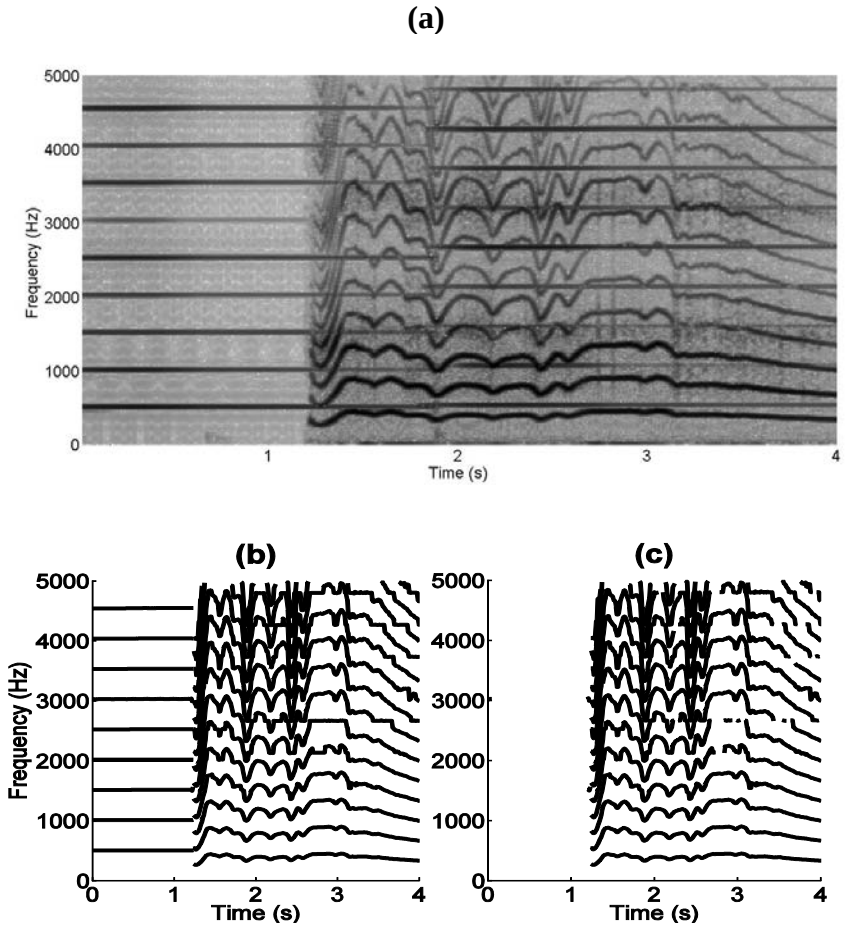


Figure A. 6: (a) Spectrogram of harmonium-voice mixture (b) Sinusoidal tracks before and (c) after SD pruning with a 2 Hz threshold.

### A.2.2 Singing Voice Detection Experiments Using Track Pruning

For use in the singing voice detection stage we consider another feature called the called sinusoidal track harmonic energy (STHE), which is basically the NHE computed from the sinusoids that survive the above track pruning stage. As with NHE, the STHE is normalized by its maximum attained value over a single musical performance. This feature is now included in the singing voice detection evaluation of Section 7.1.2.1.

The 10-fold cross validation was performed using the STHE feature after SD Pruning (SD) for the 2-class problem with [4 4] components for the entire training data. The different SD thresholds tried were 2 Hz, 3Hz and 5 Hz. The first three rows of Table A 2. shows the cross validation results for the STHE feature set after SD pruning with different SD thresholds. It can be seen that a SD threshold of 2 Hz gives the best trade-off between vocal and instrumental accuracy. Another point of interest is that the vocal accuracy is comparatively lower than the instrumental accuracy. This is mainly due to the presence of

stable voice tracks especially at lower frequencies. As the SD threshold is increased it can be seen that the vocal accuracy increases due to the survival of more low frequency voice tracks. These results compare favorably with the 10-fold cross validation results for the other feature sets shown in Table 7.10.

Table A 2: 10-fold cross validation results for the STHE feature for SVD-Hind Training data for different track-pruning thresholds.

| **Feature set** | **Parameters** | **Vocal accuracy (%)** | **Instrumental accuracy (%)** | **Overall accuracy (%)** |
|---|---|---|---|---|
| STHE | SD > 5 Hz | 76.65 ± 10.60 | 89.01 ± 7.93 | 79.87 ± 6.66 |
| STHE | SD > 3 Hz | 78.67 ± 9.40 | 88.37 ± 8.10 | 81.20 ± 5.86 |
| STHE | SD > 2 Hz | 80.87 ± 7.89 | 88.21 ± 8.40 | **82.78 ± 5.01** |
| STHE | SD >2 Hz (no Pitch pruning) | 81.26 ± 10.68 | 79.66 ± 15.02 | 80.84 ± 5.72 |

Table A 3: Comparison of testing database results for different feature sets (now including STHE as FS4) for SVD-Hind (Testing) and PF0-ICM (VTTH) data

| | **SVD-Hind (Testing)** | | **PF0-ICM (VTTH)** | |
|---|---|---|---|---|
| **Feature set** | **Vocal recall (%)** | **Instrumental recall (%)** | **Vocal recall (%)** | **Instrumental recall (%)** |
| FS1 | 92.17 | 66.43 | 91.61 | 40.91 |
| FS2 | 92.38 | 66.29 | 87.53 | 57.40 |
| FS3 | 89.05 | 92.10 | 86.60 | 45.22 |
| FS4 | 83.45 | 90.24 | 85.62 | 86.34 |

The classifier was trained with the entire dataset with the STHE feature with a SD pruning threshold of 2 Hz, and tested on the outside data for the 2-class problem with [4 4] GMM components. Both the SVD-Hind testing and the PF0-ICM (VTTH) data were tested separately. The results for both the testing data for STHE feature (as FS4) are shown along with the results of (FS1-FS3) of Table 7.10 in Table A 3. The STHE performs better in the instrumental sections as compared to the outside dataset results of FS1 (MFCC) and FS2 (the acoustic feature set). For the PF0-ICM (VTTH) signals it can be observed that the performance of the STHE feature does not degrade even in the presence of the loud secondary melodic instrument (i.e. *harmonium*) as compared to the performance of the other feature sets.

# References


Allen, J. (1977). Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing , ASSP-25* (3), 235-238.

Arias, J., Pinquier, J., & Andre-Obrecht, R. (2005). Evaluation of classification techniques for audio indexing. *13th European Signal Processing Conference (EUSIPCO).* Istanbul.

Aucouturier, J.-J., & Patchet, F. (2007). The influence of polyphony on the dynamic modeling of musical timbre. *Pattern Recognition Letters , 28* (5), 654-661.

Badeau, R., Richard, G., & David, B. (2008). Fast and stable YAST algorithm for principal and minor subspace tracking. *IEEE Transactions on Signal Processing , 56* (8), 3437-3446.

Battey, B. (2004). Bezier spline modeling of pitch-continuous melodic expression and ornamentation. *Computer Music Journal , 28* (4), 25-39.

Battiti, R. (1994). Using Mutual Information for selecting features in a Supervised Neural Net Learning. *IEEE Transactions on Neural Networks , 5* (4), 537-550.

Belle, S., Rao, P., & Joshi, R. (2009). Raga identificiation by using swara intonation. *Frontiers of Research in Speech and Music (FRSM).* Gwalior.

Berenzweig, A., & Ellis, D. (2001). Locating singing voice segments within music signals. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* New York.

Berenzweig, A., Ellis, D., & Lawrence, S. (2002). Using voice segments to improve artist classification of music. *22nd International Conference of the Audio Engineering Society.* Espoo.

Betser, M., Collen, P., Richard, G., & David, B. (2008). Estimation of frequency for AM/FM models using the phase vocoder framework. *IEEE Transactions on Signal Processing , 56* (2), 505-517.

Boersma, P. (1993). Accurate Short-term Analysis of the Fundamental Frequency and the Harmonics-to-Noise. *Institute of Phonetic Sciences, 17*, pp. 97-110. Amsterdam.

Boersma, P., & Weenink, D. (2005). *PRAAT: Doing Phonetics by Computer*. Retrieved from Computer Program.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Audio, Speech, and Signal Processing , 27* (2), 113-120.

Bor, J., Rao, S., & van der Meer, W. (1999). *The raga guide: A survey of 74 Hindustani ragas.* London: Zenith media.

Boumann, C. *Cluster: An unsupervised algorithm for modeling Gaussian mixtures.* Retrieved June 9, 2009, from http://www.ece.purdue.edu/~bouman

Brossier, P. *Aubio: a library for audio labeling*. Retrieved August 27, 2009, from Computer Program: http://subio.org

Brossier, P. (2006). *The Aubio library at MIREX 2006.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2006:Audio_Melody_Extraction_Results

Brown, J. (1992). Music fundamental frequency tracking using a pattern matching method. *Journal of the Acoustical Society of America , 92* (3), 1394-1402.

Burred, J., Robel, A., & Sikora, T. (2010). Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *IEEE Transactions on Audio, Speech, and Language Processing , 18* (3), 663-674.

Cancela, P. (2008). *Tracking melody in polyphonic audio.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2008:Audio_Melody_Extraction_Results

Cancela, P. (2008). *Tracking melody in polyphonic audio. MIREX 2008.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2008:Audio_Melody_Extraction_Results

Cancela, P., Lopez, E., & Rocamora, M. (2010). Fan chirp transform for music representation. *13th International Conference on Digital Audio Effects (DAFx-10).* Graz, Austria.

Cannam, C. *Sonic Visualiser*. Retrieved August 27, 2009, from Computer Program: http://www.sonicvisualiser.org

Cano, P. (1998). Fundamental frequency estimation in the SMS analysis. *COST G6 Conference on Digital Audio Effects.* Barcelona.

Cao, C., Li, M., Liu, J., & Yan, Y. (2007). Singing melody extraction in polyphonic music by harmonic tracking. *International Conference on Music Information Retrieval.* Vienna.

Celemony. *Direct Note Access: the new Melodyne dimension*. Retrieved August 27, 2007, from Celemony: Tomorrow's audio today: http://www.celemony.com/cms/index.php?id=dna

Chou, W., & Gu, L. (2001). Robust singing detection in speech/music discriminator design. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* Salt Lake City.

Christensen, M., Stoica, P., Jakobsson, A., & Jensen, S. (2008). Multi-pitch estimation. *Signal Processing , 88* (4), 972-983.

Cook, P. (1999). Pitch, periodicity and noise in the voice. In *Music, Cognition and Computerized Sound* (pp. 195-208). Cambridge: MIT Press.

Cornelis, O., Lesaffre, M., Moelants, D., & Leman, M. (2009). Access to ethnic music: Advances and perspectives in content-based music information retrieval. *Signal Processing Special issue on Ethnic Music Audio Documents: From the preservation to the fruition , 90* (4), 1008-1031.

de Cheveigne, A. (2006). Multiple F0 Estimation. In D. Wang, & G. Brown (Eds.), *Computational Auditory Scene Analysis: Principles, Algorithms and Applications.* Wiley-IEEE Press.

de Cheveigne, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America , 111* (4), 1917-1930.

Downie, S. (2003). Music information retrieval. *Annual review of information science and technology , 37*, 295-340.

Downie, S. (2008). The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustics, Science and Technology , 29* (4), 247-255.

Dressler, K. (2006). *An Auditory Streaming Approach to melody extraction.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2006:Audio_Melody_Extraction_Results

Dressler, K. (2010). *Audio melody extraction for MIREX 2009.* Ilmenau: Fraunhofer IDMT.

Dressler, K. (2006). Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. *9th International Conference on Digital Audio Effects.* Montreal.

Duan, Z., Zhang, Y., Zhang, C., & Shi, Z. (2004). Unsupervised single-channel music source separation by average harmonic structure modeling. *IEEE Transactions on Audio, Speech, and Language Processing , 16* (4), 766-778.

Durrieu, J.-L., Richard, G., & David, B. (2009). *A source/filter approach to audio melody extraction.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2009:Audio_Melody_Extraction_Results

Durrieu, J.-L., Richard, G., & David, B. (2008). *Main melody extraction from polyphonic music excerpts using a source/filter model of the main source.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2008:Audio_Melody_Extraction_Results

Durrieu, J.-L., Richard, G., David, B., & Fevotte, C. (2010). Source/Filter model for unsupervised main melody extraction from polyphonic signals. *IEEE Transactions on Audio, Speech, and Language Processing , 18* (3), 564-575.

Duxbury, C., Bello, J. P., Davies, M., & Sandler, M. (2003). Complex domain onset detection for musical signals. *6th International Conference on Digital Audio Effects (DAFx-03).* London.

Emiya, V., Badeau, R., & David, B. (2008). Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches. *European Signal Processing Conference.* Lausanne.

Every, M. (2006). Separation of musical sources and structure from single-channel polyphonic recordings. *Ph.D. Dissertation* . York: University of York.

Every, M., & Jackson, P. (2006). Enhancement of harmonic content in speech based on a dynamic programming pitch tracking algorithm. *InterSpeech.* Pittsburgh.

Fernandez-Cid, P., & Casajus-Quiros, F. (1998). Multi-pitch estimation for polyphonic musical signals. *IEEE International Conference on Acoustics, Speech and Signal Processing.* Seattle.

Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty. *IEEE International Conference on Multimedia and Expo (ICME).* New York.

Forney, G. (1973). The viterbi algorithm. *Proceedings of the IEEE , 61* (3), 268-278.

Fuhrmann, F., Haro, M., & Herrera, P. (2009). Scalability, Generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music. *10th International Conference on Music Information Retrieval (ISMIR).* Kobe.

Fujihara, H., & Goto, M. (2008). Three techniques for improving automatic synchronization between music and lyrics: Fricative detection, filler model and novel feature vectors for vocal activity detection. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* Las Vegas.

Fujihara, H., Goto, M., Kitahara, T., & Okuno, H. (2010). A modeling of singing voice robust to accompaniment sounds and its applications to singer identification and vocal-timbre-similarity-based music information retrieval. *IEEE Transactions on Audio, Speech, and Language Processing , 18* (3), 638-648.

Fujihara, H., Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H. (2006). F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and viterbi search. *IEEE International Conference on Acoustics, Speech and Signal Processing.* Toulouse.

Gomez, E., & Herrera, P. (2008). Comparative analysis of music recordings from Western and non-Western traditions by automatic tonal feature extraction. *Empirical Musicology Review , 3* (3), 140-156.

Goodwin, M. (1997). Adaptive signal models: Theory, algorithms and audio applications. *Ph.D. Dissertation* . Massachusetts Institute of Technology.

Goto, M. (2004). A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real world audio signals. *Speech Communication , 43*, 311-329.

Griffin, D., & Lim, J. (1988). Multiband Excitation Vocoder. *IEEE Transactions on Acoustics, Speech and Signal Processing , 36* (8), 1223-1235.

Hall, M., Frank, E., Holmes, G., Pfarhringer, B., Reutemann, P., & Witten, I. (2009, June). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter , 11* (1), pp. 10-18.

Hermansky, H., & Morgan, N. (1994). RASTA Processing of Speech. *IEEE Transactions on Speech and Audio Processing , 2* (4), 578-589.

Hermes, D. (1988). Measurement of pitch by sub-harmonic summation. *Journal of the Acoustical Society of America , 83* (1), 257-264.

Hermes, D. (1993). Pitch analysis. In *Visual Representation of Speech Signals.* Chichester: John Wiley and Sons.

Hess, W. (2004). Pitch determination of acoustic signals - An old problem and new challenges. *International Conference on Acoustics*, (pp. 1065-1072). Kyoto.

Hess, W. (1983). *Pitch determination of speech signals- algorithms and devices.* Berlin: Springer.

Hsu, C.-L., & Jang, R. (2010). On the improvement of singing voice separation for monoaural recordings using the MIR-1k dataset. *IEEE Transactions on Audio, Speech, and Language Processing , 18* (2), 310-319.

Hsu, C.-L., & Jang, R. (2010). *Singing pitch extraction at MIREX 2010.* Retrieved from MIREX Audio Melody Extraction Contest: http://nema.lis.illinois.edu/nema_out/mirex2010/results/ame/adc04/index.html

Hsu, C.-L., Jang, R., & Chen, L.-Y. (2009). *Singing pitch extraction at MIREX 2009.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2009:Audio_Melody_Extraction_Results

Hurley, N., & Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory , 55* (10), 4723-4741.

Jairazbhoy, N. (1999). *The Rāgs of North Indian Music.* Bombay: Popular Prakashan.

Johnston, J. (1988). Transform coding of audio signals using perceptual noise criteria. *IEEE Journal of Selected Areas in Communication , 6* (2), 314-323.

Jones, D., & Parks, T. (1990). A high-resolution data-adaptive time-frequency representation. *IEEE Transactions on Acoustics, Speech and Language Processing , 38* (12), 2127-2135.

Joo, S., Jo, S., & Yoo, C. (2009). *Melody extraction from polyphonic audio signal.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2009:Audio_Melody_Extraction_Results

Joo, S., Jo, S., & Yoo, C. (2010). *Melody extraction from polyphonic audio signal.* Retrieved from MIREX Audio Melody Extraction Contest: http://nema.lis.illinois.edu/nema_out/mirex2010/results/ame/adc04/index.html

Keiler, F., & Marchand, S. (2002). Survey on extraction of sinusoids in stationary sounds. *5th International Conference on Digital Audio Effects.* Hamburg.

Kim, K.-H., & Hwang, I.-H. (2004). A multi-resolution sinusoidal model using adaptive analysis frame. *12th European Signal Processing Conference.* Vienna.

Kim, Y., & Whitman, B. (2004). Singer identification in popular music recordings using voice coding features. *5th International Conference on Music Information Retrieval.* Barcelona.

Kim, Y., Chai, W., Garcia, R., & Vercoe, B. (2000). Analysis of a countour-based representation for melody. *International Symposium on Music Information Retrieval.* Plymouth.

Kittler, J., Hatef, M., Duin, R., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Recognition and Machine Intelligence , 20* (3), 226-239.

Klapuri, A. (2008). Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Transactions on Audio, Speech and Language Processing , 16* (2), 255-266.

Klapuri, A. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing , 11* (6), 804-816.

Klapuri, A. (2004). Signal processing methods for the automatic transcription of music. *Ph.D. Dissertation* . Tampere: Tampere University of Technology.

Klapuri, A., & Davy, M. (Eds.). (2006). *Signal Processing Methods for Music Transcription.* Springer Science + Business Media LLC.

Lagrange, M., & Marchand, S. (2007). Estimating the instantaneous frequency of sinusoidal components using phase-based methods. *Journal of the Audio Engineering Society , 55* (5), 385-399.

Lagrange, M., Gustavo Martins, L., Murdoch, J., & Tzanetakis, G. (2008). Normalised cuts for predominant melodic source separation. *IEEE Transactions on Audio, Speech, and Language Processing , 16* (2), 278-290.

Lagrange, M., Marchand, S., & Rault, J.-B. (2007). Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds. *IEEE Transactions on Audio, Speech, and Language Processing , 15* (5), 1625-1634.

Lagrange, M., Marchand, S., & Rault, J.-B. (2006). Sinusoidal parameter extraction and component selection in a non-stationary model. *5th International Conference on Digital Audio Effects (DAFx-02)*, (pp. 59-64). Hamburg, Germany.

Lagrange, M., Raspaud, M., Badeau, R., & Richard, G. (2010). Explicit modeling of temporal dynamics within musical signals for acoustic unit similarity. *Pattern Recognition Letters , 31* (12), 1498-1506.

Levitin, D. (1999). Memory for Musical Attributes. In P. Cook (Ed.), *Music, Cognition and Computerized Sound* (pp. 214-215). MIT Press.

Li, Y., & Wang, D. (2005). Detecting pitch of singing voice in polyphonic audio. *IEEE International Conference on Acoustics Speech and Signal Processing, 3*, pp. 17-20. Philadelphia.

Li, Y., & Wang, D. (2007). Separation of singing voice from music accompaniment for monoaural recordings. *IEEE Transactions on Audio, Speech, and Language Processing , 15* (4), 1475-1487.

Lidy, T., Silla, C., Cornelis, O., Gouyon, F., Rauber, A., Kaestner, C., et al. (2010). On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing and accessing non-Western and ethnic music collections. *Signal Processing Special issue on Ethnic Music Audio Documents: From the preservation to the fruition , 90* (4), 1032-1048.

Logan, B. (2000). Mel-frequency cepstral coefficients for music modeling. *International Symposium on Music Information Retrieval.* Plymouth.

Luengo, I., Saratxaga, I., Navas, E., Hernaez, I., Sanchez, J., & Sainz, I. (2007). Evaluation of pitch detection algorithms under real conditions. *IEEE International Conference on Acoustics, Speech, and Signal Processing, IV*, pp. 1057-1060. Honolulu.

Lukashevich, H., Grunhe, M., & Dittmar, C. (2007). Effective singing voice detection in popular music using ARMA filtering. *10th International Conference on Digital Audio Effects (DAFx-07).* Bordeaux.

Maddage, N., Xu, C., & Wang, Y. (2003). A SVM-based classification approach to musical audio. *International Conference on Music Information Retrieval.* Baltimore.

Maher, R. (1990). Evaluation of a method for separating digitized duet signals. *Journal of the Audio Engineering Society , 38* (12), 956-979.

Maher, R., & Beauchamp, J. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America , 95* (4), 2254-2263.

Marchand, S., & Depalle, P. (2008). Generalization of the derivative analysis method to non-stationary sinusoidal modeling. *11th International Conference on Digital Audio Effects (DAFx-08)*, (pp. 281-288). Espoo, Finland.

Markaki, M., Holzapfel, A., & Stylianou, Y. (2008). Singing voice detection using modulation frequency features. *Workshop on Statistical and Perceptual Audition (SAPA-2008).* Brisbane.

Marolt, M. (2008). A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia , 10* (8), 1617-1625.

McAulay, J., & Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing , ASSP-34* (4), 744-754.

Nakano, T., Goto, M., & Hiraga, Y. (2007). MiruSinger: A singing skill visualization interface using real-time feedback and music CD recordings as referential data. *9th IEEE International Symposium on Multimedia Workshops.* Beijing.

Ney, H. (1983). Dynamic programming algorithm for optimal estimation of speech parameter contours. *IEEE Transactions on Systems, Man, and Cybernetics , SMC-13* (3), 208-214.

Nokia. *Qt, A cross-platform application and C++ UI framework*. Retrieved from http://qt.nokia.com/

Nwe, T., & Li, H. (2007). Exploring vibrato-motivated acoustic features for singer identification. *IEEE Transactions on Audio, Speech, and Language Processing , 15* (2), 519-530.

Nwe, T., & Li, H. (2008). On fusion of timbre-motivated features for singing voice detection and singer identification. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* las Vegas.

Painter, T., & Spanias, A. (2000). Perceptual coding of digital audio. *Proceedings of the IEEE , 88* (4), 451-513.

Paiva, R. P. (2006). Melody detection in polyphonic audio. *Ph.D. Dissertation* . University of Coimbra.

Paiva, R. P., Mendes, T., & Cardoso, A. (2006). Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience and melodic smoothness. *Computer Music Journal , 30* (4), 80-98.

Pandya, P. (2005). Beyond Swayambhu Gandhar: A spectral analysis of perceived Tanpura notes. *Ninad - Journal of the ITC Sangeet Research Academy , 19*, 5-15.

Peeters, G. (2004). *A large set of audio features for sound description (similarity and classification) in the CUIDADO project.* CUIDADO I.S.T. Project Report.

Poliner, G., & Ellis, D. (2005). A classification approach to melody transcription. *International Conference on Music Information Retrieval.* London.

Poliner, G., Ellis, D., Ehmann, A., Gomez, E., Streich, S., & Ong, B. (2007). Melody Transcription From Music Audio: Approaches and Evaluation. *IEEE Transactions on Audio, Speech and Language Processing , 15* (4), 1247-1256.

Proutskova, P., & Casey, M. (2009). You call that singing? Ensemble classification for multi-cultural collections of music recordings. *10th International Conference on Music Information Retrieval (ISMIR).* Kobe.

Rabiner, L. (1977). On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Audio, Speech, and Signal Processing , 25*, 24-33.

Rabiner, L., Cheng, M., Rosenberg, A., & McGonegal, C. (1976). A comparitive performance study of several pitch detection algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing , ASSP-24* (5), 399-418.

Raju, M., Sundaram, B., & Rao, P. (2003). TANSEN: A query-by-humming based music retrieval system,". *National Conference on Communications.* Chennai.

Ramona, M., Richard, G., & David, B. (2008). Vocal detection in music with support vector machines. *IEEE International Conference on Audio, Speech, and Signal Processing.* Las Vegas.

Rao, P., & Shandilya, S. (2004). On the detection of melodic pitch in a percussive background. *Journal of the Audio Engineering Society , 50* (4), 378-390.

Regnier, L., & Peeters, G. (2009). Singing voice detection in music tracks using direct voice vibrato detection. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* Taipei.

Rocamora, M., & Herrera, P. (2007). Comparing audio descriptors for singing voice detection in music audio files. *Brazilian Symposium on Computer Music.* Sao Paulo.

Rynannen, M., & Klapuri, A. (2006). Transcription of the singing melody in polyphonic music. *International Conference on Music Information Retrieval.* Victoria.

Rynnanen, M., & Klapuri, A. (2008). *Audio melody extraction for MIREX 2008.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2008:Audio_Melody_Extraction_Results

Rynnanen, M., & Klapuri, A. (2008). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal , 32* (3), 72-86.

Rynnanen, M., & Klapuri, A. (2006). *Transcription of the singing melody in polyphonic music (MIREX 2006).* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2006:Audio_Melody_Extraction_Results

Scheirer, E. (2000). Music Listening Systems. *Ph.D. Dissertation* . Boston: Massachusetts Institute of Technology.

Schroeder, M. (1968). Period histogram and product spectrum: New methods for fundamental frequency measurement. *Journal of the Acoustical Society of America , 43*, 829-834.

Secrest, B., & Doddington, G. (1982). Postprocessing techniques for voice pitch trackers. *IEEE International Conference on Audio, Speech, and Signal Processing*, (pp. 172-175). Dallas.

Selfridge-Field, E. (1998). Conceptual and Representational Issues in Melodic Comparison. *Melodic Comparison: Concepts, Procedures, and Applications, Computing in Musicology , 11*, 73-100.

Serra, X. (1997). Music sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Picialli, & G. De Poli (Eds.), *Musical Signal Processing.* Swets & Zeitlinger.

Shenoy, A., Wu, Y., & Wang, Y. (2005). Singing voice detection for karaoke application. *Visual Communications and Image Processing.* Beijing.

Slaney, M. (1998). *The auditory toolbox.* Interval Research Corporation.

SourceForge. *Qwt Library*. Retrieved from http://qwt.sourceforge.net

Subramanian, M. (2002). An analysis of gamakams using the computer. *Sangeet Natak , 37*, 26-47.

Sundberg, J. (1987). A rhapsody on perception. In *The Science of the Singing Voice.* Northern Illinois University Press.

Sundberg, J. (1987). Perception of the singing voice. In J. Sundberg, *The Science of the Singing Voice.* Northern Illinois University Press.

Synovate. (2010). *The Indian Music Consumer.* Mumbai: Nokia Music Connects 2010.

Tachibana, H., Ono, T., Ono, N., & Sagayama, S. (2010). *Extended abstract for audio melody extraction in MIREX 2010.* Retrieved from MIREX Audio Melody Extraction Contest: http://nema.lis.illinois.edu/nema_out/mirex2010/results/ame/adc04/index.html

Tachibana, H., Ono, T., Ono, N., & Sagayama, S. (2009). *Melody extraction in music audio signals by melodic component enhancement and pitch tracking.* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2009:Audio_Melody_Extraction_Results

Talkin, D. (1995). A robust algorithm for pitch tracking. In *Speech Coding and Synthesis.* Amsterdam: Elsevier Science.

Tolonen, T., & Karjalainen, M. (2000). A computationally efficient multipitch model. *IEEE Transactions on Speech and Audio Processing , 8* (6), 708-716.

Tzanetakis, G. (2004). Song-specific bootstrapping of singing voice structure. *IEEE International Conference on Multimedia and Expo (ICME).* Taipei.

Tzanetakis, G., & Cook, P. (2002). Music genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing , 10* (5), 293-302.

Tzanetakis, G., Kapur, A., Schloss, A., & Wright, M. (2007). Computational Enthnomusicology. *Journal of Interdisciplinary Music Studies , 1* (2), 1-24.

Vallet, F., & McKinney, M. (2007). Perceptual constraints for automatic vocal detection in music recordings. *Conference on Interdisciplinary Musicology.* Tallin.

Van der Meer, W., & Rao, S. (1998). *AUTRIM (AUtomated TRanscription of Indian Music).* Retrieved August 09, 2007, from http://musicology.nl/WM/research/AUTRIM.html

Van Hemert, J. (1988). Different time models in pitch tracking. *Speech'88, 7th FASE Symposium*, (pp. 113-120). Edinburgh.

VCreateLogic. *GCF-A custom component framework.* Retrieved from http://www.vcreatelogic.com/products/gcf/

Wang, Y., & Zhang, B. (2008). Application-specific music transcription for tutoring. *IEEE Multimedia , 15* (3), pp. 70-74.

Wells, J., & Murphy, D. (2010). A comparative evaluation of techniques for single-frame discrimination of non-stationary sinusoids. *IEEE Transactions on Audio, Speech and Language Processing , 18* (3), 498-508.

Wendelboe, M. (2009). *Using OQSTFT and a modified SHS to detect the melody in polyphonic music (MIREX 2009).* Retrieved from MIREX Audio Melody Extraction Contest: http://www.music-ir.org/mirex/wiki/2009:Audio_Melody_Extraction_Results

Wikipedia. *Wikipedia The Free Encyclopedia*. Retrieved October 20, 2010, from Counterpoint: http://en.wikipedia.org/wiki/Counterpoint

Wise, J., Caprio, J., & Parks, T. (1976). Maximum likelihood pitch estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing , ASSP-24* (5), 418-423.

Wolfe, J. *Harmonic singing v/s normal singing*. Retrieved November 5, 2005, from Music Acoustics: http://www.phys.unsw.edu.au/~jw/xoomi.html

Wu, M., Wang, D., & Brown, G. (2003). A multipitch tracking algorithm for noisy speech. *IEEE Transactions on Speech and Audio Processing , 11* (3), 229-241.

Xiao, L., Zhou, J., & Zhang, T. (2008). Using DTW based unsupervised segmentation to improve the vocal part detection in pop music. *IEEE International Conference on Multimedia and Expo (ICME).* Hannover.

Zhang, T. (2003). System and method for automatic singer identification. *IEEE International Conference on Multimedia and Expo (ICME).* Batlimore.

Zhang, Y., & Zhang, C. (2005). Separation of voice and music by harmonic structure modeling. *19th Annual Conference on Neural Information Processing Systems.* Vancouver.

Zhou, R. *Polyphonic transcription VAMP plugin.* Retrieved August 27, 2009, from http://isophonics.net/QMVampPlugins.

# List of Publications


[P1] Rao, V., Gaddipati, P., & Rao, P. (2011). Signal-driven window-length adaptation for sinusoid detection in polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*. (Accepted – June 2011).

[P2] Rao, V., & Rao, P. (2010). Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, *18*(8), 2145–2154.

[P3] Rao, V., Gupta, C., & Rao, P. (2011). Context-aware features for singing voice detection in polyphonic music. *9th International Workshop on Adaptive Multimedia Retrieval*, Barcelona, Spain. (Submitted for review).

[P4] Rao, V., Ramakrishnan, S., & Rao, P. (2009). Singing voice detection in polyphonic music using predominant pitch. *InterSpeech*. Brighton, U.K.

[P5] Rao, V., & Rao, P. (2009). Improving polyphonic melody extraction by dynamic programming-based dual-F0 tracking. *12th International Conference on Digital Audio Effects* (*DAFx*). Como, Italy.

[P6] Rao, V., & Rao, P. (2008). Vocal melody detection in the presence of pitched accompaniment using harmonic matching methods. *11th International Conference on Digital Audio Effects* (*DAFx*). Espoo, Finland.

[P7] Bapat, A., Rao, V., & Rao, P. (2007). Melodic contour extraction for Indian classical vocal music. *Music-AI* (*International Workshop on Artificial Intelligence and Music*) *in IJCAI*. Hyderabad, India.

[P8] Pant, S., Rao, V., & Rao, P. (2010). A melody detection user interface for polyphonic music. *National Conference on Communication* (*NCC*). Chennai, India.

[P9] Rao, V., Ramakrishnan, S., & Rao, P. (2008). Singing voice detection in north Indian classical music. *National Conference on Communication* (*NCC*). Mumbai, India.

[P10] Rao, V., & Rao, P. (2008). Objective evaluation of a melody extractor for north Indian classical vocal performances. *International Symposium on Frontiers of Research in Speech and Music* (*FRSM*). Kolkata, India.

[P11] Rao, V., & Rao, P. (2007). Vocal trill and glissando thresholds for Indian listeners. *International Symposium on Frontiers of Research in Speech and Music* (*FRSM*). Mysore, India.

[P12] Rao, P., Rao, V., & Pant, S. (2009). A device and method for scoring a singing voice. *Indian Patent Application,* No. 1338/MUM/2009.

[P13] Santosh, N., Ramakrishnan, S., Rao, V., & Rao, P. (2009). Improving singing voice detection in the presence of pitched accompaniment. *National Conference on Communication* (*NCC*). Guwahati, India.

[P14] Rao, V., & Rao, P. (2008 & 2009). Melody extraction using harmonic matching. *The Music Information Retrieval Exchange – MIREX 2008 & 2009*. URL: http://www.music-ir.org/mirex/abstracts/2009/RR.pdf

[P15] Rao, V., Pant, S., Bhaskar, M., & Rao, P. (2009). Applications of a semi-automatic melody extraction interface for Indian music. *International Symposium on Frontiers of Research in Speech and Music* (*FRSM*). Gwalior, India.

# Acknowledgements

This thesis is the result of the helpful efforts of many people. Foremost, I am deeply indebted to my advisor, Prof. Preeti Rao. From accepting to guide a student in the relatively untested waters of music signal processing for Indian music to the submission of this thesis, her contribution has been immense. She has been extremely helpful and always made herself available for discussion. The discussions with her were always very insightful and would inspire me further in research. Her constant encouragement and backing helped me keep the research, and myself, on course. It has indeed been a privilege working with her.

I am thankful to Prof. P. C. Pandey and Prof. V. Rajbabu for agreeing to be on my research committee. Their suggestions and remarks at the annual progress seminars were extremely constructive. I would like to thank Prof. Gaël Richard and Prof. Hema Murthy for kindly reviewing my thesis. Their suggestions were very useful in improving the presentation of my research in this thesis. I would also like to thank Dr. Suvarnalata Rao of the NCPA, Mumbai, for providing the musicological perspective over the course of this research and for giving me access to excerpts of multi-track Indian classical music recordings made by the NCPA. Her suggestions at my annual progress seminars and other interactions were invaluable in keeping the research relevant in the musical context.

I would now like to thank all the members (past and present) of the Digital Audio Processing Lab (DAPLAB), IIT Bombay for their direct/indirect assistance in my research. I thank Sachin Pant for developing the melody extraction interface and for his constant coding assistance, especially for the MIREX submissions. I also thank Ashutosh Bapat, S Ramakrishnan, Pradeep Gaddipatti,and Chitralekha Gupta for their research contributions to various aspects of the work on melody extraction. I would like to thank my fellow Ph.D. students in DAPLAB: Pushkar Patwardhan, Pradeep Kumar Puthiyaveetil, Srikanth Joshi, Vaishali Patil and especially Veena Karjigi, for proofreading all my publication submissions and for the interesting discussions. I thank all my colleagues in the lab, with whom I have interacted at some point: Manohar, Bhavik Patel, Bhavik Shah, Sivakumar, Hariharan Subraminanian, Nilesh Tamrakar, Sunil Thapliyal, Ramsingh, Manish Kesarkar, Rishabh Bhargava, Vijay Kartik, N. Santosh, Nipun Dave, Sujeet Kini, Shreyas Belle, Nikhil Bhave, Rohan Shah, K.L. Srinivas, Chitralekha Bhat, Sankalp Gulati and Pranav Jawale. They have

been very helpful and supportive all along. I have been fortunate in having worked alongside such wonderful people.

Finally and most importantly I would like to thank all the members of my family, my mother, wife, brother, sister, father-, mother-, sisters- and brothers-in-law, who have been extremely kind, supportive, helpful, caring and considerate. This thesis would have been impossible without their understanding, continued support, limitless patience, constant encouragement and love. This thesis is dedicated to them. And to my daughter, who is a gleaming ray of sunshine even on the gloomiest of days, I can proudly say “Daddy doesn’t go to school anymore, Shriya!”