Automatic Transcription of the Melody from Polyphonic Music

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität Ilmenau

von Dipl.-Ing. Karin Dressler, geboren am 29. Januar 1978 in Erfurt

Gutachter: Prof. Dr.-Ing. Dr. rer. nat. h. c. mult. Karlheinz Brandenburg
Prof. Dr. rer. nat. habil. Meinard Müller
Dr.-Ing. Emilia Gómez Gutiérrez

Tag der Einreichung: 02.12.2016
Tag der wissenschaftlichen Aussprache: 03.04.2017

urn:nbn:de:gbv:ilm1-2017000136

Für Opa

Abstract

This dissertation addresses the problem of melody detection in polyphonic musical audio. The proposed algorithm uses a bottom-up design consisting of five processing modules: the spectral analysis, a pitch determination algorithm, the tracking of tone objects, the tracking of musical voices including the identification of the melody voice, and MIDI note estimation. Each module leads to a more abstract representation of the audio data and hence to a reduction of information load, which allows a very efficient computation of the melody. Nonetheless, the dataflow is not strictly unidirectional: on several occasions, feedback from higher processing modules controls the processing of low-level modules.

The spectral analysis is based on a technique for the efficient computation of short-time Fourier spectra in different time-frequency resolutions, the so called multi-resolution FFT, which provides the best frequency resolution in the lowest frequency band, and an increasing time resolution in higher frequency bands.

The pitch determination algorithm (PDA) is based on the pair-wise analysis of spectral peaks. The idea of the technique lies in the identification of partials with successive (odd) harmonic numbers and the subsequent subharmonic summation. Additional ratings are introduced to avoid octave errors and to discriminate partials from different audio sources, exploiting clues like harmonicity, timbral smoothness, the appearance of intermediate spectral peaks, and harmonic number. Salient pitches chosen from the resulting pitch spectrogram denote adequate starting points for high-level tone objects.

Although melody detection implies a strong focus on the predominant voice, the proposed tone processing module aims at extracting multiple fundamental frequencies (F0). Current state-of-the-art algorithms promote either the iterative detection of the predominant pitch and its subsequent deletion, or the joint pitch candidate estimation. In this thesis, a novel approach is presented that combines the iterative with the joint method: the long-term spectral envelope of existing tones is used to inhibit spectral peaks of the current analysis frame prior to the pitch estimation. This feedback from the tone processing module ensures that the PDA detects primarily the fundamental frequencies of new tones. At the same time, the joint evaluation of active tones tackles the problem of shared harmonics and helps to uncover octave errors.

In order to identify the melody, the best succession of tones has to be chosen. This thesis describes an efficient computational method for auditory stream segregation that processes a variable number of simultaneous voices. Although no statistical model is implemented, probabilistic relationships that can be observed in melody tone sequences are exploited.

The MIDI note module of the algorithm aims at the identification of the tone's onsets and offsets, as well as the assignment of a discrete tone height according to an equal temperament scale, which does not depend on a predetermined reference frequency. While the MIDI module uses at least some musicological knowledge, all other algorithm parts do not make prior assumptions about the instruments, the music genre, the tuning, or even musical scales. For this reason, the system can be utilized on different kinds of audio.

The presented melody extraction algorithm has been evaluated during the MIREX audio melody extraction task. MIREX stands for Music Information Retrieval Evaluation eXchange. The goal of this exchange is to compare algorithms and systems relevant for the multidisciplinary field of Music Information Retrieval, including symbolic music, audio, and other subdisciplines. The MIREX results show that the proposed algorithm belongs to the state-of-the-art-algorithms, reaching the best overall accuracy in MIREX 2014. Moreover, MIREX results in the multiple F0 detection task show that the proposed tone processing method allows a reliable and very efficient identification of multiple F0s. Another strength of the melody extraction algorithm is its computational efficiency and its potential for real-time processing with a small time latency.

Keywords: melody extraction in polyphonic music, music information retrieval, pitch determination algorithm, multiple fundamental frequency estimation, tone tracking, sound source segregation, timbre estimation in polyphonic audio, onset detection, identification of the melody voice, audio to midi conversion, reference frequency estimation, MIREX audio melody extraction.

Zusammenfassung

Diese Dissertation befasst sich mit dem Problem der Melodiextraktion aus polyphonem musikalischen Audio. Der vorgestellte Algorithmus umfasst ein “bottom-up”-Design und besteht aus fünf Modulen: die Spektralanalyse, der Pitchbestimmungsalgorithmus, das Verfolgen von Tonobjekten, das Verfolgen von musikalischen Stimmen sowie die Identifikation der Melodiestimme und die Bestimmung von MIDI-Noten. Jedes dieser Module liefert eine abstraktere Darstellung der Audiodaten und führt somit zu einer Reduzierung der Informationsmenge, was eine effiziente Extraktion der Melodie erlaubt. Allerdings ist der Datenstrom nicht unidirektional – bei verschiedenen Gelegenheiten steuert Feedback von höheren Verarbeitungsmodulen die Verarbeitung von vorangestellten Modulen.

Die Spektralanalyse basiert auf einer Technik zur effizienten Berechnung von Kurzzeit-Fourier-Spektren in verschiedenen Zeit-Frequenz-Auflösungen, die sogenannte “multi-resolution FFT”, welche die beste Frequenzauflösung im unteren Frequenzbereich bereitstellt und eine größere Zeitauflösung in den höheren Frequenzbereichen.

Der Pitchbestimmungsalgorithmus basiert auf der paarweisen Analyse von spektralen Maxima. Die Idee dieser Technik liegt in der Identifikation von Teiltönen mit einer aufeinander folgenden (ungeraden) harmonischen Nummer und ihrer anschließenden subharmonischen Summation. Zusätzliche Bewertungen werden eingeführt, um Oktavfehler zu vermeiden und Teiltöne von verschiedenen Audioquellen zu unterscheiden. Dabei werden verschiedene Hinweise verwendet, wie zum Beispiel die Harmonizität, die Glattheit der spektralen Hüllkurve, das Auftreten von zwischenliegenden spektralen Maxima und die harmonische Nummer. Hervortretende Pitches werden aus dem entstehenden Pitch-Spektrogramm ausgewählt. Sie bilden die Startpunkte für Tonobjekte.

Obwohl die Melodieextraktion einen starken Fokus auf die vorherrschende Stimme voraussetzt, zielt das Tonverarbeitungsmodul auf eine Extraktion von allen auftretenden Grundfrequenzen (F0) ab. Die meisten Ton-Extraktions-Algorithmen, welche dem neuesten Stand der Technik entsprechen, verwenden entweder die iterative Detektion des vorherrschenden Pitches und seiner anschließenden Löschung oder die gleichzeitige Bestimmung aller Pitch-Kandidaten. In dieser Dissertation wird ein neuer Ansatz vorgestellt, welcher die iterative Methode mit der gleichzeitigen Bestimmung der Pitch-Kandidaten verbindet: Die spektrale Hüllkurve von aktiven Tönen wird benutzt, um spektrale Maxima des aktuellen Analysezeitfensters vor der Pitchbestimmung zu unterdrücken. Dieses Feedback der Tonverarbeitungseinheit stellt sicher, dass der Pitchbestimmungsalgorithmus hauptsächlich die Grundfrequenzen von neuen Tönen extrahiert. Die gleichzeitige Auswertung von aktiven Tonobjekten ermöglicht es, Oktavfehler zu entdecken und das Problem von gemeinsamen Harmonischen anzugehen.

Um die Melodiestimme zu identifizieren, muss die beste Abfolge von Tönen ausgewählt werden. Diese Dissertation beschreibt eine effiziente Methode für die automatische Segregation von sogenannten auditiven Klangströmen. Dabei verarbeitet die vorgestellte Methode eine variable Anzahl von gleichzeitigen Stimmen. Auch wenn kein statistisches Model implementiert wurde, werden probabilistische Zusammenhänge, die in Abfolgen von Melodietönen beobachtet werden können, verwendet.

Das MIDI-Noten-Modul des Algorithmus verfolgt die Identifikation von Tonanfang und Tonende sowie die Festlegung einer diskreten Tonhöhe. Die Festlegung der Tonhöhe geschieht gemäß einer temperierten Stimmung, welche die Skala in 12 Halbtonschritte mit dem exakt gleichen Abstand einteilt. Eine bestimmte Referenzfrequenz wird dabei nicht vorausgesetzt. Während das MIDI-Noten-Modul wenigstens einige musikwissenschaftliche Kenntnisse verarbeitet, gibt es bei allen anderen Teilen des Algorithmus keine vorausgehenden Annahmen zu Instrumenten, zum Musikgenre, zur Stimmung oder zu den verwendeten Skalen. Aus diesem Grund kann das System für verschiedenste Arten von Audio verwendet werden.

Der vorgestellte Melodieextraktionsalgorithmus wurde während des MIREX “audio melody extraction task” evaluiert. MIREX steht für Music Information Retrieval Evaluation eXchange. Das Ziel dieses Austausches ist der Vergleich von Algorithmen und Systemen aus dem multidisziplinären Feld des “Music Information Retrieval”. Die Resultate zeigen, dass der Algorithmus zum Stand der Technik gehört – es wurde die beste Gesamtgenauigkeit der im Jahr 2014 ausgewerteten Algorithmen erreicht. Eine weitere Stärke des Melodieextraktions-Algorithmus ist seine kurze Rechenzeit und sein Potenzial für eine Bestimmung der Melodie in Echtzeit mit einer sehr geringen Zeitverzögerung. Außerdem zeigen Resultate im MIREX “multiple F0 fundamental frequency estimation and tracking task”, dass der vorgestellte Algorithmus eine zuverlässige und sehr effiziente Bestimmung von mehreren Grundfrequenzen erlaubt.

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

So what is this thesis about? I have answered this question many times during the last years, and my answer is as follows: I teach the computer to extract the melody from a music recording. Imagine you have a sound file containing some music. A computer program reads your file and analyzes the content. It detects the melody tones among all the other sounds and writes them to another file, for example a MIDI file[1].

The nice thing about my thesis topic is that it can be explained in very simple words. Most people quickly understand what I am trying to do. If people have at least some musical or technical background, very soon we are discussing first ideas about the solution of the problem. I had the same feeling when I – together with two fellow students – started a research project on melody extraction for the curriculum of my engineering degree. The problem is well defined, it can be solved! That is what we thought. I have to admit, we did not even get close to extracting melody tones after one year of work. Even though the resulting algorithm was not a success story, I was hooked; and I was still quite optimistic about a final solution. In my opinion, three or four years of serious work should do the trick.

I was ready to use any tool that might be useful to solve the problem. This is the advantage of the engineering approach which aims at good results in the first place – and not so much at a shiny scientific formula. For example, in this first research project, we were supposed to use image processing algorithms for melody extraction. You think this is a ridiculous idea? Well, I agree. It did not work. So on to the next set of tools. Sure, an impressive formula would not be bad, especially if you are writing a PhD thesis, but if an endless series of "if, then, else" gives the better result, I would not hesitate to use it. Why agonize about wave mechanics in the cochlea, this spiraled, hollow, conical chamber of bone in the human inner ear, in which the sound waves propagate, just to be detected by our sensory cells? Why even bother with the firing rates of the hair cells on the basilar membrane? I wanted to extract the melody, not to build a model of the human auditory system!

[1] MIDI (Musical Instrument Digital Interface) is an electronic musical instrument industry specification that enables a wide variety of digital musical instruments, computers and other related devices to connect and communicate with one another.

After writing that last sentence, I confess: I have read more books about psychoacoustics and the perception of sound than about any other topic. Although I never intended to build a model of the human auditory system, I found myself searching for inspiration in psychology books. Why? Because it works. During the implementation of my algorithm, I noticed that the best solutions often resembled the human auditory system. Well, there is a simple reason for this: the vast majority of music is deliberately composed in such a way so that it can be understood by humans.

At first, the consumption of unfamiliar music is a hard piece of work for the brain and may result in a serious headache. But very soon, without ever getting formal instructions about scales, harmony or rhythm, the brain is able to form unconscious expectations on how the music will enfold. Good music finds a balance between the fulfillment of expectations and elements of surprise. Surprise is an important ingredient of music, because as David Huron (2006) states: Minds need to reach, not simply grasp. Brains need to be challenged, not simply pampered.

Music offers a challenging task to the brain, which may be described as follows: decompose a complex mixture of different sound sources and track them over time.

Imagine how the multitude of instruments in a symphony orchestra fuses into a manageable number of musical voices. Yet on the contrary, imagine how the alternating low and high tones of a yodeling voice segregate into two voices. It becomes clear that the perception of music is steered by the characteristics of the human auditory system.

If you listen to the recording of a Metal band, the lead voice will be covered by much musical noise – drums, a bass guitar, a few electric guitars... In addition, the sound is alienated with a number of sound effects. Putting it briefly, in terms of melody recognition the signal to noise ratio is bad. Still, you are able to make out the lead voice, whether it carries a recognizable melody or not.

We have seen a tremendous progress in music transcription during the last years, but for all that, a computer algorithm that reliably extracts the predominant voice from Rock music does not exist. This thesis describes one of the mile stone algorithms for melody extraction. Its performance is not even close to the listening abilities of humans. In a case like this, researchers say: "There is future work to do." But it is worth it! Because listening to music is not only a pleasure, it is also an exercise in survival!

1.1. Motivation, Scope, and Applications

It is clear that a computer algorithm for melody extraction is neither a cure against cancer nor does it help to maintain world peace. My personal motivation for this topic is easily explained. I was fascinated by the fact that the task of melody extraction, which is easily accomplished by humans, proves to be a real challenge for computers. However, my mother has other concerns. "Couldn't you do research on speech recognition?" she asked me once. This question actually translates into: "And how on earth are you going to sell a product and earn lots of money?"

This is the actual dilemma of music information retrieval (MIR). Downie (2003) revisits the facets of music information and underlines the importance to extract, analyze, organize and retrieve this data. Yet, while very serious MIR scientists fiddle about their complex and rather ingenious algorithms, the input to those algorithms is still only music. While language is recognized as one of the essential ingredients of what makes us human, music is often perceived as an unimportant by-product of evolution (Pinker, 1998). And besides, it is no secret that music is fun. The truth is, most MIR researchers do not only love their research, they also play instruments, like to sing, or at least have a very profound opinion on what is or is not good music.

But how can one be serious about research if it comes in a bundle with something so enjoyable? Since Darwin, it is common practice to find some evolutionary excuse for the existence of every peculiarity of life. This might be the reason why Mithen (2006) saw the need for an evolutionary history of music. In his book "The Singing Neanderthals" he presents various reasons for the importance of music: it acts as a mean of communication and social bonding, it may be used for religious statements. And yes, music can also be seen as a product of sexual selection – anyone, who has the time to sing and dance, and still manages survival, must be a great companion! Well, the problem is, even if there is evolutionary evidence that music has a right to existence, it does not provide an answer to the money question. That is why I should proceed and tell you something about the applications that can be built with a melody extraction algorithm.

The most popular example for an application based upon melody data is *query by humming* (QbH). This application helps you to find a song in a large database by singing or humming its melody. The advantage of this search engine is apparent, because very often the melody is the only thing about a song that can be reproduced from memory. Sometimes, people argue that this application already exists. Well, it is true – to some extent. The problem is that the database to be searched has to be assembled by hand. A QbH demo application developed by Fraunhofer[2] contains about 6000 songs, which is not a high number compared to the 20 million songs that can be purchased on Apple's iTunes[3] platform. A QbH application only makes sense, if the database covers many pieces. Yet, once computer algorithms automatically extract the melodies from polyphonic music in a sufficient quality, query by humming will be an everyday tool.

If such a database of melodies existed, its applications would be numerous. The

[2] musicline QbH: `http://www.musicline.de/de/melodiesuche`

[3] iTunes: `http://www.apple.com/de/itunes/`

data could also be used for the detection of cover songs – or the detection of plagiarism (Dittmar et al., 2012b). And there is another application related to the music search problem. Music recommenders are tools to suggest music based on the songs you already like. Most commonly, social data is exploited to make such recommendations. This is a legitimate approach, because if music enthusiasts have similar music preferences, they will usually discover interesting unknown songs in each other's collections. Social recommenders work very well for established music, but they cannot work for music that has just been released. And of course there is a vast amount of very peculiar (or unique) music from the long tail of the world's music corpus that may never amass enough statistical data to allow a recommendation. In this case musical features extracted from the audio could complement the social data and help to characterize a musical piece.

When Mithen describes the evolutionary advantages of music, he emphasizes one aspect that has not yet been addressed by the introduced applications: the joy of making music (together). In ancient times, making music used to be a social event. Today, the production of music in recording studios sets a standard of professional perfectionism that somehow leaves the ordinary human outside the creative process. Our ears are pampered with the crystal clear sound from CDs, where expensive recording technology is used to make every piece of music a spotless listening experience. This may be the reason why many people think that they are not musical and as a consequence they refuse to take part in musical activities (Honing, 2011).

Computer music applications help to bring the joy of active music making back to the people. There are popular releases of the video gaming industry like Guitar Hero[4], Rock Band[5] and the karaoke game SingStar[6] that focus on the fun aspect of music. Some of the games employ MIR-techniques to analyze the attempts at producing music in order to give the user some feedback.

Systems of music education, like play-along CDs or instructional videos for learning an instrument, pursue more serious objectives. The main advantage of these systems is that users can practice with their own musical instruments instead of playing on special game controllers that have only a vague resemblance with real-world instruments.

Play-along CDs feel like a digression from MIR technologies, because they do not actually use any. Then again, this example allows a nice transition to interactive learning software that actually puts the presented melody extraction algorithm to operation. Songs2see[7] is a recent development by the Fraunhofer IDMT, which aims at combining the advantages of educational and gaming software (Dittmar et al., 2012a). It allows the user to play on real instruments, and – by employing MIR-

[4] Guitar Hero: `http://www.guitarhero.com`

[5] Rock Band: `http://www.rockband.com`

[6] Singstar: `http://www.singstar.com`

[7] Songs2See: `http://www.songquito.com/index.php/de/`

technologies for the automatic detection of beat and pitch – the software is able to provide some feedback in terms of rhythm and melody. Users can choose song titles from the provided music collection, but they can also use the software to create their own training material.

Songs2see is surely a fun application and that is exactly what it should be! So once again we face our initial problem: music is a nice social interaction, but will MIR research ever climb to the level of seriousness and respectability? It should, because as I have stated before, the ability of listening to music may be an aid to survival. Consider one of our earlier ancestors wandering through a thick forest:

> The aged man walking beside him barks in his deep rumbling voice, telling him how much better the old times were, while his associate sings in her bell-like voice the endless lament about the old-fashioned furnishing in their dark cave. On top of it all, the baby has been wailing for two hours straight. But he didn't hear it all. Nonetheless, his senses were focused. There. Again! The cracking sound came from behind them. While not exactly matching the steady rhythm of their footsteps the sound was too consistent to be purely random. Something was following them!

What do we learn from this example? Well, if you really want to build a cool application for speech recognition, it should better be good at listening to music, too. Because a sound seldom comes alone!

1.2. Outline of Dissertation

The dissertation has eight chapters. After this introduction, Chapter 2 presents an overview of the field of melody detection in polyphonic music, highlighting the milestone developments as well as state-of-the-art systems for melody extraction.

The two introductory chapters are followed by five technical chapters, in which the implementation of the system is described in detail. Thereby, the organization of the technical chapters reflects the processing modules of the proposed algorithm in so far that each processing module comprises a different chapter (see also the flowchart presented in Figure 1.1). Hence, Chapter 3 describes the spectral analysis front-end of the system, the so called multi-resolution FFT. Chapter 4 introduces a novel algorithm for the detection of the predominant pitch, whereas Chapter 5 proposes an approach for the detection and the tracking of multiple fundamental frequencies (i.e. tone objects). It also elaborates a method for the generation of MIDI note output, comprising the topics of onset and offset detection, the estimation of the tuning frequency, and the assignment of a discrete tone height to tones with a varying frequency. Chapter 6 presents an efficient computational method for auditory stream

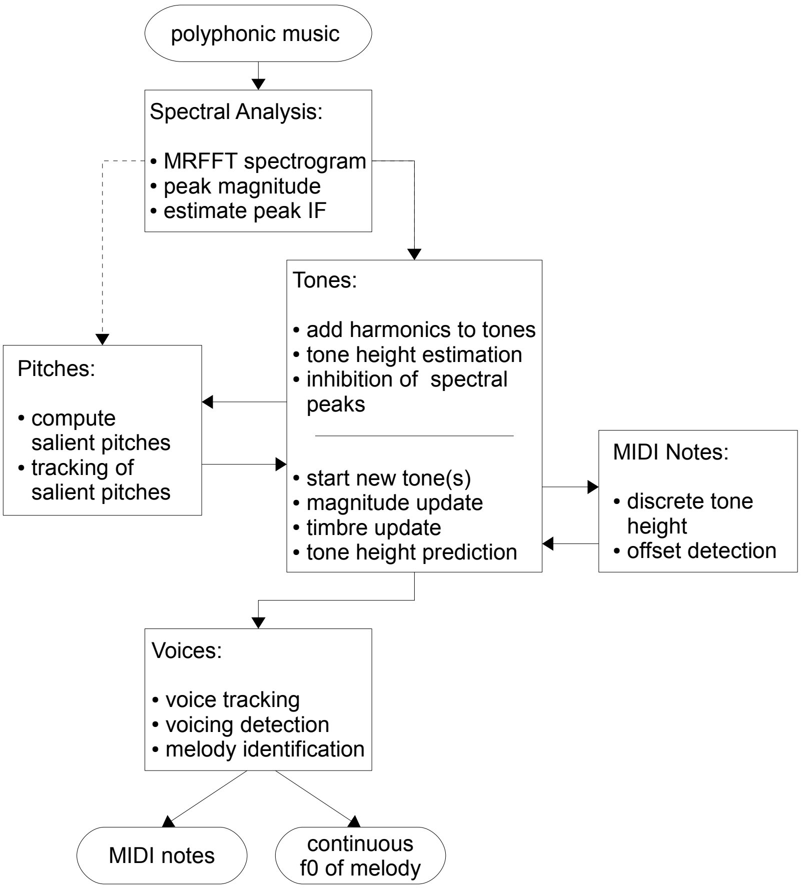


Figure 1.1. Algorithm Overview

segregation. Moreover, it describes how the melody voice is chosen from the range of the detected voices. Following the technical parts of the thesis, Chapter 7 gives a detailed evaluation of the algorithm by reviewing the results of the MIREX audio melody extraction task and the MIREX multiple fundamental frequency estimation and tracking task. Finally, the conclusion in Chapter 8 summarizes the project and suggests possible directions for future research.

Chapter 2.

State of the Art

2.1. Introduction

Two decades ago, no one believed that the extraction of the melody from polyphonic music was possible. Just until in 1999, Goto and Hayamizu introduced the PreFest[1] algorithm that attempted to do exactly this – extract the melody from real-world CD recordings (Goto and Hayamizu, 1999). Their idea was revolutionary because of its simplicity. As the melody is often the predominant voice in the sound mixture, it should be sufficient to detect the fundamental frequency of the strongest periodic sound.

Yet, even under the given premise, melody extraction is not a straight forward task, because the target signal is usually accompanied by lots of additional noise. Soon it became clear that pitch estimators which were designed for the analysis of speech or monophonic instrument samples are not up to the challenging task. The algorithms had to be adapted to meet the new requirements, and beyond that new strategies were developed, often shifting the research focus to previously disregarded topics.

During the first years encouraging results could be presented in the previously non-existent research area and the accuracy of melody extraction algorithms increased rapidly. However, the advancement has slowed down noticeably and seemingly has come to a halt. Have researchers abandoned this research topic? Or is the slow development already the sign of a glass ceiling effect – an inherent upper limit for the melody extraction accuracy?

Well, the idea of the melody being the predominant voice has fueled the initial research, but it also sets restrictions to the type of music that can be analyzed successfully. Especially in instrumental music, the melody voice is often not clearly predominant, as we find instruments playing together with a comparable loudness. If (like in most current approaches) the magnitude is the most important feature to identify the melody tones, it becomes very difficult to distinguish the melody from the accompaniment.

[1] PreFest stands for predominant F0 estimation.

Researchers are currently working hard to overcome this problem. In this chapter, we will have a closer look at the latest developments.

2.2. Current Approaches

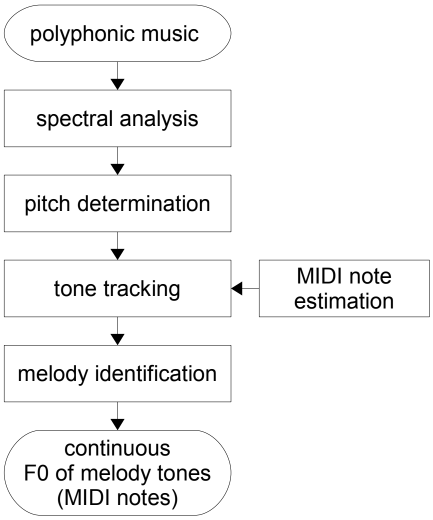


Figure 2.1. Typical Processing Modules of Melody Extraction Algorithms

The flowchart in Figure 2.1 shows the typical signal processing modules of a melody extraction system: spectral analysis, pitch estimation, tone tracking, melody identification, and MIDI note estimation. Actually, only the melody identification itself can be exclusively linked to the melody extraction research – other processing steps are well established problems in other research areas, like for example the modeling of the human auditory system, speech analysis, instrument identification or music transcription.

Of course researchers dealing with melody extraction benefit from the variety of published solutions. The most interesting question is, however, where individual researchers have detected the biggest potential to improve the algorithm performance. Their decision for or against existent techniques, the development of new techniques and a close look at the current state of the art reveals interesting details and at the same time uncovers major trends in the melody extraction research.

As there are many potential combinations of different techniques we will outline the state of the art for each processing module individually in the following sections.

2.2.1. Spectral Analysis

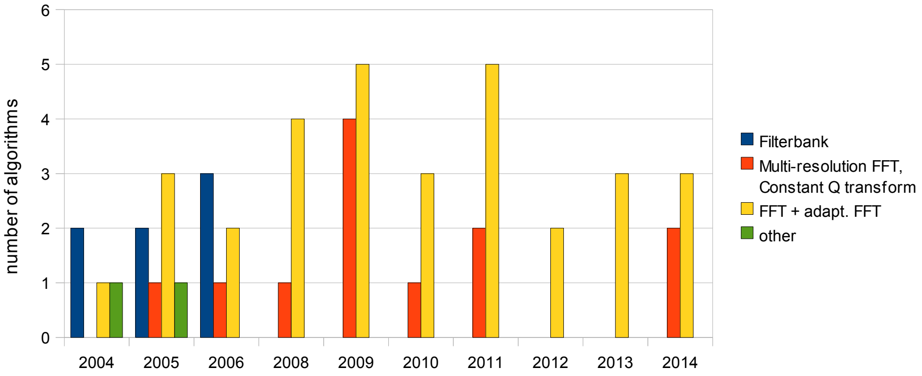


Figure 2.2. MIREX trends for the spectral analysis front-end

The aim of the spectral analysis is to transform the audio signal from the time domain to the frequency domain, which facilitates the identification of distinct sound sources in the audio input: Melody tones are required to have a pitch which allows the ordering of sounds on a frequency-related scale. A common attribute of pitched sounds (in melodies) is that they consist of sinusoidal partials. Hence, the most relevant melody information can be found in the deterministic components of the audio signal, which often can be identified as spectral peaks in the frequency domain.

Figure 2.2 shows an overview of the spectral analysis front-ends of algorithms submitted to the MIREX audio melody extraction task[2]. Although there are many possible combinations of techniques, most spectral analysis front-ends can be assigned to one of three different approaches: a filter bank implementation, a fast Fourier transform (FFT), or a short-time Fourier transform in different time-frequency resolutions, including the constant Q transform and the multi-resolution FFT.

Typically, algorithms implementing a filter bank simulate the mechanical frequency-selectivity of the basilar membrane (de Cheveigné and Kawahara, 1999; Paiva et al., 2005; Heinz, 2006; Klapuri, 2008). For this reason, most filter bank approaches are a closer model of human auditory perception than for example the short-time Fourier transform, in which the time-frequency resolution remains constant and is primarily chosen with regard to the desired application.

The multi-resolution FFT (MR FFT) as well as the constant Q transform are based on the short-time Fourier transform (STFT), yet they roughly mimic the human

[2] The information about “MIREX trends” was extracted from the published extended abstracts on the MIREX web site (and personal communication with the authors). As there are some algorithms for which no abstract was published and some extended abstracts do not contain all necessary information, the picture remains incomplete. Nonetheless, we believe that even with the available data certain trends can be shown.

auditory system: the spectral resolution is better at lower frequencies, whereas the temporal resolution improves at higher frequencies. Essentially, the MR FFT is equivalent to the calculation of the FFT using different analysis window lengths. An efficient implementation of this idea is presented in (Dressler, 2006). The constant Q transform can be seen as a series of logarithmically spaced filters having a band width that is usually proportional to the center frequency of the filter. Its calculation can be accomplished efficiently by the convolution of the FFT spectrum with window kernels in the frequency domain (Brown and Puckette, 1992). The constant Q transform is only rarely used in melody extraction systems, but examples can be found in (Tachibana et al., 2010) and (Chien et al., 2012).

It can be noted from Figure 2.2 that most recent melody extraction algorithms, which have been submitted to MIREX, apply the FFT as a spectral analysis front-end (Pertusa and Iñesta, 2008; Yeh et al., 2010; Emiya et al., 2010; Salamon, 2013; Song et al., 2014). In fact, there has not been a single system utilizing a filter bank implementation after the year 2006.

There is evidence that the choice of the spectral analysis front-end has not such a marked influence on the overall accuracy of the melody extraction. Klapuri (2008) showed that a system for multiple fundamental frequency estimation based on the FFT performed comparable to a previous algorithm implementing a complex auditory model presented in (Klapuri, 2003b). However, Klapuri states that the auditory model-based method is better in utilizing the higher order overtones of a harmonic sound and has a small advantage in the processing of bandlimited signals or signals, in which parts of the spectrum are not usable due to interference. He attributes this advantage to the half-wave rectification step, which is difficult to reproduce in pure time or frequency domain methods. Admittedly, the effect is barely noticeable in the analysis of real-world music.

Salamon and Gómez (2011) submitted two algorithms to the MIREX audio melody extraction task in 2011. The systems were similar, apart from the spectral analysis front-end: the first algorithm utilized an FFT, the second implemented the multi-resolution FFT (Dressler, 2006). The MIREX results show that the performance gain due to the multi-resolution approach was not significant, as the increase of the overall accuracy was only 0.7 percent.

If the type of the spectral analysis front-end is not crucial for the melody extraction results, it is not surprising that the FFT becomes the method of choice. It is a standard tool for spectral analysis – readily available and easy to apply. Yet, the main reason for its application can be seen in its computational efficiency. And the efficiency goes beyond the spectral decomposition of the audio signal, as the short-time Fourier spectrogram provides easily accessible information about the magnitude and the instantaneous frequency (IF) of the sinusoidal partials.

In summary, most state-of-the-art algorithms employ the FFT as means to spectral analysis, which is most often followed by the identification of salient spectral peaks and the estimation of their instantaneous frequencies, in order to improve upon the limited frequency resolution of the spectrogram. Several algorithms for IF estimation are compared in (Marchand and Lagrange, 2006).

Moreover, it can be noted that many melody extraction systems include a magnitude weighting step which reestimates the importance of the spectral components. The weighting may be performed either in the time or in the spectral domain. In the examined algorithms, two different motivations for the weighting can be discerned: the weighting may be perceptually motivated and mimic the sensitivity of humans to signals of different frequencies according to equal-loudness contours (Heinz, 2006; Paiva, 2006; Salamon and Gómez, 2012), or, the weighting aims to level out magnitude differences between spectral peaks independent of their frequency. The latter procedure is often referred to as spectral whitening. Hereby, one simple possibility is taking the logarithm of the Fourier magnitudes. A more advanced example for spectral whitening in the frequency domain is given in (Klapuri, 2006).

Although the trend goes clearly in the direction of FFT-based approaches, the inclusion of more demanding musical pieces in the MIREX evaluation may again turn the research focus to alternative spectral analysis front-ends.

2.2.2. Pitch Estimation

Traditionally, pitch estimation has been discussed in close relation to human perception (de Cheveigné, 2005). One central aspect of the discussion is the question, whether the pitch estimator (which detects the perceived tone height of complex sounds) implements a pattern matching approach in the spectral domain or whether the pitch is derived from purely temporal cues. Pitch phenomena observed in experiments on human hearing give evidence for both mechanisms (Moore, 2003; de Cheveigné, 2005). For this reason, it is not very surprising that both approaches can be found in melody extraction systems.

Even though the melody is usually the predominant pitch in the audio mixture, pitch extraction methods which are designed to work with monophonic audio often fail to produce satisfactory results with music, as the algorithm has to be very robust against interference from simultaneous sounds. Neither a simple autocorrelation of the audio waveform nor a harmonic sieve used in the spectral domain would give satisfactory results if applied to music.

Temporal approaches suitable for the processing of polyphonic music are characterized by the band-wise processing of the audio signal: First, the signal is analyzed by a filter bank. Then, the subband signals are compressed, half-wave rectified and low-pass filtered. The predominant signal period is detected in each frequency channel: the period detection within a frequency band may be conducted by an autocorrelation function (Klapuri, 2003a; Paiva et al., 2005), a difference function (de Cheveigné and Kawahara, 2002), or by the application of the STFT (Klapuri, 2008). Finally, the period information of the distinct frequency channels is combined to obtain a measure of the pitch salience.

In the spectral approach, the pitch detection is performed using the spectral representation of the audio signal. Popular methods include the PreFEst algorithm developed by Goto (2004) and the subharmonic summation algorithm proposed by Hermes (1988). Yet certainly, there is a great variety of other solutions and refinements, often in combination with standard approaches. Rao and Rao (2010) use a combination of a harmonic matching algorithm and the two-way-mismatch-error method, as originally described in (Maher and Beauchamp, 1994). Chien et al. (2012) evaluate the likelihood of each pitch candidate by comparing the observed constant-Q spectrum with a set of (synthesized monophonic) vocal spectrum examples with the same hypothesized pitch and a scaled loudness. Another possibility is the use of probabilistic methods: Poliner and Ellis (2006) implemented a support vector machine to classify the melody directly from the spectral components; Durrieu et al. (2010) computed the best non-negative linear combination of spectral combs of all possible notes, evaluating also the smoothness of timbre between analysis frames.

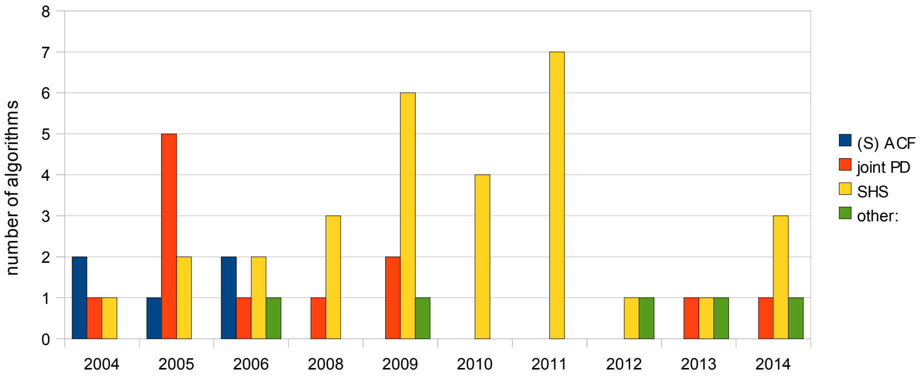


Figure 2.3. MIREX trends for pitch determination algorithms

Figure 2.3 shows the latest trends for the algorithms submitted to MIREX. At present, the most popular method for pitch estimation in melody extraction systems is based on subharmonic summation (SHS), which was used for example in (Yeh et al., 2012; Ikemiya et al., 2014; Dressler, 2014b). Another technique is the joint pitch determination, in which all partials are evaluated at once to determine the most salient pitches from the audio signal. This method is very popular in multiple F0 estimation algorithms, but some melody extraction algorithms use it as well. One example is the approach by Song et al. (2014), who implement a joint pitch estimation based on a Bayesian network.

We can also see a notable decline of pitch estimators using the (summary) autocorrelation function (ACF). After reading the previous section, this observation does not come as a surprise, as it is a consequence of the utilized spectral analysis front-end: since the FFT is preferred over filter-bank implementations, techniques based on the Fourier spectrum are the most obvious pitch estimators to use.

Of course the technical solutions found in melody extraction systems cannot be compared with the complex auditory models implemented by researchers who try to mimic the human auditory system. The current solutions submitted to MIREX neither simulate wave mechanics in the cochlea nor reproduce the firing pattern of the hair cells. So trends in melody extraction do not necessarily answer the question on how the pitch is determined by the human auditory system. Moreover, the latest MIREX results show that the sole extraction of the predominant pitch is not sufficient to build a reliable melody extraction system. Only recently, researchers have begun to exploit more features describing melody tones – some of them are closely related to the timbre of sounds. The deduction of timbre information from polyphonic audio is a challenging problem in itself, because there is usually no prior knowledge about the sound sources.

Algorithms which are designed explicitly for the estimation of multiple fundamental frequencies exploit the spectral structure of musical sounds to address the problem of shared harmonics. For this reason it is worthwhile to have a look at the latest developments in this closely related research area.

2.2.3. Multiple Fundamental Frequency Estimation

The output from pitch estimation algorithms, designed for the determination of the predominant periodic sound, is not a one-to-one representation of existing musical notes. Rather, corresponding pitch magnitudes can be seen as probabilities of perceiving a predominant pitch at a certain frequency. In order to retrieve other (weaker) tones, the effects of the predominant tone have to be factored out, because considerable pitch strengths occur in the pitch spectrogram at harmonic or subharmonic frequencies of its fundamental frequency (F0), or in combination with other periodic sound sources.

For the detection of multiple pitches two approaches prevail: the iterative detection of the predominant pitch and its subsequent deletion (de Cheveigné and Kawahara, 1999; Klapuri, 2003a; Duan et al., 2009), and the joint pitch candidate evaluation (Pertusa and Iñesta, 2008; Yeh et al., 2010; Emiya et al., 2010; Benetos et al., 2013; Recoskie, 2014).

The iterative approach is characterized by the iterative detection and the subsequent deletion of the predominant periodic tone. For the successful elimination of the tone, its approximate spectral envelope must be determined. To achieve this, the smoothness of the spectral envelope is evaluated. Thereby, regularity constraints may be applied in the frequency dimension (comparing the magnitudes between harmonic neighbors), or in the time dimension, as abrupt changes in the harmonic's amplitude are not expected.

In the joint estimation, a certain number of promising pitch candidates is selected from a pitch spectrogram in order to reduce the number of possible note combinations. Then, all possible combinations of candidates are evaluated by jointly partitioning the harmonic magnitudes among the candidates for each combination. Of course, the supposed smoothness of the spectral envelope is also evaluated during this procedure. The set of tones that explains best the spectral data is finally selected.

The joint estimation method is a very recent technique and seems to be a promising approach with a lot of potential. Many of the current joint estimation algorithms use spectrogram factorization techniques like non-negative matrix deconvolution (Bertin et al., 2010; Durrieu, 2010; Recoskie, 2014; Müller, 2015) and probabilistic latent component analysis (Benetos et al., 2013). At present, the advantage over the iterative estimation-cancelation-method is not pronounced (Klapuri, 2006, 2008). Klapuri found that the joint cancelation method performs slightly better in high polyphonies, but of course the high computational complexity of the method has also to be taken into account.

The state-of-the-art algorithm by Elowsson and Friberg (2014) broke the glass ceiling that was observed in recent years in the MIREX multiple F0 estimation task (Benetos et al., 2012). Elowsson and Friberg implemented a polyphonic transcription with a deep layered learning approach, which is a machine learning algorithm that models a hierarchy of high-level data abstractions by using model architectures composed of multiple non-linear transformations. One layer of data that they are seeking to discern is that of fundamental frequency in music and they do so by using a layer-wise decomposition of the perceptual phenomena.

2.2.4. Tone Tracking

The formation of tone objects is often seen as a line tracking problem in the pitch spectrogram – the instances of salient pitches in distinct analysis frames have to be linked in order to form continuous tones[3].

Another possibility to build continuous tone objects is to track partials in the Fourier spectrogram, an idea that has first been applied for the analysis and synthesis of speech (McAulay and Quatieri, 1986). The goal is to link spectral peaks of successive

[3] Admittedly, not all melody extraction algorithms implement an intermediate tone tracking stage. There are many algorithms that omit the tone processing in favor of a global melody extraction solution using a probabilistic model to derive a continuous melody pitch line.

analysis frames to recover the evolution of the tone. In the linear prediction model, the next frequency value of the sinusoidal is approximated by a linear combination of past frequencies (Lagrange et al., 2003). Serra (1989) introduced a partial tracking method which is guided by the previous spectral peak, but also by the identified fundamentals in the current frame. Kereliuk and Depalle (2008) propose a combinatorial hidden Markov model for tracking partials, where the observable spectral peaks are considered as symbols emitted from a set of hidden partial trajectories.

Partial tracking has its merits in the analysis and synthesis of speech and monophonic sounds, but in melody extraction algorithms the tracking is usually performed on the output of the multiple F0-estimation (e.g. the pitch spectrogram). Goto (2004) implemented a multiple-agent architecture, where so-called agents trace temporal trajectories of salient F0 estimates. The tracking agents evaluate frequency proximity and the magnitude of successive pitch candidates, backed up by a reliability control. Another possibility is to track predominant seed pitches forward and backward in time until some expiration condition is reached (Cao et al., 2007; Cancela, 2008; Salamon and Gómez, 2012).

One difficulty in tone tracking is surely the low signal to noise ratio – each sound source is potentially disturbed by any other sound in the audio mixture. Humans are usually not disturbed by the temporary masking of notes, as the brain reconstructs the obscured parts. The so-called spectral restoration of masked tones is reported for durations up to 300 ms (Warren, 1999). Unfortunately, current computer algorithms are quite sensitive about the missing evidence of note continuity. Most algorithms tackle the problem by allowing a short time period without evidence for the note in the pitch spectrogram (Salamon and Gómez, 2012). The occurring gap in the note is usually filled in by interpolation using the closest available pitch values.

2.2.5. Midi Note Estimation

The tracking of pitch contours on a continuous time basis leads us towards the segmentation of individual notes – a transcription task that includes the detection of note onsets and offsets, and the estimation of the discrete tone height (MIDI note numbers). At present, only a few melody extraction algorithms tackle this problem, although there are many applications that would benefit from the explicit identification of musical notes.

The usual way to detect onsets or offsets in monophonic audio is to look for transient regions in the signal, a notion that leads to many different implementations (Bello et al., 2005). The earliest onset detection algorithms used to work directly with the wave form $x(t)$ in the time domain and evaluated the energy of the overall signal or its zero-crossing-rate. Current onset detection algorithms, which are specialized on this task, derive the onsets of tones or percussive sounds directly from the Fourier spectrum, using magnitude-based (spectral flux), phase-based (weighted phase deviation) or complex domain (complex difference) onset detection functions (Bello et al., 2005; Dixon, 2006; Böck et al., 2012b; Müller, 2015). Such approaches can be found in melody extraction algorithms, too, but more often onset detection is based on mid-level representations of the signal. So usually the detection takes place in the pitch spectrogram, where an increase of pitch salience denotes an onset, while a sudden decrease in pitch salience may signal an offset (Chai, 2001).

In the polyphonic context, it is wise to employ additional features, as the magnitude of a tone can often not be estimated reliably. The most obvious supplement is the pitch of the tone: a distinct variation in pitch marks the onset of a new note. In order to use this feature, the audio input is often restricted to those instruments, where individual notes have a stable frequency.

Paiva et al. (2008) aims to distinguish MIDI notes from polyphonic music, including the human singing voice. For the onset detection, amplitude and frequency-based segmentation are combined with a set of filtering rules that take into account glissando, vibrato and other forms of frequency modulation.

Ryynänen and Klapuri (2006) have implemented a method for the detection of singing melodies in polyphonic music, which transcribes MIDI notes and detects the musical key of the recording. Their strategy is noteworthy, because two probabilistic models are implemented: a hidden Markov model (HMM) for note events, which is used to represent note candidates, and a Gaussian mixture model, which describes the musicological aspects of the musical piece and controls the transitions between note candidates.

Benetos and Dixon (2011) perform note estimation as a post-processing step on the output of a multiple F0-estimator. Each pitch is modeled by a two-state HMM, denoting pitch activity/inactivity. The HMMs are trained on orchestral music and piano music for the note tracking task. From the system design it becomes clear that the algorithm does not cope well with frequency modulated tones.

Gómez et al. (2012) implement an automatic transcription for their predominant pitch estimation system. They first estimate the tuning frequency and then implement an iterative strategy for note segmentation and labeling: after the segmentation step the tuning frequency is refined and the nominal note pitches are recomputed based on the new tuning frequency.

Another interesting approach is presented by Laaksonen (2014), who used chord transcriptions to enhance the automatic melody transcription. From already available chord transcriptions he estimated the key of the musical piece, then he used the key and chord information to assign the most probable MIDI note number to the extracted melody notes.

Apart from onset detection there are many problems related to MIDI note estimation that are not addressed by current melody extraction algorithms in a satisfactory manner, for example the estimation of the tuning frequency (Dressler and Streich, 2007), the estimation of the perceived pitch of vibrato notes (Shonle and Horan, 1980; d'Alessandro and Castellengo, 1994; Brown and Vaughn, 1996) as well as the categorical perception of the tone height (Burns and Ward, 1978). Another problem is the accurate segmentation of consecutive notes at the same pitch, as it is difficult to distinguish between note offsets and amplitude modulation (Paiva et al., 2008).

2.2.6. Melody Identification

Melody can be defined as a linear succession of musical tones which is perceived as a single entity. The melody is often the predominant voice in the sound mixture. This means it stands out from the background accompaniment. There are several features that increase the salience of the melody, for example loudness, frequency variation, timbre, and note onset rate.

State-of-the-art melody extraction algorithms mainly exploit two characteristics to identify the melody voice: 1) the predominance of the melody voice in terms of loudness and 2) the smoothness of the melody pitch contour. At present two main algorithm types for the identification of the melody voice can be distinguished:

On the one hand, probabilistic frameworks are used to find the optimal succession of tones. They combine pitch salience values and pitch proximity constraints in a cost function that is evaluated by optimal path finding methods like the hidden Markov Model (HMM) (Hsu et al., 2009), a Markov chain (Chien et al., 2012), dynamic programming (Rao and Rao, 2009), the Viterbi algorithm (Durrieu et al., 2010; Ikemiya et al., 2014), or a Bayesian framework (Song et al., 2014). Probabilistic frameworks often accomplish the tone trajectory forming and the identification of the melody voice at once. The application of a statistical model provides an out of the box solution that simultaneously evaluates different features of the melody voice, as long as they can be expressed mathematically in a cost function or a maximum likelihood function.

On the other hand, there are rule-based approaches that trace multiple pitch contours over time using criteria like salience and pitch proximity in order to link pitch candidates of adjacent analysis frames (Wendelboe, 2009; Joo et al., 2009; Chien et al., 2011; Salamon and Gómez, 2012). Subsequently, a melody line is formed from these tone-like pitch trajectories, using rules that take the necessary precautions to assure a smooth melodic contour.

Of course such a division between algorithm types is rather artificial. It is easy to imagine a system that uses tone trajectories as input for a probabilistic framework, and vice versa a statistical approach can be used to model tones. In fact,

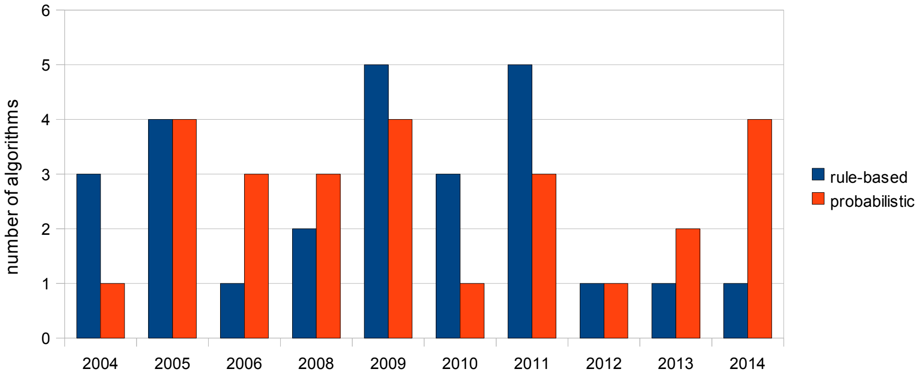


Figure 2.4. MIREX trends for melody identification technique

Ryynänen and Klapuri (2006) have implemented a method for the automatic detection of singing melodies in polyphonic music, where they derive an HMM for note events from fundamental frequencies, their saliences and an accent signal.

From Figure 2.4 it can be seen that both approaches – the probabilistic method as well as the rule-based method – are employed in current melody extraction systems. The reached melody extraction accuracies in the MIREX evaluation do not strongly favor one solution over the other, although at present the highest accuracies are reached by the rule-based methods. Yet, as the overall accuracy depends on multiple factors, it is difficult to deduce which algorithm parts contribute most to the final results.

In any case, whenever there is more than one strong voice in the audio mixture, the identification of the melody voice becomes a challenging problem. Of course, it is not unusual to find another strong voice in real-world music, as a booming bass line is almost mandatory in many music genres. Masataka Goto describes a system for the automatic detection of the melody and bass line for real-world music in (Goto, 2004). Using realistic assumptions about contemporary music, the problem of the concurrent melody and bass line is addressed by intentionally limiting the frequency range for both voices using band pass filters. Rao and Rao (2009) present an approach towards the solution of this problem, giving an example for dynamic programming with dual F0 tracking. The system continuously tracks an ordered pair of two pitches, but it cannot ensure that the two contours will remain faithful to the respective sound sources.

Apart from the mandatory salience and frequency proximity features, Salamon and Gómez (2012) used a set of interesting characteristics that help to identify the melody pitch contour: the mean pitch height of the contour, the duration of the contour and the presence of vibrato. Furthermore, not only feature values are used, but also their standard deviations. Especially the presence of vibrato seems to be a promising feature that clearly enhances the detection of the melody. It can be utilized to focus on the human singing voice (Tachibana et al., 2010), but of course, it can also be applied in a non-destructive way in order to preserve the ability to identify melody tones that are not frequency-modulated.

Chapter 3.

Spectral Analysis

3.1. Introduction

Regarding spectral analysis in melody extraction systems, three main algorithm types can be distinguished: short-time Fourier transform (STFT)-based algorithms, filterbank implementations which mimic the auditory system, and multi-resolution approaches like the constant-Q transform or the multi-resolution FFT.

Most recent melody extraction algorithms apply the fast Fourier transform (FFT) as spectral analysis front-end (Rao and Rao, 2009; Tachibana et al., 2010; Durrieu, 2010; Salamon, 2013; Ikemiya et al., 2014; Song et al., 2014). One reason for this preference is the fact, that the FFT is a standard tool for spectral analysis – computationally efficient, readily available and easy to apply. Moreover, the analysis results between algorithm types do not differ much: Klapuri (2006) showed that a system for multiple F0 estimation based on the short-time Fourier transform performed comparable to a previous algorithm implementing a complex auditory model (Klapuri, 2003b). Salamon and Gómez (2011) submitted two algorithms to the MIREX audio melody extraction task – one with an FFT spectral analysis front-end, one with a multi-resolution FFT front-end – and found that the multi-resolution algorithm performed only 0.7 percent better. This is not a significant difference, indicating that most of the algorithm accuracy is gained at higher processing levels.

Independent of the used spectral analysis method, an important question to be addressed is the choice of the proper time-frequency resolution. To cover signal changes, we have to increase the analysis bandwidth, but at the same time we have to maintain an adequate discrimination of concurrent sounds. If a frequency modulated complex tone is pictured in a spectrogram, a moderate change in frequency is observed for the low harmonics, but vivid dynamics are noted for its higher harmonics, because the amount of frequency modulation is multiplied with the harmonic number. If the signal's frequency is changing in time an accurate measurement of frequency should be as local as possible (Puckette and Brown, 1989). This implies that the analysis window size should be as local as possible, but again, there is a trade-off between this claim and the wish to discriminate concurrent signal components. In order to comply with the stationarity criterion of the spectral transform, one solution lies in analysis methods which provide a more or less logarithmic frequency scale, for example the constant-Q transform or auditory models using filter banks (de Cheveigné and Kawahara, 1999; Klapuri, 2003b; Heinz, 2006; Paiva, 2006).

Unfortunately such techniques are often computationally expensive, so the Fast Fourier Transform remains the tool of choice in time-critical applications. In need of good frequency resolution, long FFT windows have to be applied – accepting a distorted spectrum for faster changing signal components. A compromise is a multi-resolution analysis based on the FFT algorithm. A prominent example is the application of a multirate filter bank in combination with the FFT used by Goto (2004). Another straight forward idea is the calculation of the FFT with different window lengths, resulting in different time-frequency resolutions, which has been successfully employed in melody extraction (Dressler, 2009), onset-detection (Böck et al., 2012a), and multiple fundamental frequency estimation and tracking (Elowsson and Friberg, 2014).

Essentially, the multi-resolution FFT (MR FFT) is an efficient implementation of this idea, which was first described in (Dressler, 2006). Thereby, the choice of the time-frequency resolution is guided by the following considerations: In polyphonic music, we find a mixture of voices in the low and middle frequency region, while the harmonics of the leading voice dominate the higher spectral bands (Goto, 2004). Thus a good frequency resolution is required mostly in the low frequency regions, where the harmonics exhibit a quasi-stationary frequency compared with the FFT filter bandwidth. With increasing harmonic number the frequency modulation of the partials becomes more evident, so for higher harmonics the stationarity criterion is often violated. The MR FFT analysis offers the possibility to adapt the frequency resolution accordingly and it considerably improves the instantaneous frequency estimation for the higher frequency regions. Moreover, the MR FFT improves the detection of short tones, if they have significant energy in high frequency regions. In our melody extraction system, the improvement reached by the application of the MR FFT instead of an FFT denotes 3.8 percent in the Overall Accuracy measure.

In (Dressler, 2006), the declared aim is the distinction between sinusoids and noise (deterministic–stochastic classification (Serra, 1989; Masri, 1996)). Yet in practice, local sinusoidality criteria for the detection of sinusoids often fail, because they depend on the shape of the spectral window function. Masri and Bateman (1995) presented a method for the identification of non-stationary sinusoids for well-defined types of spectral distortion, but in real-world signals the frequency modulation will not follow such idealized trajectories. We found later that a distinction between sinusoids and noise is not necessary for the purpose of note transcription, because the subsequent processing module (predominant pitch estimation) is robust enough to handle noisy data. That is why this processing step is omitted.

3.2. Multi-Resolution FFT

In the following sections we describe the implementation of the multi-resolution FFT as described in the publication (Dressler, 2006).

3.2.1. Implementation

Given a sequence of data samples $x[n]$ the Short Time Fourier Transform is defined as $X_l[k]$:

$$\begin{gathered} X_l[k] = \sum_{n=0}^{M-1} x[n+lL] \cdot e^{-j2\pi kn/N}, \\ l = 0, 1, ... \text{ and } k = 0, 1, ..., N-1 \end{gathered} \tag{3.1}$$

in which

N	is the number of STFT points
L	is the time advance of the data frame (hop-size)
M	is the size of the data frame
l	is the number of the data frame
k	is the frequency bin number.

Choosing for the Nth primitive root of unity $w = e^{j2\pi/N}$ allows $X_l[k]$ to be expressed in a shorter notation

$$\begin{gathered} X_l[k] = \sum_{n=0}^{M-1} x[n+lL] \cdot w^{-kn}, \\ l = 0, 1, ... \text{ and } k = 0, 1, ..., N-1. \end{gathered} \tag{3.2}$$

The values of N, L and M are the control parameters of the STFT, which determine certain characteristics of the spectrogram representation: the spacing of the discrete time-frequency grid of the spectrogram depends on the sampling rate f_s, the number of STFT points N and on the time advance of the data window L, which is also called hop-size. The grid spacing is determined by $\Delta f_{\text{grid}} = \frac{f_s}{N}$ and $\Delta t_{\text{grid}} = \frac{L}{f_s}$.

The grid spacing is not necessarily the time and frequency resolution we obtain from the spectrogram. The frequency resolution (the ability to distinguish two closely spaced frequencies from the original input signal) and also the time resolution is determined by the sampling rate, the size of the data window M and also by the shape of the window function, which will not be under consideration here. The respective frequency and time resolution are given by $\Delta f = \frac{f_s}{M}$ and $\Delta t = \frac{M}{f_s}$.

If we use zero-padding, the frequency resolution is worse than the spacing between the frequency bins, because the used data frame M is smaller than the number of

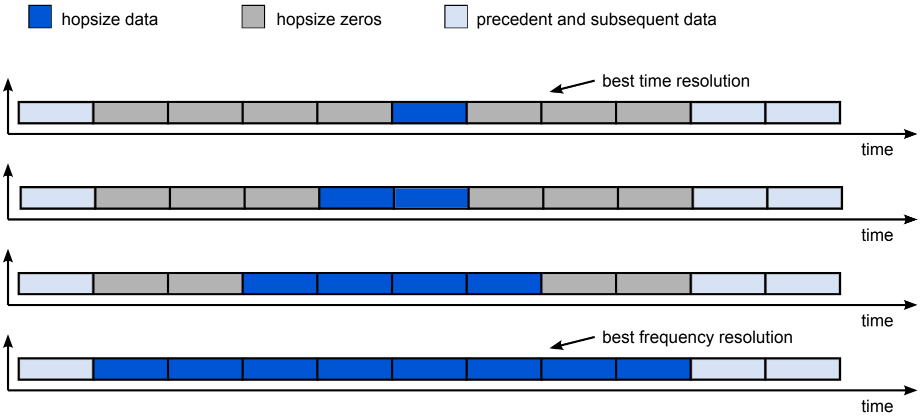


Figure 3.1. Data samples and zero-padding in the different MR FFT time-frequency resolutions. The width of one rectangle corresponds to the hop-size L.

STFT points N. If we use overlapping data frames ($L < M$), the time resolution does not increase, but nevertheless more STFT frames are included on the time axis. The additional information is obtained by interpolation.

Figure 3.1 shows a schematic representation of the (zero-padded) data used in the four resolutions in the MR FFT. The different time-frequency resolutions arise by changing the data frame size M, yet keeping the number of STFT points N and the hop-size L constant. The difference between the number of data samples on the one hand and the number of STFT points on the other hand is compensated by the insertion of zeros, the so-called zero-padding. As a consequence the spacing of the time-frequency grid remains unchanged among the different spectrograms. It can also be noted that the data window of the best time resolution is not centered. This results in a small asynchrony of about 3 ms in the estimated frequency values of this spectrogram resolution.

The basic idea of the MR FFT is derived from the fact, that the summation operation is associative, thus we are allowed to split and reorder summations. In a reformulation of equation (3.2) (for clarity with $M = N$), we split the original sum with length N into N/L sums of length L (hop-size). Hereafter we can again sum the partial sums, and the result is of course the same:

$$\begin{aligned} X_l[k] &= \sum_{n=0}^{N-1} x[n + lL] \cdot w^{-kn} \\ &= \sum_{c=0}^{\frac{N}{L}-1} \sum_{n=cL}^{(c+1)L-1} x[n + lL] \cdot w^{-kn}. \end{aligned} \tag{3.3}$$

The inner sum in equation (3.3) can be expressed as a (time-shifted) zero-padded

STFT of the data sequence $x_c[n]$:

$$X_c[k] = \sum_{n=0}^{N-1} x_c[n] \cdot w^{-kn}, \quad k = 0, 1, ..., N-1, \tag{3.4}$$

with

$$x_c[n] = \begin{cases} x[n+lL], & \text{for} \quad cL \leq n < (c+1)L; \\ 0, & \text{elsewhere;} \end{cases}$$

in which c is a circular counter related to the data frame number l by $c = l \bmod \frac{N}{L}$.

This transform can be computed by an FFT algorithm. The resulting complex Fourier coefficients are stored in a circular buffer of the dimension $[N/L, k_{\text{max}}]$, since one elementary transform is used in N/L calls of the MR FFT method, and only up to k_{max} frequency bins may be of interest for the subsequent analysis.

The FFT spectra $X_c[k]$ form the basis of the MR FFT: all different resolutions can be calculated as a summation of the (time-shifted) elementary transforms. Summing up to N/L neighboring elementary transforms increases the frequency resolution from f_s/L to f_s/N with the increasing number of summands r. In order to comply with the condition for windowing in the frequency domain (see Section 3.2.2), the number of summands r is restricted to certain values, as the fraction $N/M = N/(rL)$ has to be an integer value. For example, if $N = 2048$ and $L = 256$, the sum of $r = 1, 2, 4, 8$ elementary transforms is possible – resulting in four different spectrogram resolutions with $M = 256, 512, 1024, 2048$.

While the magnitudes of the summed spectrograms are immediately valid, the phase of the complex Fourier coefficients has to be corrected in order to make windowing in the frequency domain possible. The phase error is due to the time-shift of the data which introduces a phase shift in the frequency domain according to the shifting theorem of the STFT

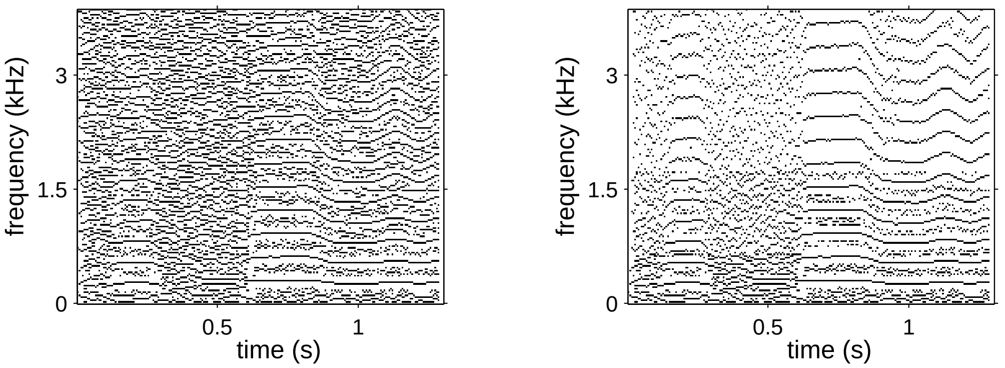
$$x[n+L] \underset{N}{\longmapsto} X[k] \cdot w^{kL}. \tag{3.5}$$

The angle of the phase shift depends on the frequency of the designated frequency bin k and the circular counter c, which is related to the data frame number l as defined in (3.4). Fortunately this effect can be canceled by multiplying the phase-shifted spectrum $X_r^*[k]$ with a twiddle factor as follows:

$$X_r[k] = X_r^*[k] \cdot w^{-k\, c_{\text{min},r} L}, \quad r = 1, 2, 4, ..., N/L, \tag{3.6}$$

where r is the number of summed elementary transforms $X_c[k]$ and $c_{\text{min,r}}$ is the circular counter index of the smallest frame number $l_{\text{min,r}}$ of the summed elementary transforms

$$c_{\text{min,r}} = l_{\text{min,r}} \bmod \frac{N}{L}. \tag{3.7}$$



(a) FFT: spectral peaks (b) MR FFT: spectral peaks

Figure 3.2. Comparison of spectral analysis using either FFT or MR FFT

For audio data sampled at $f_s = 44.1\text{kHz}$, we employ a multi-resolution FFT with $N = 2048$ and $L = 256$, resulting in four distinct spectrogram resolutions. While the best time resolution of 5.8 ms is obtained with the elementary transform ($M = 256$), the highest frequency resolution is achieved by the summation of all elementary transforms ($M = 2048$) and amounts to 21.5 Hz. The spectrogram with the most accurate frequency representation is used in the low frequency region, or to be exact, in the first six critical bands of the Bark scale. Accordingly, every other resolution covers five critical bands up to the maximum frequency $f_{\text{max}} = 5000$ Hz, i.e. $k_{\text{max}} = 232$.

Figure 3.2 shows a comparison of the spectral peaks obtained with either FFT (a) or the MR FFT (b). The decreasing number of peaks for the high frequency regions in the MR FFT example is due to a masking effect, which can be explained by the wider main lobe of the spectral window function with decreasing frequency resolution.

3.2.2. Windowing in the Frequency Domain

It is obvious that time domain windowing cannot be used with the MR FFT. But rather than applying the window in the time domain, there is the option to perform frequency domain windowing, because the transform of a product is equivalent to the convolution of the two corresponding transforms. Admittedly, convolution is a time consuming operation, and it is only an alternative if the discrete spectrum of the window function is a short sequence of convolution coefficients. Fortunately some common windows have this desired property. The temporal weightings of interest have the general form

$$\begin{aligned} h[n] &= \sum_{m=0}^{M/2} (-1)^m a_m \cos\left(\frac{2\pi}{M} mn\right) \\ &= a_0 - a_1 \cos\left(\frac{2\pi}{M} n\right) + a_2 \cos\left(\frac{2\pi}{M} 2n\right) - \\ &\quad a_3 \cos\left(\frac{2\pi}{M} 3n\right) + ..., \qquad n = 0, 1, ..., M-1, \end{aligned} \tag{3.8}$$

and

$$\sum_{m=0}^{M/2} a_m = 1,$$

in which M is the size of the data window and a_m are real constants (Nuttall, 1981). Since the most important windows of this form have $a_m \neq 0$ only for small m, equation (3.8) is reduced to a few terms.

If there are nonzero coefficients a_m for $m = 0, 1, .., K$, the continuous spectral window function $H(\omega)$ consists of a summation of $2K + 1$ weighted Dirichlet kernels

$$H(\omega) {=} a_0\, D(\omega) + \sum_{m=1}^{K} (-1)^m \frac{a_m}{2} \left[D\Big(\omega {-} \frac{2\pi}{M} m\Big) + D\Big(\omega {+} \frac{2\pi}{M} m\Big) \right], \tag{3.9}$$

where $D(\omega)$ is the Dirichlet kernel as given in

$$D(\omega) = \Big(+j\frac{\omega}{2}\Big) \frac{\sin\left(\frac{M}{2}\omega\right)}{\sin\left(\frac{1}{2}\omega\right)}. \tag{3.10}$$

The Dirichlet kernel is implicitly available through the STFT with a rectangular window. So for the discrete case and if $M = N$ equation (3.9) simplifies as indicated in

$$\begin{aligned} X[k]|_{\text{win}} = & a_0\, X[k] + \sum_{m=1}^{K} (-1)^m \frac{a_m}{2} \left(X[k-m] + X[k+m] \right), \\ & k = 0, 1, ..., N-1. \end{aligned} \tag{3.11}$$

That is the reason why these windows are especially useful for frequency domain windowing, because they can be described by a short $(2K + 1)$ sequence of convolution coefficients. For example the Hann and Hamming windows possess only two nonzero coefficients, the Blackman window three:

- Hann: $a_0 = 0.5$, $a_1 = 0.5$
- Hamming: $a_0 = 0.53836$, $a_1 = 0.46164$

- Blackman: $a_0 = 0.42$, $a_1 = 0.5$, $a_2 = 0.08$

Other windows of this form with very good sidelobe behavior are described in (Nuttall, 1981).

Yet, we have to pay attention to the fact that equation (3.11) only holds for the special case $M = N$. In order to apply frequency windowing to the transform of zero-padded data we have to refine

$$X[k]|_{\text{win}} = a_0\, X[k] + \sum_{m=1}^{K} (-1)^m \frac{a_m}{2} \left(X\left[k - m\frac{N}{M}\right] + X\left[k + m\frac{N}{M}\right] \right) \qquad k = 0, 1, ..., N-1. \tag{3.12}$$

The term $m\frac{N}{M}$ in (3.12) must be an integer, because we only know the Fourier coefficients $X[k]$ at discrete bin locations k. Hence the number of possible spectrogram resolutions is reduced to a subset by the condition $(\frac{N}{M} = \frac{N}{rL}) \in \mathbb{N}$, where r is the number of summed elementary transforms.

3.3. Estimation of the Instantaneous Frequency

There are many methods for the estimation of the instantaneous frequency (IF) and magnitude from Fourier coefficients. Keiler and Marchand compared some of the most popular ones in (Keiler and Marchand, 2002).

Two different classes of IF estimators are distinguished, namely magnitude and phase-based frequency estimators. The magnitude based frequency estimation methods evaluate the shape of the Fourier magnitude spectrum around a spectral peak. Considering three Fourier magnitudes, a refined location of the maximum is estimated, for example by parabolic interpolation (Serra, 1989). The location of the maximum corresponds to an enhanced frequency measure for the sinusoidal component.

Keiler and Marchand (2002) found that methods which are in some way based on the phase information of the STFT spectrum give the best results regarding frequency resolution. In real-world music, there is usually much interference encountered between partials of different sound sources, so this property is extremely important for melody extraction.

Marchand and Lagrange (2006) evaluate the performance of three phase-based frequency estimation approaches which analyze the variation of the complex Fourier coefficients in adjacent analysis frames. The three methods comprise the reassignment method, the difference estimator commonly used in the phase-vocoder approach, and the derivative estimator. The theoretical equivalence of the estimation methods is shown, however, small differences remain which are attributed to numerical errors of the mathematical functions used in the implementations. The numerical errors seem to be more pronounced for the reassignment method and the derivative estimator. However, according to Marchand the precision of the derivative estimator can be improved using different trigonometric functions depending on the evaluated frequency range.

Charpentier and Brown both developed independently an IF estimation method based on the derivative of the phase, in which the computation of the IF depends only on the current STFT frame (Brown and Puckette, 1993; Charpentier, 1986). Thus, contrary to the IF estimation methods examined in (Marchand and Lagrange, 2006), there are no overlapping STFT frames required.

We found that it is beneficial to use the average values from two frequency estimators in order to gain a more robust IF measure. Hence, we apply the well-known phase vocoder method and the approach described by Charpentier and Brown.

3.3.1. Phase Vocoder Method

The well-known phase vocoder method proposed by Flanagan and Golden computes the instantaneous frequency $f_i[k]$ from the phase difference $\Delta\phi[k]$ of successive phase spectra as follows (Flanagan and Golden, 1966):

$$f_i[k] = (k + \kappa[k]) \frac{f_s}{N}, \tag{3.13}$$

with

$$\kappa[k] = \frac{N}{2\pi L} \operatorname{princarg} \left[\phi_l[k] - \phi_{l-1}[k] - \frac{2\pi L}{N} k \right],$$

in which *princarg* is the principal argument function mapping the phase to the $\pm\pi$ range. The bin offset κ denotes the deviation of the partial's IF from the bin frequency expressed in the unit bin. If the estimated bin offset of a peak is less than $\pm 1/2$, we can say that the instantaneous frequency of the peak corresponds to the bin frequency. In order to estimate valid IF over a range of frequency bins with the phase vocoder method, the use of overlapping STFT windows (or zero-padding) is required, because otherwise the phase difference between frames might exceed 2π.

3.3.2. Charpentier/Brown IF Estimation

Charpentier and Brown both developed independently an instantaneous frequency estimation method, in which the computation of the IF depends only on the current STFT frame (Charpentier, 1986; Brown and Puckette, 1993). Thus, contrary to the phase vocoder method, no overlapping STFT frames are required. One drawback, however, is the requirement of the unwindowed STFT, so the time domain windowing of the sample data is not possible. In order to reduce the effects of spectral leakage, windowing can be implemented as a convolution in the frequency domain.

If the analysis window is shifted backwards one sample[1] in the time domain, this results in a phase shift of $2\pi k/N$ in the frequency domain. The corresponding Fourier coefficients $X^*[k]$ of the Fourier spectrum are estimated by

$$X^*[k] = w^{-k}\, X[k]. \tag{3.14}$$

If Hann windowing is applied in the frequency domain the shifted STFT coefficient can be obtained by the following formula:

$$X^*[k]|_{\text{win}} = w^{-k} \left[\tfrac{1}{2}X[k] - \tfrac{1}{4}(w\, X[k-1] + w^{-1}X[k+1])\right]. \tag{3.15}$$

This means a shifted version X^* of complex Fourier coefficients $X[k-1]$ and $X[k+1]$ has to be calculated:

$$\begin{aligned} X^*[k-1] &= wX[k-1] = e^{j2\pi/N}\, X[k-1] \\ X^*[k+1] &= w^{-1}X[k+1] = e^{-j2\pi/N}\, X[k+1] \end{aligned} \tag{3.16}$$

The Nth primitive root of unity w is used in order to twiddle right and left the neighbors of bin k. The factor w^{-k} can be omitted in the actual calculation. This factor denotes the expected phase shift $2\pi(k/N)$ of the bin's center frequency kf_s/N. This phase shift is again added in equation (3.18).

Then we do Hann windowing in the frequency domain for the original Fourier coefficients and for the one sample shifted Fourier coefficients:

$$\begin{aligned} X[k]|_{\text{win}} &= \tfrac{1}{2}X[k] - \tfrac{1}{4}(X[k-1] + X[k+1]) \\ X^*[k]|_{\text{win}} &= \tfrac{1}{2}X[k] - \tfrac{1}{4}(X^*[k-1] + X^*[k+1]). \end{aligned} \tag{3.17}$$

The phase shift $\Delta\phi_i$ consists of the expected phase shift and the difference between the phase angles of the respective Fourier coefficients.

$$\Delta\phi_i[k] = 2\pi(k/N) + \text{princarg}(\tan^{-1} X[k]|_{\text{win}} - \tan^{-1} X^*[k]|_{\text{win}}) \tag{3.18}$$

The instantaneous frequency f_i is finally estimated as

$$f_i[k] = \frac{\Delta\phi_i[k]}{2\pi} f_s. \tag{3.19}$$

[1] In case of the STFT a circular shift is performed within the analysis frame.

If the Charpentier/Brown frequency estimation is used with zero padding, not the neighboring Fourier coefficients are used, but coefficients $X[k \pm N/M]$. Equation (3.18) has to be modified to

$$\Delta\phi_i[k] = 2\pi(k/N) + \frac{N}{M}\text{princarg}(\tan^{-1} X[k]|_{\text{win}} - \tan^{-1} X^*[k]|_{\text{win}}). \qquad (3.20)$$

3.4. Estimation of the Instantaneous Magnitude

The choice of the windowing function determines processing gain and scalloping loss, which affect the magnitude response of the STFT (Harris, 1978). While the processing gain introduces a constant gain (or rather attenuation) for all signal components and therefore can be neglected in the analysis, the scalloping loss alters the magnitude relations between signal components:

The STFT can be seen as a bank of bandpass filters, whose outputs are combined to get a spectral representation of the signal. However, the overall STFT magnitude response curve is not entirely flat. The fluctuations in the magnitude response are called picket fence effect, or scalloping loss. The maximum magnitude response is reached if the instantaneous frequency of the signal component corresponds exactly to the center frequency of an STFT bin. The worst response occurs if the instantaneous frequency lies half-way between two STFT bins. The severity of the effect depends on the actual windowing function applied to the data. The scalloping loss may be as high as -3.9 dB for the rectangular window, for the Hann-window the maximum processing loss is only -1.42 dB.

This systematic error in the magnitude of the STFT coefficients may be corrected using for example a parabolic interpolation technique (Serra, 1989) or by using the exact shape of the windowing function in the frequency domain (Keiler and Marchand, 2002). Since the Hann-window can be described analytically in the frequency domain (Nuttall, 1981), we use the latter method to refine the magnitude. However, the technique is applied only to the first MR FFT resolution (the lowest frequency range), because in the zero-padded MR FFT spectra, the error is very small as the main-lobe of the window function is sampled at a higher rate.

The instantaneous magnitude of the sinusoidal peak is calculated from the magnitude of the STFT coefficient $|X[k]|$ and the estimated bin offset κ, which denotes the deviation of the partial's instantaneous frequency $f_i[k]$ from the bin's center frequency f_k expressed in the unit bin (see also equation 3.13):

$$\kappa[k] = \frac{N\,(f_k - f_i[k])}{f_s}, \qquad (3.21)$$

where f_s is the sampling frequency and N denotes the window size of the STFT. For a spectral peak referring to a sinusoidal partial, κ lies ideally in the range between

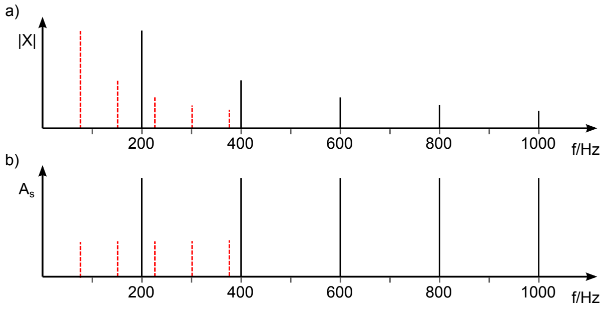


Figure 3.3. Magnitude weighting for two tones with a spectral rolloff of 6 dB per octave: a) STFT magnitude b) weighted magnitude

-0.5 and 0.5, but due to the interaction between signal components or due to the non-stationary frequency of a partial, other values may appear. As noisy components should not be boosted further, the value of κ is restricted to the ideal range.

Using the bin offset κ and the actual partial magnitude $|X[k]|$ we can estimate the true maximum value of the Hann window function in the frequency domain:

$$A_{\text{peak}} = \frac{|X[k]|}{W^*_{\text{Hann}}\left(\kappa[k]\right)}, \tag{3.22}$$

in which W^*_{Hann} is the normalized Hann window kernel, that has a maximum value of 1 for $\kappa = 0$:

$$W^*_{\text{Hann}}(\kappa) = \frac{\sin\left(\pi\kappa\right)}{\pi\kappa\left(1-(\kappa)^2\right)}.$$

(To obtain the exact analytical representation for the Hann window in the frequency domain, the kernel W^*_{Hann} has to be multiplied with 1/2.)

3.5. Magnitude Weighting

In order to obtain the weighted magnitude A_{peak} for the spectral peak at STFT bin k, its STFT magnitude $|X[k]|$ is multiplied with the peak's instantaneous frequency f_i.

$$A_{\text{peak}}[k] = |X[k]| \cdot f_i[k] \tag{3.23}$$

This weighting introduces a 6 dB magnitude boost per octave. In effect the weighted signal is proportional to the signal derivative.

The proposed magnitude weighting is based on the spectral slope of musical sounds, which measures how quickly the spectrum of an audio sound abates towards higher frequencies. Speech and instruments used in music have a spectral slope between -3 dB and -12 dB per octave (Tsang and Trainor, 2002). Mathews (2001) identifies the musically most interesting rates of spectral roll-off between 3 and 9 dB per octave. Sundberg (1979) found that the long-term average spectral slope of speech and orchestra music is -6 dB per octave. Hence, the weighting shall equalize the impact of low and high harmonics for the average complex tone in music, which ideally has a spectral slope of -6 dB per octave.

Figure 3.3 allows a qualitative comparison between the STFT magnitudes and the weighted magnitudes. The example uses two complex tones with 5 harmonics and a spectral slope of -6 dB per octave. After the weighting the harmonics of each tone have equal magnitudes. It can also be noted that the resulting spectrum is not flat. In this respect the proposed weighting differs from spectral whitening methods (for example (Klapuri, 2003a)), as it markedly damps the low frequency bands.

All subsequent processing steps are computed with the weighted spectral magnitude A_{peak}.

Chapter 4.

Predominant Pitch Estimation

4.1. Introduction

Complex harmonic tones have acoustic waveforms that repeat over time. The perceived pitch of such tones often corresponds to the repetition rate of the sound, or – in terms of frequency – to the fundamental frequency (F0). Still, even for tones with harmonic spectra the perceived pitch might deviate substantially from the fundamental frequency. Also, it may happen that the perceived tone height is ambiguous, depending upon the listener or the musical context. As pitch is a perceptual property, there is no definition based on purely objective physical terms. Usually pitch is seen as the perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from low to high (Klapuri and Davy, 2006, Chapter 1). A more restrictive definition is given by Plack and Oxenham (2005), who define pitch as 'that attribute of sensation whose variation is associated with musical melodies'.

Even though melody tones in general evoke a clear musical pitch, the analysis of real world music is a big challenge as the signal may include many different sound sources. Usually, the number of sources and their spectral envelopes are previously unknown quantities. Inharmonic spectra may occur, as well as percussive sounds. The estimation method has to be robust against spurious components, the interference of partials from different sounds, and octave ambiguities.

Often the melody voice corresponds to the predominant pitch which stands out from the backing accompaniment. But it may also be the case that two or more voices of comparable strength play simultaneously. Hence, even in the scope of melody extraction, pitch estimation cannot be limited to the detection of the predominant pitch. Rather, multiple fundamental frequencies have to be estimated to reliably identify the melody voice. For the detection of multiple fundamental frequencies, different approaches have been proposed which address the problem of shared harmonics. Such approaches include the iterative detection of the predominant pitch and the subsequent deletion of the tone (de Cheveigné and Kawahara, 1999; Klapuri, 2003a), as well as the joint pitch candidate selection (Klapuri, 2008; Pertusa and Iñesta, 2008; Yeh et al., 2010).

The proposed pitch estimation algorithm implements a delayed iterative approach and is based on the idea of subharmonic summation as described by Terhardt (1983). Subharmonic summation explains well the perceived pitch of harmonic complex tones and quantitatively predicts a great variety of pitch phenomena. There is one shortcoming of the above method that becomes very apparent in the analysis of polyphonic audio: each spectral peak creates a huge number of candidate virtual pitches, so the resulting pitch spectrogram becomes quite complex. In the presented algorithm, the number of possible subharmonics can be reduced considerably by the pair-wise processing of spectral peaks. In order to address the problem of shared harmonics and octave ambiguities, additional measures are introduced. The measures exploit the physical properties of musical sounds, for example the average spectral slope of complex tones, the harmonicity of partials, and the smoothness of the spectral envelope.

The algorithm described in (Dressler, 2011) is designed to solely detect the predominant pitch. In order to detect pitches besides the predominant one, the multiple F0 detection algorithm relies on high-level information extracted by the subsequent tone processing stage: the tone processing module provides information about the harmonic magnitudes of already detected tone objects. Those spectral peaks which are well explained by existing tone objects are inhibited prior to the actual pitch estimation. This way the starting points of weaker tones may be detected.

4.2. Prerequisites and Overview

The flowchart displayed in Figure 4.1 gives an overview of the pitch estimation method, which builds upon and extends the method described in (Dressler, 2011). The proposed method is based on the weighted spectral peaks of the MR FFT magnitude spectrogram and their respective instantaneous frequencies (IF). For the pitch estimation, spectral peaks in the frequency range between 55 Hz and 5 kHz are processed. The lower limit has been set according to the typical frequency range of melody notes, the higher limit denotes the frequency threshold for the induction of a virtual pitch in the human auditory system (Moore, 2003, Chapter 6). In order to obtain more stable IF measures, the average frequency of two estimation methods is used, namely the well-known phase vocoder (Flanagan and Golden, 1966) and a method proposed by Charpentier (1986) and Brown and Puckette (1993). Prior to the pitch estimation, each peak magnitude is weighted with its respective IF (see Section 3.5).

Since the pitch estimation algorithm is used to find starting points of high-level tone objects, it is important to give more weight to peaks/partials that are not covered by the long term timbre (spectral envelope) of existing tones. For this reason, a close interaction between pitch estimation algorithm and the high-level tone processing is necessary: in each analysis frame, prior to the detection of the

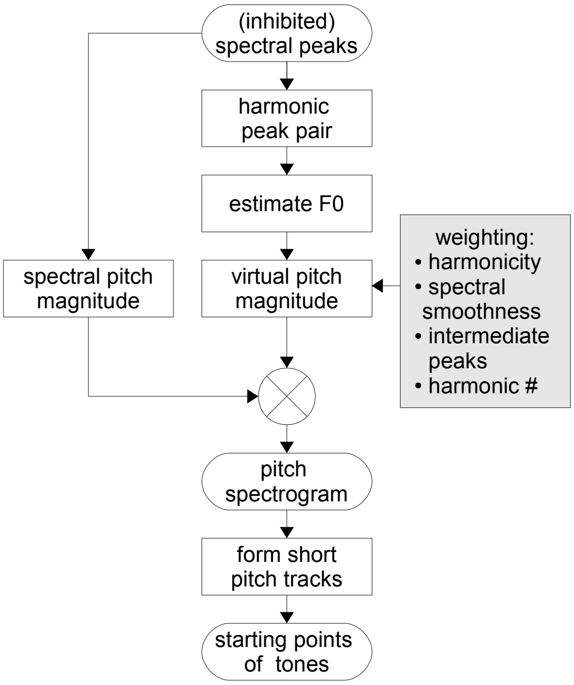


Figure 4.1. Overview of Pitch Estimation Algorithm

most salient pitches, the spectral peaks are first evaluated by the tone processing module in order to attenuate all spectral peaks that are well explained by the spectral envelope of existing tones. The remaining spectral peaks constitute the input to the pitch estimation method.

Terhardt (1983) draws a conceptual distinction between spectral pitch and virtual pitch. According to Terhardt, spectral pitch is a primary auditory sensation corresponding to the place of a local maximum excitation of the cochlear membrane. Virtual pitch, also referred to as residual pitch, is described as a secondary sensation which is deduced from a set of spectral pitches on another stage of auditory processing. A distinction between spectral pitch and virtual pitch is also apparent in the proposed algorithm structure shown in Figure 4.1. As indicated by the leftmost path in the flowchart, the weighted peak magnitude is added as spectral pitch magnitude directly to the pitch spectrogram.

The estimation of the virtual pitch magnitudes includes more processing steps. Consecutively, two spectral peaks at a time are combined into a candidate harmonic peak pair. It is then assumed that both peaks are successive (odd) harmonics (with harmonic numbers 1 and 2, 2 and 3,... as well as 1 and 3, 3 and 5, etc.). Following this assumption, it is possible to calculate the fundamental frequency of the perceived virtual pitch. Some additional weightings are applied which rate the probability that both peaks are indeed successive (odd) harmonics: 1) the harmonicity weighting rates the frequency relation between spectral peaks, 2) the spectral smoothness criterion determines the maximum supported virtual pitch magnitude, 3) the presence of intermediate spectral peaks reduces the impact of the considered peak pair, and 4) the harmonic number also influences the virtual pitch magnitude. After all peak pairs have been processed, the virtual pitch magnitudes are added to the pitch spectrogram.

4.3. Inhibited Spectral Peak Magnitude

The pitch strengths computed by the pitch estimation algorithm described in (Dressler, 2011) can be seen as probabilities of perceiving a predominant pitch. In order to retrieve other (weaker) tones, the effects of the predominant pitch(es) have to be factored out, because considerable pitch strengths occur at integer multiples of its fundamental frequency or – in combination with other periodic sound sources – at harmonically related frequencies. In order to identify tones besides the predominant periodic sound, an inhibition mechanism is implemented that diminishes the impact of already identified tones in the pitch spectrogram: spectral peak magnitudes are inhibited to the extent that respective harmonic magnitudes are established in high-level tone objects. As the amount of inhibition is deduced from already estimated long-term timbre information, it immediately affects the spectral peak magnitudes. Hence, one advantage of the proposed inhibition method is the fact that the pitch spectrogram is only estimated once in each analysis frame.

Two different magnitudes after inhibition are distinguished – the inhibited and the damped spectral peak magnitude. Both magnitudes are based on the difference between the weighted spectral peak magnitude A_{peak} (see Section 3.5) and the sum of all harmonic magnitudes $\hat{A}_{\mathrm{h}}$ (see Section 5.4.5), which are associated with the spectral peak. The distinction between the two values is the different amount of inhibition.

The inhibited magnitude $A_{\mathrm{peak\_inh}}$ seeks a complete inhibition of the spectral peak. It is computed as follows: for each spectral peak, the harmonic magnitudes $\hat{A}_h$ which are associated with that spectral peak are summed. Then, the sum is subtracted from the peak magnitude A_{peak}:

$$A_{\mathrm{peak\_inh}} = A_{\mathrm{peak}} - \min\left(\sum_{i}^{\mathrm{num\ tones}} A_{h,i} \quad , A_{\mathrm{peak}}\right). \tag{4.1}$$

The damped magnitude $A_{\mathrm{peak\_damped}}$ does not allow a full inhibition – a small fraction of the spectral magnitude always remains:

$$A_{\mathrm{s\_damped}} = 0.3A_{\mathrm{peak}} + 0.7A_{\mathrm{peak\_inh}}. \tag{4.2}$$

Both magnitudes replace the spectral peak magnitude A_{peak}, which is used in the original implementation of the pitch estimation algorithm as described in (Dressler, 2011). The inhibited peak magnitude $A_{\text{peak\_inh}}$ denotes the maximum pitch magnitude which may be finally added to the pitch spectrogram, $A_{\text{peak\_damped}}$ is used for the computation of the various weighting factors.

4.4. Spectral Pitch Magnitude

The spectral peak itself invokes a spectral pitch perceived at its own instantaneous frequency. So at first the inhibited magnitude $A_{\text{peak\_inh}}$ is added to the pitch spectrogram, if the instantaneous frequency f_i is in the desired pitch frequency range $f_{\min} \leq f_i < f_{\max}$. Since the pitch spectrogram has a logarithmic frequency scale, f_i is converted to a cent value c_i:

$$c_i = 1200 \log_2 \left(\frac{f_i}{f_{\text{ref}}} \right) \qquad \text{with} \quad f_{\text{ref}} = f_{\min}. \tag{4.3}$$

The minimum pitch frequency $f_{\min} = 55$ Hz is used as reference frequency. In this case the lowest possible cent value in the pitch spectrogram is zero. The maximum pitch frequency denotes $f_{\max} = 1318.5$Hz (E6). If the frequency resolution of the pitch spectrogram buffer is set to 1 cent, the estimated cent values can be used as indices to the spectrogram.

The spectral pitch magnitude is represented by a Gaussian weighted with $A_{\text{peak\_inh}}$:

$$g(c) = A_{\text{peak\_inh}} e^{-\frac{1}{2}\left(\frac{c-c_i}{35}\right)^2}. \tag{4.4}$$

The Gaussian reaches half its maximum value with a cent offset of $|c - c_i| \approx 41$ cent. The width of the Gaussian has been adjusted experimentally by the evaluation of the melody extraction system [1].

4.5. Virtual Pitch Magnitude

According to Terhardt (1983) the formation of virtual pitch is essentially a process of subharmonic matching. He presumed that each of the spectral pitches evokes candidate virtual pitches at its subharmonic frequencies. The subharmonic frequencies are found by dividing the partial frequency f_i by integer numbers from 1 up to the maximum allowed harmonic number [2]. Basically, the virtual pitch is perceived where the majority of the candidate virtual pitches of different spectral peaks match.

[1] In order to save computation time, the Gaussian weightings are precomputed and only 100 values are added to the pitch spectrogram, which has a resolution of 1 cent.

[2] Terhardt sets the maximum harmonic number to 12. In the presented approach, harmonics up to harmonic number 20 are considered.

The presented approach builds upon the idea of subharmonic matching. Still, contrary to Terhardt, we do not assume virtual pitch candidates at each subharmonic frequency of a spectral peak. Rather, we demand that only (odd) successive harmonics evoke a virtual pitch (Gerson and Goldstein, 1978). This way, the number of candidate virtual pitches can be decreased noticeably, because the considered subharmonic frequencies are derived from the frequency intervals between spectral peaks (see Section 4.5.1).

In principle the virtual pitch magnitude is derived from the weighted spectral magnitude of the identified harmonic. However, several additional ratings are introduced that estimate the probability of the virtual pitch. Consecutively, the identified harmonics are rated according to harmonicity, timbral smoothness, the appearance of intermediate spectral peaks, and harmonic number (see sections 4.5.2–4.5.5). A candidate partial that violates those principles is likely to belong to another sound source, or it might be heard as an individual tone (Hartmann, 1997, Chapter 6).

4.5.1. Pair-Wise Subharmonic Summation

In order to detect (odd) successive harmonics, spectral peaks are evaluated in pairs. Successively, each spectral peak is combined with all other peaks. For each peak pair, it is assumed that both spectral peaks are partials with an (odd) successive harmonic number (harmonic numbers 1 and 2, 2 and 3,... as well as 1 and 3, 3 and 5, etc.). Using the supposed harmonic relationship of the spectral peaks, the most likely harmonic numbers can be derived from their instantaneous frequencies f_{low} and f_{high}.

At first, it is assumed that both peaks are successive harmonics. In this case, the harmonic number h_{low} of the partial with the lower frequency f_{low} is computed as:

$$\frac{h_{\text{low}}}{h_{\text{low}}+1} = \frac{f_{\text{low}}}{f_{\text{high}}} \Rightarrow h_{\text{low}} = \text{round}\left(\frac{f_{\text{low}}}{f_{\text{high}}-f_{\text{low}}}\right). \tag{4.5}$$

The harmonic number of the partial with the higher frequency is $h_{\text{high}} = h_{\text{low}} + 1$.

Then, the supposed harmonic numbers are calculated assuming odd successive harmonics:

$$\frac{h_{\text{low}}}{h_{\text{low}}+2} = \frac{f_{\text{low}}}{f_{\text{high}}} \Rightarrow h_{\text{low}} = \text{round}\left(\frac{2f_{\text{low}}}{f_{\text{high}}-f_{\text{low}}}\right). \tag{4.6}$$

Using equation 4.6, the computed harmonic number is valid only if the rounded result is indeed an odd number. Naturally, the harmonic number of the partial with the higher frequency is $h_{\text{high}} = h_{\text{low}} + 2$.

Because of equation (4.6) odd harmonics are "discovered" more often than even harmonics. To avoid an increased impact of odd harmonics in the final pitch spectrogram, all identified harmonic numbers for one peak are at first solely listed. After all possible peak pairs have been evaluated, the computed virtual pitch is added to the pitch spectrogram only once for each harmonic number found.

The virtual pitch frequency f_p is computed individually for each partial by the straightforward division of instantaneous peak frequency and estimated harmonic number, e.g. $f_p = f_i/h$. Experimental results have shown that harmonics with a harmonic number h greater than 20 do not improve the estimation accuracy.

The virtual pitch magnitude is estimated using several weightings which are described in the following sections.

4.5.2. Harmonicity

Let's imagine a spectral peak pair with the instantaneous frequencies $f_{\text{low}} = 300$ Hz and $f_{\text{high}} = 400$ Hz. According to equation (4.5) the harmonic number is calculated as $h_{\text{low}} = 300\,\text{Hz}/(400 - 300)\,\text{Hz} = 3$. Since the values form an example of an ideal harmonic relation between successive harmonics, the result is exactly the harmonic number and has not be rounded. If we consider another peak pair with the instantaneous frequencies $f_{\text{low}} = 300$ Hz and $f_{\text{high}} = 415$ Hz, the result of equation (4.5) before rounding is approximately $h^*_{\text{low}} = 2.6$. In this case it may be doubted that both peaks are successive harmonics, because the frequency interval is not close to any ideal harmonic relation.

Most of the evaluated peak pairs are actually not successive (odd) harmonics. The estimated ideal harmonic relation can be a criterion to rule out such peak pairs. When a low numbered harmonic of a complex tone is progressively mistuned, the pitch of the complex changes (Darwin et al., 1994). The pitch change reaches a maximum at about 3% mistuning and and by about 8% the pitch of the complex tone has returned to its initial value. At the same time, the increasingly mistuned harmonic begins to stand out from the complex and is perceived as a discrete tone. Darwins experiment showed that the frequency offset from the ideal harmonic position should not be greater than about 135 cent – otherwise the harmonic has no impact on the overall pitch. We achieved the best melody extraction accuracies, when the allowed offset between the estimated frequency interval and the exact harmonic interval is set to 100 cent:

$$1200 \cdot \left| \log_2 \left(\frac{f_{\text{high}}}{f_{\text{low}}} \right) - \log_2 \left(\frac{h_{\text{high}}}{h_{\text{low}}} \right) \right| < 100 \tag{4.7}$$

If, the frequency interval between two peaks shows a 100 cent offset from the exact harmonic interval, both peaks will probably not belong to the same sound source. But maybe both peaks are indeed successive harmonics, and only the estimated instantaneous frequencies are erroneous. Anyway, the virtual pitches which are induced by both peaks will not combine to a joint pitch in the pitch spectrogram, because the Gaussian function which is used in the summation has its inflection points at ±35 cent. Hence, the estimated pitch of this peak pair is ambiguous. Nonetheless, such a marked offset is allowed in order to obtain as many valid peak pairs as possible – even though the frequency relation is quasi inharmonic. Very often the ambiguity is resolved by the summation of other harmonics, so that in the end the best matching virtual pitch frequency can be estimated with some reliability.

On the logarithmic frequency scale only the lowest neighboring harmonics have very distinct frequency intervals (1200, 702, 498, 386 cent), while for example the intervals between harmonics 14/15 and 15/16 are 119.4 cent and 111.7 cent, respectively. This means that only for the lower harmonics the harmonicity can be an effective criterion to rule out peak pairs. As the frequency intervals between high harmonics are very similar on the logarithmic frequency scale, even small deviations from the ideal harmonic frequencies can lead to a faulty estimation of the harmonic number. That is the reason why the virtual pitch estimates from high harmonics are not very reliable.

The harmonicity rating r_i is implemented as a boolean value – the peak-pair is discarded if the condition given in 4.7 does not hold.

4.5.3. Attenuation by Intermediate Peaks

Usually, each spectral peak is combined with a number of different peaks from the lower and higher frequency range. Among the possible peak combinations, the pairings of immediately neighboring spectral peaks are of particular interest for the pitch estimation. Nonetheless, we do not use the order of the peak combination directly as a measure, because the spectrum also includes spurious peaks, which might skew the rating. Rather, the damped magnitudes of the intermediate spectral peaks are summed up and compared to the magnitudes of the evaluated peak pair. If the noise level is comparatively low, at least the noise peaks will not influence the rating too much.

The rating factor r_m, which represents the attenuation of the virtual pitch magnitude due to intermediate spectral peaks, is given by

$$r_m = \frac{A_{\text{min}}}{A_{\text{min}} + \sum_i A_i}. \tag{4.8}$$

The term $\sum_i A_i$ denotes the sum of all peak magnitudes $A_{\text{peak\_damped}}$ that exist between the evaluated peaks. The term A_{min} is the smaller (damped) spectral magnitude of the evaluated peak pair, e.g. $A_{\text{min}} = \min(A_{\text{low}}, A_{\text{high}})$. The resulting rating factor r_m is used to weight the supported magnitude, which is described in the next section.

The main benefit of this criterion is the prevention of octave errors. Of course, intermediate peaks also occur because of the overlapping spectra of simultaneous sound sources. But certainly, advantage is taken from the fact that often the timbres of different instruments dominate in different spectral regions. For example, the strongest partials of the bass instrument are often found in the low frequency range, while the melody voice usually has strong harmonics in the high frequency regions.

4.5.4. Spectral Smoothness

Most instrument sounds show pronounced peaks (formants) as well as regions with lower energy in their spectral envelope. While it is impossible to make predictions about the spectral power distribution as a whole, a certain smoothness of the spectral shape is observed. The assumed smoothness of the spectral envelope can be used as an additional criterion in the pitch extraction.

As a consequence it can be assumed that successive harmonics have more or less the same magnitude. However, some stopped-pipe wind instruments, for example the clarinet or the panpipe, have timbres that mostly contain odd harmonics. In this case it is hard to find partials with successive harmonic numbers. So again, odd successive harmonics have to be considered to estimate the smoothed spectral envelope.

The supported smoothed virtual pitch magnitude S_h depends on the damped spectral peak magnitudes $A_{\text{peak\_damped}}$ of the neighboring harmonics. To improve the readability of the following formulas, the damped spectral magnitudes of the candidate harmonics are notated as A_h, A_{h-1}, A_{h+1} etc., this means the assumed harmonic number is indicated by the subscript.

At first, the preliminary support magnitudes S^- and S^+ are estimated separately for each frequency direction. If the current harmonic number h is even, S^- and S^+ are computed from the magnitudes of the harmonic neighbors A_{h-1} and A_{h+1} and the attenuation by intermediate spectral peaks r_m:

$$\begin{aligned} S^- &= r_{m,h-1} \cdot \min(4 \cdot A_{h-1}, A_h) \qquad \text{and} \\ S^+ &= r_{m,h+1} \cdot \min(4 \cdot A_{h+1}, A_h). \end{aligned} \tag{4.9}$$

If the current harmonic number h is odd, also the odd harmonic neighbors are considered:

$$\begin{aligned} S^- &= r_{m,h-2} \cdot \min(4 \cdot A_{h-2}, A_h) \qquad \text{and} \\ S^+ &= r_{m,h+2} \cdot \min(4 \cdot A_{h+2}, A_h). \end{aligned} \tag{4.10}$$

In this case, the bigger supported magnitude in each direction is taken for the subsequent calculation. If a partial has only one harmonic neighbor, the other support magnitude is set to zero.

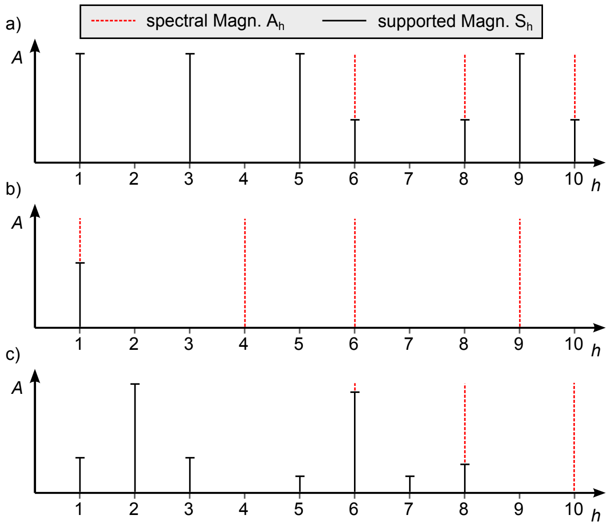


Figure 4.2. Combination of harmonic candidates and supported virtual pitch magnitudes: a) combination of 7 partials, which have at least one neighboring (odd) harmonic, b) combination of 5 partials, which have no harmonic neighbors, c) combination of 8 partials which have distinct weighted magnitudes.

The final supported magnitude S_h is a weighted sum of S^- and S^+. In the weighted sum, the smaller supported magnitude gets a higher weight. Three conditions are distinguished:

$$S_h = \begin{cases} 0.75\,A_h + 0.6\,S^+ & \text{if } h = 1 \\ 0.4\,S^- + S^+ & \text{if } S^- > S^+ \\ 0.4\,S^+ + S^- & \text{else.} \end{cases} \tag{4.11}$$

The estimated magnitude support S_h must not be greater than the inhibited peak magnitude $A_{\text{peak\_inh}}$, otherwise S_h is reduced to the value of $A_{\text{peak\_inh}}$. The constant factors used in equations 4.9 - 4.11 have been found empirically.

Figure 4.2 shows the smoothed spectral envelopes for different combinations of sinusoidals. The three examples show how missing or weak harmonic neighbors lead to a reduction of the supported virtual pitch magnitude. If no harmonic neighbors can be identified (as is the case for partials 4, 6 and 9 in Figure 4.2 b), no virtual pitch is induced. Still, the implemented timbral smoothing allows a certain degree of variation in the spectral envelope, as can be noted in Figure 4.2 c).

The required support from neighboring harmonics is an important difference to Terhardt's algorithm (Terhardt, 1983). If some intermediate harmonics are missing or cannot be detected the algorithmic results differ drastically, as can bee seen in Figure 4.2, example b). In our algorithm, the non-supported harmonics 4, 6 and 9 do not contribute at all to the overall pitch of a complex tone. In Terhardt's subharmonic summation, however, each spectral peak is a candidate harmonic and its instantaneous frequency is divided subsequently by integer numbers 1 to 12 to estimate potential fundamental frequencies of a complex tone. A so-called virtual pitch pattern is obtained by an advanced algorithm of subharmonic coincidence assessment, and and in this case, virtual pitches of harmonics 4, 6 and 9 would match at the same fundamental frequency.

4.5.5. Harmonic Impact

A small, but positive effect is gained if the impact of the higher harmonics is reduced by a small amount. The damping of higher harmonics amounts to only 1 dB per octave. The weighting factor r_h depends on the harmonic number h:

$$r_h = h^{-\frac{1}{20\log(2)}}. \tag{4.12}$$

The parameter r_h denotes the harmonic impact.

4.5.6. Estimation of the Rated Virtual Pitch Magnitude

Finally, in order to obtain the virtual pitch magnitude A_v, the supported peak magnitude S_h is multiplied with the harmonic impact r_h:

$$A_v = r_h \cdot S_h. \tag{4.13}$$

The resulting virtual pitch magnitude A_v is added to the pitch spectrogram in the same way as the spectral pitch magnitudes (see Section 4.4).

4.6. The Harmonic Count Spectrogram

The harmonic count spectrogram counts the number of harmonics at a certain pitch value. Thereby, the harmonic count does not just increase by one every time a spectral peak is turned into a (virtual) pitch. Often, the added pitch magnitude is diminished compared to the spectral peak magnitude – according to the inhibition by existing tones, to harmonicity, spectral smoothness, and the attenuation by intermediate peaks. Hence, the reached inhibited pitch magnitude is set into relation with the full magnitude of the spectral peak. So the maximum value that can be reached for each added (virtual) pitch is one, the minimum is zero.

The harmonic count spectrogram is computed with a resolution of 25 cent.

4.7. The Pitch Spectrogram

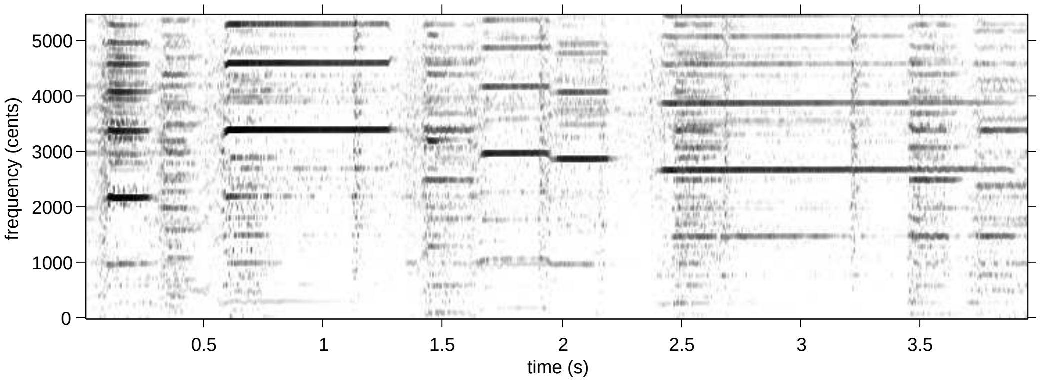


Figure 4.3. jazz1.wav: Pitch Spectrogram

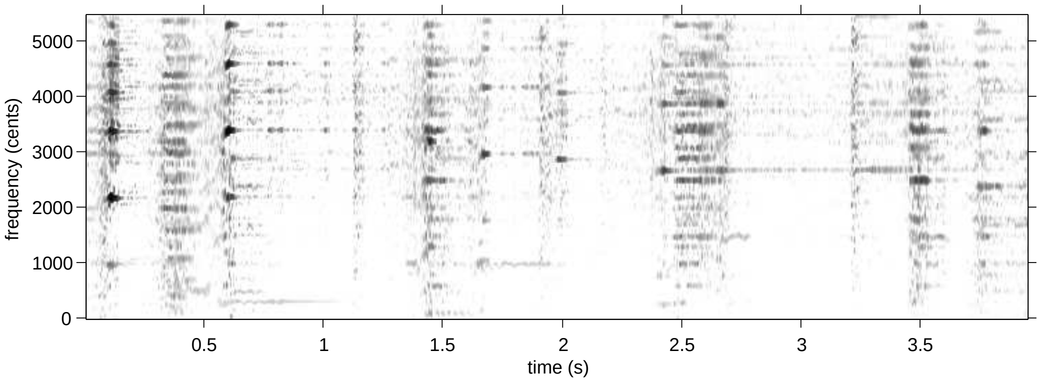


Figure 4.4. jazz1.wav: Inhibited Pitch Spectrogram

Figures 4.3 and 4.4 show a comparison between the unaltered pitch spectrogram and the inhibited pitch spectrogram, which is used to find starting points of new tone objects. Once a tone is started, the partials of the tone are inhibited. Then, the inhibited pitch spectrogram is computed. In this way, another – weaker – tone becomes the predominant one and may start a new tone.

It should be noted that the pitch spectrogram gives only a general idea of the perceived tone strength and tone height. The estimated pitch salience can help to identify the predominant tone in the audio mixture, but it does not exactly correspond to the perceived tone magnitude. In particular, the magnitude estimate should not depend substantially on the existence of other audio sources, as it does in the proposed pitch estimation algorithm.

Another problem is the simultaneous estimation of pitch frequency and pitch magnitude in one processing step. Psychoacoustic experiments show that the perceived pitch is to a large extend determined by the lowest harmonics, so the lowest harmonics should receive a stronger weighting (Moore, 2003, Chapter 6). Still, the high harmonics contribute significantly to the perceived pitch strength. It becomes clear that two separate processing steps are required to calculate tone magnitude and pitch.

But there is another major problem about the pitch spectrogram. There is no information about timbre. Some phenomena, as for example the continuity effect of a masked tone, cannot be replicated without the knowledge of timbre. If tones are masked by noise or concurrent sounds, even the most careful analysis and evaluation of the spectral peaks cannot resolve such problems without the knowledge of past events. Often a pitch track cannot be continued, because adequate pitch candidates are missing. Here, the question arises whether the tone has indeed finished or whether the tone is currently masked by another sound.

A more abstract processing level must be assumed that uses information from a longer time span in order to establish the spectral envelopes of different tones. Then, it is possible to compute a more reliable estimate of tone height and tone magnitude. Also, the impact of noise and masking can be diminished, and often octave ambiguities can be resolved. That is the reason why the tone tracking problem is postponed to a higher processing level and salient pitches are used merely as indicator for starting points of tone objects.

Chapter 5.

Tone Estimation and Tracking

5.1. Introduction

Subsequent to the pitch determination, a typical processing step would be pitch tracking: salient pitches of individual analysis frames are connected to form continuous tracks (Goto, 2004; Paiva, 2006; Cao et al., 2007; Salamon, 2013). The melody voice is often the predominant musical voice in the audio signal. Therefore, a very basic approach to tackle the problem of melody extraction would simply pick the frequency with the highest salience in each analysis frame (Dressler, 2011). Wendelboe (2009) describes a melody detection algorithm that plainly chooses the maximum subharmonic summation value of each analysis frame and nonetheless obtains a remarkable raw pitch detection accuracy. Anyway, very few melody extraction algorithms use the pitch magnitude in such a straight forward manner, because the frame-wise pitch analysis is not very reliable for sound mixtures.

Admittedly, not all melody extraction systems explicitly detect tones or pitch tracks. Some systems omit this processing step and use the frame-wise pitch estimates and their saliences as input to a statistical framework which retrieves the most probable melody pitch contour (Poliner and Ellis, 2006; Durrieu, 2010; Rao and Rao, 2010). However, timbral features gain importance for the selection of melody tones and although such features can also be incorporated in a global statistical framework, a separate tone processing is often more efficient.

While the pitch determination algorithm described in the previous chapter uses solely information from one analysis frame to detect the predominant periodic signal, the tone processing aims at the formation of multiple continuous tone objects and at the computation of high-level information like tone magnitude, tone onset and offset as well as the discrete tone height. Thereby, data from several analysis frames has to be assembled and converted to reliable long-term information. A speciality of the proposed algorithm is the computation of the tone's spectral envelope. The goal is to reproduce the harmonic magnitudes of the tone as accurately as possible.

5.2. Overview

The input to the tone processing module is the list of salient pitches, the magnitude spectrogram of the MR FFT, and a list of all spectral peaks including their magnitudes and instantaneous frequencies. The output of the tone processing module is a list of high-level tone objects, which provide information about instantaneous features that describe the evolution of the tone, as well as more abstract long-term features which allow an efficient retrieval of the melody voice.

The tone processing is the most complex module within the melody extraction system. Since many different processing steps are involved, it is easy to loose track of things and therefore very difficult to get the big picture of the algorithm. That is the reason why the most important ideas are introduced in this overview.

One of the challenges in sound segregation is to distinguish between the following two cases: at the one hand there are concurrent musical voices that often have harmonically related fundamental frequencies (F0), at the other hand there are octave errors with the very same harmonic frequency relations. So the main challenge is to distinguish legitimate tones from erratic tones which are often found at integer multiples of the actual fundamental frequency – the so called octave errors. This is not an easy task, because it is not unusual to find actual tones playing simultaneously at such frequency ratios.

In order to cope with the problem of octave errors and shared harmonics, most algorithms for the extraction of multiple F0s implement either the iterative approach (which repeatedly detects the predominant periodic signal and deletes it from the sound source) or the joint estimation (where all spectral peaks are evaluated together to find the most probable set of pitches). The proposed algorithm has many characteristics of the iterative approach, but it also exploits ideas from the joint estimation method. We will briefly introduce both concepts in the following sections.

5.2.1. Iterative Detection

A common technique for the estimation of multiple F0s is the iterative estimation-suppression-estimation method (Parsons, 1976; de Cheveigné and Kawahara, 1999; Klapuri, 2003a). In each turn of the algorithm, the predominant periodic sound is identified, its spectral envelope is estimated, and subsequently the sound is removed from the audio mixture. Then, the detection and suppression step is repeated on the residual signal, until the estimated number of concurrent F0s is reached. A challenging problem is the handling of shared harmonics. If two F0s are harmonically related, much energy of the weaker sound might already be removed during the suppression of the stronger one. This problem becomes most eminent if both tones have an octave relationship. A possible solution that has been investigated by

Klapuri (2003a) and others are additional constraints regarding the smoothness of the spectral envelope.

Our implementation is related to the iterative method. The most noteworthy difference to other iterative approaches is the feedback link that exists between tone processing and the pitch determination algorithm. Contrary to the above mentioned methods, the identification of tones strongly relies on information gathered in previous analysis frames. After a new tone object is started, it's spectral envelope is gradually established over time. Since the spectral envelope changes rather slowly, the information from the last analysis frame can be used to determine the inhibition of the spectral peaks in the current analysis frame. In this context, Bregman coined the term "Old-Plus-New" heuristic – a concept that he explained as follows (Bregman, 1994, p. 222):

> If you can plausibly interpret any part of a current group of acoustic components as a continuation of a sound that just occurred, do so and remove it from the mixture. Then take the difference between the current sound and the previous sound as the new group to be analyzed.

This idea is put into practice in several parts of the algorithm, but the best example is the pitch determination algorithm (PDA) that takes a residual spectrum as input: In each analysis frame, at first the appropriate spectral peaks are assigned as harmonics to existing tones, then, all assigned spectral peaks are inhibited according to the long-term spectral envelope of the tone. This means that the PDA introduced in the previous chapter will only be executed once in each analysis frame, taking the already inhibited spectral peaks as input. If the suppression of existing tones is effective, new or weak tones can be detected by the PDA and a new tone object might be started.

One advantage of the proposed inhibition method is that the PDA is only executed once in each analysis frame. The main disadvantage is certainly the problem of error propagation. Usually, iterative approaches calculate all fundamental frequencies in each analysis frame from the scratch and are therefore not affected by any wrongly detected pitches in previous frames. In the presented approach, however, an erroneously started note has a marked influence on the overall detection of multiple F0s. Even though the predominant voice is retrieved quite reliably, different combinations of tone objects might arise from a mixture of harmonically related notes – just by a slight change of algorithm parameters.

There are some mechanisms to correct spurious tones during the subsequent joint evaluation of tones, but in case of an error, often a noticeable time lag is introduced for the detection of the true pitches. And unfortunately, some errors are not detected at all. This is the reason why much attention is directed to the intermediate pitch tracking stage (see Section 5.3) that allows a more reliable detection of tone onsets.

5.2.2. Joint Evaluation

The joint evaluation of tones is primarily dedicated to the problem of shared harmonics: For each pitch candidate combination, the contribution of the partials to the distinct F0 is estimated jointly, so that every F0 has the same chance to profit from shared harmonics. For example, Klapuri (2006) suggests a harmonic summation model in which the pitch salience contribution of each sound is reduced by the inhibition from sounds with shared harmonics, as determined by their estimated spectral envelopes. The simultaneous computation of the harmonic magnitudes is a pronounced advantage of the joint evaluation method over the iterative approach, in which the first detected F0 usually profits more from a shared harmonic than some F0 that is detected in a subsequent iteration of the algorithm. Benetos and Dixon (2011) propose a method for the automatic transcription of music signals based on joint multiple F0 estimation. The problem of the pitch candidate selection (2^{10} possible combinations for 10 pitch candidates) is handled by a salience function that evaluates harmonics up to the 12th overtone. Overlapping partials are treated by a method which takes the estimated spectral envelopes into account.

The main challenge of all joint estimation methods is to efficiently identify the number and final combination of pitch candidates, because the number of possible combinations rises very quickly with the number of pitches. That is the reason why most iterative approaches are computationally more efficient. Nonetheless, many recent methods use the joint estimation approach, although the results that are achieved with either the iterative or the joint estimation method are not significantly different (Klapuri, 2006).

Yeh (2008) uses a combination of iterative and joint estimation: first, non harmonically related F0 are estimated in an iterative estimation-suppression process. Then, harmonically related F0 are estimated based on spectral envelope smoothness. And finally all partials are jointly processed based on harmonicity, spectral envelope smoothness, energy in the lowest partials and synchronous amplitude evolution.

It becomes obvious that the smoothness of the spectral envelope is an important feature for all multiple F0 estimation algorithms. Spectral envelopes tend to vary smoothly as a function of frequency and time. So again, in order to find the best combination in a set of pitch candidates, Pertusa and Iñesta (2008) do not only maximize the sum of harmonic amplitudes while minimizing the number of utilized pitches, but also evaluate the spectral envelope.

As mentioned before, our method is more or less a frame-wise delayed iterative estimation-suppression approach: in each analysis frame, the Fourier spectrum is inhibited according to the spectral envelopes of the already existing tone objects. At the most, one new tone can be detected per frame, which is done efficiently by the predominant pitch estimation on the inhibited Fourier spectrum. Nonetheless, the proposed tone processing method handles shared harmonics jointly insofar, that the summed harmonic magnitudes of a shared harmonic should not exceed the given magnitude of the corresponding spectral peak. Moreover, the joint evaluation of harmonics is an additional feature in some other processing steps: it is used for the weighting of spectral peaks, for the continuous validation of the harmonics, and for the detection of octave errors.

5.2.3. From Frame-Wise Observations to Long-Term Information

Within the proposed melody extraction algorithm, the exponential moving average (EMA) is applied in order to transfer frame-wise observations to more reliable long-term information. The calculation of the EMA is described in appendix A. The EMA can be identified as low-pass filtering, giving much weight to the most recent data, while old observations loose impact gradually. It provides an elegant way to pool past measurements without much memory consumption, since it can be calculated very efficiently using the following recursive formula:

$$\bar{y}_t = \alpha \cdot \bar{y}_{t-1} + (1 - \alpha) \cdot y_t. \tag{5.1}$$

The equation shows that the EMA at time period t can be calculated from only three numbers: the current observation y_t, the preceding EMA value $\bar{y}_{t-1}$ and the smoothing factor α. The parameter α corresponds to a specific half-life period. It might be constant or variable. Within this thesis, an EMA that is marked by a hat (for example $\hat{y}$) stands for a variable smoothing factor. If the EMA is marked by a bar (e.g. $\bar{y}$) a constant smoothing factor is used.

The corrected EMA-value is indicated by the fraction $y_t/\bar{w}_t$, in which $\bar{w}_t$ is a correction factor that refers to the sum of previously used smoothing factors. The result is a more accurate specification of the EMA (see the appendix A for a more detailed explanation).

Applied as a smoothing function, the EMA is used to calculate an adaptive average for many variables throughout the melody extraction system: for example the tone height and the tone's long-term magnitudes, but also the salience and the central pitch of musical voices. The exponential decay that is characteristic for the estimation of the weights in the EMA, is also used to determine adaptive thresholds.

5.2.4. Temporal Succession of Processing Steps

The chapter as a whole is organized in a thematic way, yet this section is supposed to show the temporal order of the processing steps. The tone processing can best be understood with help of the flowchart in Figure 5.1, which shows the main processing

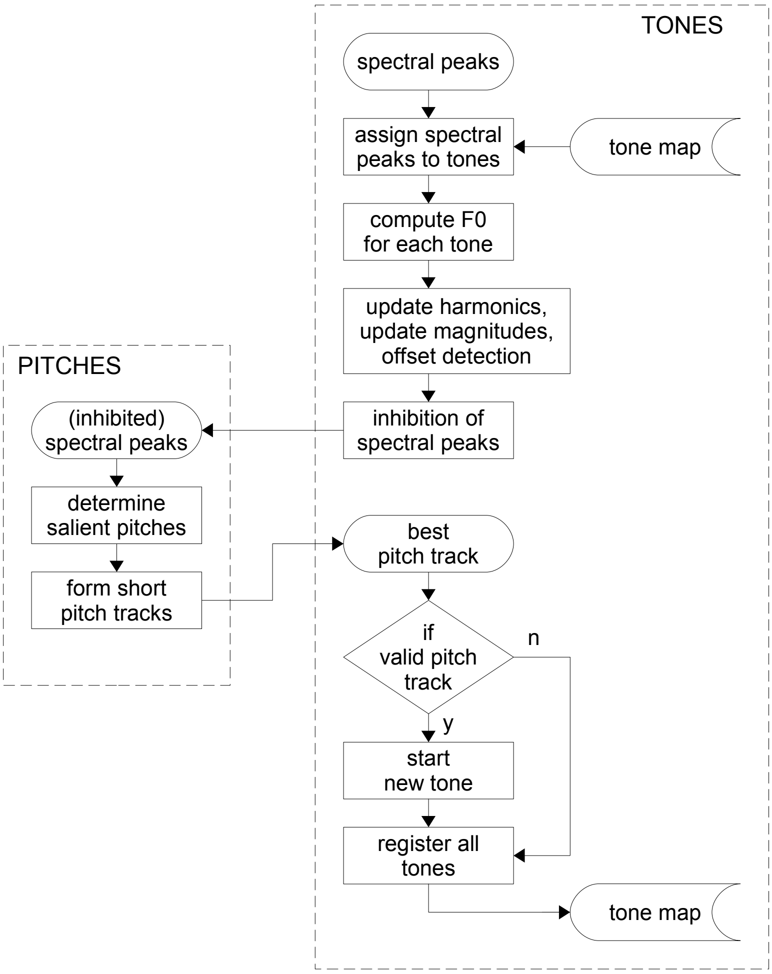

TONES
spectral peaks
assign spectral peaks to tones
tone map
compute F0 for each tone
update harmonics, update magnitudes, offset detection
PITCHES
(inhibited) spectral peaks
inhibition of spectral peaks
determine salient pitches
form short pitch tracks
best pitch track
if valid pitch track
n
y
start new tone
register all tones
tone map

Figure 5.1. Overview of the Tone Processing

steps that are performed in each analysis frame. It is explained further in the step-by-step summary given in the list below.

The flowchart shows a very short recap of the pitch determination algorithm (PDA) at the left hand side, and at the right hand side the most important tone processing steps. The first part of the tone processing module is dedicated to the processing of the existing tone objects. Once all tone objects have been updated, the pitch detection is performed based on the inhibited spectral peaks. Thereby, the attenuation of the spectral peaks is determined by the spectral envelope of the existing tones. So the PDA mainly detects pitches that are not explained by existing tones. If the most salient pitch (or rather pitch track) in the residual pitch spectrogram has a significant magnitude, a new tone will be started. The following list briefly explains all processing steps:

- Assignment of candidate harmonics to existing tones (Section 5.4.1): First, the spectral peaks are assigned to all tones which are close to the peak's instantaneous frequency or to its associated subharmonic frequencies (virtual pitch). The virtual pitches of a spectral peak are identified by the pair-wise combination with another spectral peak.

- Computation of the tone height (Section 5.6): First, the assigned spectral peaks are weighted according to the long-term spectral envelope of the tone and by the joint evaluation of shared harmonics. This way the impact of noise and other sound sources can be decreased noticeably. Then, the tone height is calculated by an iterative algorithm using the estimated weights. After the F0 of the tone has been updated, spectral peaks with a great offset to the ideal harmonic frequency are removed, while appropriate peaks which have not yet been registered are added.

- Update of the tone's harmonics (sections 5.4.4 and 5.4.5): First, the supported harmonic magnitude is estimated for each harmonic. The supported harmonic magnitude determines how well a spectral peak can be integrated into the overall timbre. The principle indicators for the harmonic fit are: 1) the offset between the spectral peak's instantaneous frequency and the ideal harmonic frequency, 2) the timbral smoothness (spectral envelopes of real sounds tend to vary smoothly as a function of frequency and time), 3) the magnitude division of shared harmonics, and 4) eventual masking is taken into account.

  Then, the harmonic magnitude is updated according to the supported harmonic magnitude.

- Magnitude update (Section 5.5): With help of the current harmonic magnitudes, the established spectral envelope, and the MRFFT magnitude spectrum, four different tone magnitude measures are estimated – namely, the tone magnitude, the supported tone magnitude, the exclusive tone magnitude and the salience. Beyond that, many other (long-term) variables describing the tone object are updated.

- Offset detection (Section 5.8): Now, an eventual tone offset can be detected. Furthermore, spurious tones and octave errors are identified and removed.

- Inhibition of spectral peaks (Section 5.4.8): The spectral envelope of the tone provides a useful feedback to the PDA. All spectral peaks that are well explained by the timbre of tones are inhibited. This way, the PDA only detects pitches that are not already covered by existing tones. The actual attenuation depends on the updated harmonic magnitude(s).

- Pitch determination algorithm (chapter 4): Computation of the pitch spectrogram with inhibited spectral peaks.

- Check for new tone onsets (Section 5.3.1): Formation of pitch tracks by the linking of salient pitches across several analysis frames in the pitch spectrogram. The most salient pitch track is a candidate for a new tone onset. If it reaches the necessary rating and passes the current magnitude threshold, a new tone is started.

- Tone map (5.9): Now, the tone's pitch is predicted for the next analysis frame and finally all tones are entered into a tone map so that they can be found by looking up a frequency value. This way it is possible to add new candidate harmonics in the subsequent analysis frame.

The MIDI note labeling takes place after the whole music file has been processed, because the estimation of the reference frequency is an offline procedure.

5.3. Pitch Tracks: Starting Tones

In our system, the starting point for a new tone is determined by a pitch track, which represents an intermediate state between frame-wise pitch and a tone object. Hereby, significant pitches are concatenated to form short continuous tracks. This intermediate processing step is required to verify uncertain pitches. Of course, it is important to start high-level tone objects promptly in order to establish timbre information. Nonetheless, there are several good reasons to postpone the tone start and at first solely track pitches in the pitch spectrogram:

- The processing of tones is very time consuming. To the contrary, pitch tracks are deliberately designed as very lightweight objects. To improve algorithm efficiency only the most promising pitch candidates should evoke new tone objects.

- Salient pitches may randomly emerge from noise. The delayed start of a tone offers the possibility to revise pitch candidates. While a salient pitch from noise might occur in one analysis frame by chance, it is quite improbable that the same will happen in the next frame in the very same frequency range.

- Although excess tone objects like octave errors can be detected and eliminated by the tone processing algorithm, the better option would be if such spurious tones are not started at all.

- In ambiguous situations (for example if many harmonically related tones start simultaneously) it is impossible to identify the most promising candidate in one analysis frame. If there are many pitches of comparable strength, the pitch tracks will be monitored for a longer time to improve the confidence of the tone onset.

When a new tone is started, an onset time has to be determined. Vos and Rasch (1981) point out that one has to distinguish between the physical and perceptual onset times of musical tones. They show that the perceptual onset time depends on the temporal envelope of the tone and occurs when the tone reaches a level of approximately 6 - 15 dB below its maximum value. In our application, the tone onset is defined as the time a tone is detected as salient pitch. So whenever a tone is started, the past pitch and magnitude values of the pitch track are transferred to the new tone and the onset time is set according to the duration of the pitch track. However, the pitch track magnitudes should not be more than 20 dB below the first tone magnitude.

5.3.1. Formation of Pitch Tracks

For the formation of pitch tracks, a subset of strong pitches is assembled in each analysis frame. A strong pitch is defined as having a magnitude which is no more than 15 dB below the maximum pitch magnitude and no more than 30 dB below the maximum tone magnitude.

For each pitch candidate, the list of existing pitch tracks is searched for an adequate link, meaning that the frequency difference must be smaller than 125 cent. If there is more than one potential link, the one with the best fitting magnitude-pitch-weighting is chosen. The weighting is determined as quotient from pitch magnitude and frequency difference in cent: $A_{\mathrm{pitch}}/(15+\Delta c)$. Any pitch candidate that cannot be linked to an existing pitch track starts a new one. Any pitch track that cannot be continued with a significant pitch is deleted. A pitch track is also deleted, when its accumulated score reaches a negative value.

Often, the most salient pitch track will again refer to the F0 of an already existing note. To avoid that new tones are started right upon an already existing tone, the magnitude threshold is higher if the pitch is close to a tone (less than 25 cent frequency distance). A similar procedure shall impede octave errors: Whenever the pitch track's frequency is about 1200 cent ($2 \cdot F0$) or 1902 cent ($3 \cdot F0$) above an existing tone, the pitch track magnitude must comply with the following condition:

$$A_{\mathrm{pitch}} > A_{\mathrm{tone}} \cdot \bar{R}_{\mathrm{freq\_deviation}}, \tag{5.2}$$

where A_{pitch} is the magnitude of the assigned pitch, A_{tone} is the tone magnitude and $\bar{R}_{\mathrm{freq\_deviation}}$ is the long-term frequency deviation rating of the tone (see Section 5.11.6 for a detailed description). The frequency deviation rating has a value between 0 and 1; a zero indicates a tone with no frequency deviation and a 1 indicates a tone with much frequency deviation, for example a tone with a pronounced vibrato. This means effectively that the more a tone is frequency modulated, the better it can prevent pitch tracks to start a new tone. The idea behind this approach is the fact, that it is very improbable that two independent tones show exactly the same frequency modulation. So the modulated tone might inhibit a new independent tone by chance, but it will not continuously do so.

In case a matching tone is found and the pitch track magnitude is not high enough, the accumulated score of the pitch track is set to zero.

With the proposed method, it can not be guaranteed that a tone object is started immediately after the tone's onset, especially, if several tones start at the same time. However, as the magnitude and the F0 of a pitch track is stored in a circular buffer for 90 ms, at least this information is available to the tone processing, when the tone is started later.

5.3.2. Rating of Pitch Tracks

The pitch track associated with a predominant pitch earns confidence points, while pitch tracks associated with a lower pitch magnitude may loose points. When a pitch track has collected a certain amount of points, it will be considered as a tone onset. If a new tone is started, the scores of the remaining pitch tracks are reset to zero.

Besides the focus on the most salient pitch, another objective should be pursued by the rating: It is desirable that very salient pitches start a new tone almost instantly, but at the same time the onset of a tone should be postponed in ambiguous situations. One common situation is the simultaneous occurrence of many candidate pitches in one analysis frame. If the number of strong pitches is very high, it might be a good option to observe the pitch tracks over a longer time period. Therefore, confidence points are awarded using two different rating schemes: A fast one, that allows a decision within a minimum time of 12 ms, and a more conservative one, that takes at least 30 ms to come to a conclusion. In ambiguous situations, both rating schemes may take a longer time to ascertain the next tone onset.

Fast Rating

For the application of the fast confidence score r_{fast}, two precondition have to apply: 1) the predominant pitch is at least 6 dB above all other pitch magnitudes and 2) the pitch magnitude should combine at least two harmonics, as there is a preference for complex tones.

In each analysis frame, the score is estimated as the quotient between the maximum pitch magnitude $A_{\text{pitch, max}}$ and a residual sum of all pitch magnitudes $i = 1, 2, ...N$

exceeding $0.3 \cdot A_{\text{pitch, max}}$ (about 10 dB below the maximum pitch):

$$r_{\text{fast}} = \frac{A_{\text{pitch, max}}}{S} \qquad \text{with} \tag{5.3}$$

$$S = \frac{1}{1-0.3} \sum_{i=1}^{N} \max\left(A_{\text{pitch, i}} - 0.3 \cdot A_{\text{pitch, max}}, 0\right).$$

In case the the predominant pitch is about 10 dB stronger than all other pitches, the score r_{fast} equals 1. The scores of individual analysis frames are accumulated.

A new tone is started if a score of at least 1.5 is reached. The fast rating is only applicable for the pitch track linked to the predominant pitch, for all other tracks the accumulated score is reset to zero.

Slow Rating

The slow rating r_{slow} evaluates pitch strength as well as the number of harmonics that are combined in a given pitch. The score is composed the following way:

- r_{slow} is initialized with 1 for a newly started pitch track.
- The pitch track linked to the predominant pitch wins 0.35 points.
- The pitch track that is linked to the pitch that combines the most harmonics wins 1 point (see also 4.6).
- A pitch track which does not score in the current analysis frame looses 0.25 points
- A pitch track with a negative score is deleted.

It is striking that the number of accumulated harmonics is more important than the actual pitch magnitude. Still, the scoring function was implemented in many different parameter combinations and this one was the best. To lay the main focus of attention on harmonic number seems to help particularly in ambiguous situations. And maybe it is not at all surprising as Terhardt (1983) denotes the same preference.

If the score is higher than 5.5, a new tone is started. If the maximum score is given to different pitch tracks over a number of frames, it might happen that no tone object is started at all. This possible disadvantage is compensated by the fact that a tone start from noise can be impeded.

5.4. Harmonics

Any complex periodic sound can be represented as a superposition of pure tones (sinusoids). A harmonic is a sinusoidal waveform whose frequency is an integer multiple of the fundamental frequency of the tone. However, in many musical instruments, this frequency relation is only proximate (e.g. string instruments), and pitched percussion instruments such as the glockenspiel, carillon and chime bars produce complex aperiodic sound waves. In the later case, we do not speak of harmonics, but use the more general term partial or sinusoidal.

In monophonic audio, each harmonic can be related to a spectral peak in the Fourier magnitude spectrum, provided that the frequency resolution is sufficient and the signal is not disturbed by noise. In polyphonic music, this relation between spectral peak and harmonic cannot be drawn so easily, because the spectrum is the result of the complex addition of several signals. Hence, the magnitude of a spectral peak is not necessarily equal to the magnitude of a harmonic, as many partials are shared by several tones. Under realistic conditions, it is not possible to recover the exact magnitudes of all harmonics from the audio mixture. So whenever we speak of the harmonic magnitude in this thesis, it should be seen as a rough approximation, or even as an informed guess.

In the presented algorithm, each harmonic is characterized by the harmonic number h and several magnitudes, which are briefly introduced below and will be explained in more detail in the subsequent sections.

- *spectral magnitude* $A_{\mathrm{h\_spec}}$: the weighted magnitude of the MR FFT magnitude spectrum at the frequency $f_h = h \cdot F0$. Please note that the magnitude is weighted (multiplied) with the frequency f_h.
- *spectral peak magnitude* $A_{\mathrm{h\_peak}}$: the magnitude of a spectral peak in the MR FFT spectrogram. Please note that the original MR FFT magnitude is weighted with the instantaneous frequency of the spectral peak.
- *long-term spectral peak magnitude* $\hat{A}_{\mathrm{h\_peak}}$: the long-term value of the spectral peak magnitude. It is only updated if a spectral peak is assigned to the harmonic.
- *harmonic magnitude threshold* T_h: this magnitude is the upper limit for the harmonic magnitude. It is also used for the calculation of the supported harmonic magnitude.
- *supported harmonic magnitude* $A_{\mathrm{h\_supported}}$: this magnitude is instantaneously estimated in each analysis frame. It is based on the spectral peak magnitude $A_{\mathrm{h\_peak}}$, which is weighted according to the support by (odd) harmonic neighbors, harmonicity, and the deviation of the tone's pitch.

- *harmonic magnitude* $\hat{A}_h$: the harmonic magnitude is the most important magnitude, as it determines the spectral envelope of the tone. It reflects the long-term evolution of the supported harmonic magnitude.

- *exclusive harmonic magnitude* $A_{\text{h\_exclusive}}$: This magnitude shows, how much of the harmonic's magnitude is covered exclusively by a tone.

- *inhibited peak magnitude* $A_{\text{peak\_inh}}$: The inhibited peak magnitude is that part of the peak's magnitude that cannot be explained by existing tones.

The magnitudes are updated in each analysis frame. Magnitudes marked with a hat or a bar depend on past values and therefore reflect the long-term evolution of the underlying values, all other magnitudes are estimated from scratch in each analysis frame.

5.4.1. Assignment of Spectral Peaks to Tones

The peaks from the MR FFT magnitude spectrum, which were detected in the frequency range between 50 Hz and 5000 Hz, have to be assigned as candidate harmonics to existing tone objects. The highest harmonic number that is registered is $h = 20$. To make the assignment computationally efficient, there has to be a way to retrieve tones by the (estimated virtual) frequency of the spectral peak. That is why pointers to the tone objects are saved in a tone map according to the expected frequency range of the tones, so that appropriate candidate harmonics can be assigned directly to the tones using the map for a fast look-up.

The assignment procedure is straight forward for the first harmonic, as the instantaneous frequency of the spectral peak has to be in the range of the predicted fundamental frequency of the tone. In order to find appropriate spectral peaks for the overtones, pairs of spectral peaks are formed and their corresponding harmonic numbers and the virtual pitches are computed using the following relationship, which was already introduced in the last chapter:

At first, it is assumed that both peaks are successive harmonics, with the instantaneous frequencies f_{low} and f_{high}. The harmonic number h_{low} of the partial with the lower frequency f_{low} is computed as:

$$\frac{h_{\text{low}}}{h_{\text{low}}+1} = \frac{f_{\text{low}}}{f_{\text{high}}} \Rightarrow h_{\text{low}} = \text{round}\left(\frac{f_{\text{low}}}{f_{\text{high}} - f_{\text{low}}}\right). \tag{5.4}$$

The harmonic number of the partial with the higher frequency is $h_{\text{high}} = h_{\text{low}} + 1$. The virtual pitch f_p is computed individually for each partial by the straightforward division of the peak's instantaneous frequency and the estimated harmonic number, $f_p = f_i/h$.

More candidate harmonics are found by also considering odd successive harmonics:

$$\frac{h_{\text{low}}}{h_{\text{low}}+2} = \frac{f_{\text{low}}}{f_{\text{high}}} \Rightarrow h_{\text{low}} = \text{round}\left(\frac{2f_{\text{low}}}{f_{\text{high}} - f_{\text{low}}}\right). \tag{5.5}$$

In this case, the computed harmonic number is valid if the rounded result is indeed an odd number. The harmonic number of the partial with the higher frequency is $h_{\text{high}} = h_{\text{low}} + 2$.

The initial distance threshold for the assignment of a candidate harmonics is defined with a relatively great margin, because the frequency of the tone cannot be predicted accurately. Depending on the frequency variability of the tone, and the last measured tone magnitude, the allowed frequency offset could be as big as 250 cent (see Section 5.9). As a result, many spectral peaks are linked to the tone object that are actually not harmonics of the tone. Furthermore, more than one spectral peak might be assigned to the same harmonic number. Hence, the real tone height has to be estimated to confirm the added candidate harmonics: the spectral peaks are rated and the resulting weights are used to compute the tone height (see Section 5.6 for a detailed explanation). After the computation of the tone height, the maximum allowed frequency distance from the ideal harmonic frequency is limited to 100 cent (one semitone). All spectral peaks that do not comply with the condition are removed. An offset of 100 cent is still high, but this threshold is well confirmed by several studies in sound segregation, for example in studies on concurrent vowel identification and mistuned harmonics (Meddis and Hewitt, 1992; de Cheveigné et al., 1997; Micheyl and Oxenham, 2010). A final judgment on the harmonic's affiliation is postponed until all shared harmonics can be evaluated jointly.

5.4.2. The Spectral Magnitudes

The spectral magnitude $A_{\text{h\_spec}}$ of a tone's harmonic is the weighted magnitude of the MR FFT magnitude spectrum at the frequency $f_h = h \cdot F0$. Thereby, the original Fourier magnitude is weighted (multiplied) with the frequency f_h. Of course, the computation of the harmonic frequency f_h is only valid if the fundamental frequency is estimated with sufficient accuracy and if the sound source has indeed a harmonic spectrum. Small offsets from the real harmonic frequency are well compensated by the algortihm, because the frequency resolution of the MRFFT decreases for higher frequencies. As the main lobe of the spectral peak becomes wider, the spectral magnitude in neighboring MRFFT frequency bins still approximates the peak magnitude. The spectral magnitude $A_{\text{h\_spec}}$ comes into play whenever no spectral peak can be found at a harmonic frequency. Then, it is assumed that the spectral peak at this frequency is masked by a concurrent sound.

The spectral peak magnitude $A_{\text{h\_peak}}$ refers to the magnitude of a spectral peak in the MR FFT spectrogram, which is assigned to the harmonic h. Again, the original

MR FFT magnitude is weighted with the instantaneous frequency of the spectral peak. The spectral peak magnitude is the highest magnitude that can be reached by a harmonic. Usually, however, the reached harmonic magnitude is smaller than the spectral peak magnitude, because of several weightings – like harmonicity or the smoothness of the spectral envelope. And of course, if several tones share a harmonic (and therefore a spectral peak), the magnitude of the spectral peak has to be partitioned among all tones.

A long-term estimate of the spectral peak magnitude is stored for each of the tone's harmonics, namely $\hat{A}_{\text{h\_peak}}$. Preferably the long-term value should instantly update to the actual peak magnitude, however, only in certain margins. So in each analysis frame, an upper and a lower limit for the magnitude is defined, $\hat{A}_{\text{peak}} \cdot 1.09$ and $\hat{A}_{\text{h\_peak}} \cdot 0.92$, respectively. Within these margins the long-term spectral peak magnitude is updated instantly to the new value. If the current spectral peak magnitude A_{peak} is outside the margins, $\hat{A}_{\text{h\_peak}}$ is updated to the next best margin. If $\hat{A}_{\text{h\_peak}}$ is very small, it takes a long time, until it reaches the value of the actual peak magnitude. That is why the EMA is used to compute a second value, which is applied whenever it is bigger than the calculated upper margin:

$$\hat{A}_{\text{h\_peak}} \leftarrow \alpha \cdot \hat{A}_{\text{h\_peak}} + (1 - \alpha) \cdot A_{\text{h\_peak}}, \tag{5.6}$$

where the parameter α refers to the half-life times specified in Section 5.4.5. $\hat{A}_{\text{h\_peak}}$ is initialized with the peak magnitude. In case there is no peak assigned as harmonic, the long-term peak magnitude is updated according the magnitude deviation of the last frame, in order to mimic a mutual amplitude modulation of the harmonics.

Now, when a spectral peak is assigned to a tone object, the long-term harmonic peak magnitude $\hat{A}_{\text{h\_peak}}$ associated with the tone's harmonic can be assessed and compared to the current peak magnitude. This information can be used later to rate all spectral-peak-to-harmonic assignments jointly, for example, to weight the importance of a spectral peak during the estimation of the tone height. In addition, the ratio between current peak magnitude and long-term peak magnitude can be utilized to detect masking.

5.4.3. Harmonic Magnitude Threshold

The harmonic magnitude threshold T_h represents the upper limit for the harmonic magnitude $\hat{A}_h$, so whenever $\hat{A}_h$ is greater than T_h, $\hat{A}_h$ is set to this value. T_h is calculated after the estimation of the tone height. For its calculation, two different cases are distinguished:

$$T_h = \begin{cases} A_{\text{peak}} \cdot \exp\left(-\frac{\ln(2)}{90^2}\, d_h^2\right), & \text{if spectral peak} \\ 0.4 \cdot A_{\text{h\_spec}}, & \text{else.} \end{cases} \tag{5.7}$$

The variables $A_{\mathrm{h\_peak}}$ and $A_{\mathrm{h\_spec}}$ are the magnitude of the assigned spectral peak and the magnitude in the weighted MR FFT magnitude spectrum at the ideal harmonic frequency $h \cdot F0$, respectively. The variable d_h denotes the offset of the spectral peak's instantaneous frequency from the ideal harmonic frequency (see equation 5.11) measured in cent. The gaussian function, which is used for weighting the spectral peak, is precomputed with a one-cent resolution.

5.4.4. Supported Harmonic Magnitude

The supported harmonic magnitude $A_{\mathrm{h\_supported}}$ is used to update the long-term harmonic magnitude A_h. It is calculated for each harmonic, whenever a spectral peak is assigned. If no spectral peak was assigned as a harmonic, the corresponding long-term harmonic magnitude is not increased and therefore the estimation of the spectral support magnitude is not necessary.

Two central ideas concerning the supported harmonic magnitude have already been introduced for the pitch determination algorithm (PDA), namely the smoothness of the spectral envelope and the harmonicity of the partials. Both features are evaluated in order to limit the maximum harmonic magnitude which can be attributed to the tone. This means $A_{\mathrm{h\_supported}}$ might be considerably lower than the magnitude of the respective spectral peak A_{peak}. However, compared to the strict requirements of the PDA, the conditions have been relaxed for the tone processing. This means that once a tone object is successfully established, there is a strong tendency to fuse all available overtones into one unitary percept, even if there are some steep formants, or if the harmonic frequencies deviate from the ideal integer multiples of the F0. And unlike in the PDA, intermediate spectral peaks (ie. any peaks that can be found between two neighboring harmonics) do not decrease the supported magnitude.

In addition, a third measure is introduced to eventually boost the magnitude of $A_{\mathrm{h\_supported}}$: if the tone has a varying fundamental frequency and at the same time a low F0 prediction error, the supported magnitude is increased noticeably. The three features mentioned above are described in the upcoming sections, as well as the final calculation of $A_{\mathrm{h\_supported}}$.

Frequency Modulation

Various researchers have discussed the possibility that the coherence of frequency modulation (FM) could be used as a cue for the grouping of simultaneous auditory components (Bregman, 1994; Darwin et al., 1994; Chowning, 1999). Quite surprisingly, experiments revealed that coherent FM modulation has a very limited effect on the differential discrimination of sounds in the human auditory system. For example, a varying F0 does not improve vowel identification (Sundberg, 1975; Darwin and Sandell, 1995), and differences in the modulation rate are apparently not used by humans to segregate tones (Darwin et al., 1994). Carlyon (1994) showed that most positive effects attributed to coherent FM can be explained by the harmonicity criterion described above, as any harmonic which does not obey the frequency deviation implied by the fundamental frequency of the tone, will finally cease distant from the ideal harmonic position.

However, there is also some evidence for positive effects. Darwin et al. (1994) found that FM modulation facilitates the integration of a mistuned harmonic into the overall percept of a complex tone, as the mistuned harmonic had a greater impact on the overall pitch in the modulated tone. (Matthews, 1999), who is an expert for the synthesis of musical sounds, stated that a mixture of periodic and random frequency variation is needed to give a musical quality to long tones, an opinion that is shared by Chowning (1999), who confirmed that vibrato and random noise enhances the perceptual integration of timbre with strong formants.

While at the one hand, a preferably smooth spectral envelope is desirable, at the other hand timbres with sharp resonances at formant frequencies do occur. Yet normally, a steep rise of 20 dB in magnitude from one harmonic to the next is a good indicator for the presence of another sound source, and moreover, there are complex tones with steep formants which are perceived as two simultaneous voices (throat singing, overtone singing), so the application of the spectral smoothness criterion is reasonable. A possible explanation for the different perception of steep formants could be a marked deviation of the tone's F0.

The implemented frequency deviation criterion does not evaluate the frequency modulation of individual harmonics, but rather acknowledges whether there is a significant frequency variation in the tone's F0 or not. If the fundamental frequency varies, the spectral envelope is allowed to have pronounced peaks. The idea is that an amplification factor is applied during the computation of the spectral support magnitude $A_{\text{spec\_support}}$, which is described in the next section. In case a partial belongs to another sound source (and the amplification factor is applied by mistake), it can be assumed that an erroneous boost of the supported magnitude is confined to a small time period, because a coherent frequency modulation of different sound sources is improbable.

In spite of the simplicity of this idea, it was difficult to turn it into a handy quantitative routine. Finally, a boost factor w_{FM} for FM tones was implemented, which more or less weakens the smoothness principle defined in the subsequent section. At first, different amplification factors were evaluated for varying degrees of frequency modulation. Results did not improve significantly compared to a baseline approach which distinguishes stable and modulated tones:

$$w_{\mathrm{FM}} = \begin{cases} 10 & \text{if varying F0,} \\ 2.5 & \text{else.} \end{cases} \tag{5.8}$$

The exact definition for tones with a varying F0 is given in Section 5.11.6.

Smoothness of the Spectral Envelope

The smoothness of the spectral envelope is an important feature which helps to decide if a spectral peak is an actual harmonic of the tone. When de Cheveigné et al. (1997) asked listeners to identify concurrent synthetic vowel pairs, a difference of 6% in F0 improved their performance. But the performance was improved further, when the amplitude of the target vowel was below that of a competing vowel (-10 and -20 dB). A possible explanation is that the magnitude differences between harmonics are more pronounced, so that it becomes easier to identify and inhibit the harmonics of the predominant vowel. Of course, sounds have different levels in musical recordings, but moreover, tones from different sound sources tend to predominate different parts of the frequency spectrum. We found that a measure for the local smoothness of the spectral envelope is an important cue to segregate simultaneous sounds.

So the first step towards the computation of the supported harmonic magnitude is the estimation of the spectral support $A_{\mathrm{spec\_support}}$, which quantifies the spectral smoothness around a certain harmonic. Any harmonic (expect the first one) has to be supported by a significant magnitude at neighboring harmonic frequencies in order to induce a virtual pitch at the fundamental frequency of the tone. This requirement has already been formulated within the pitch determination algorithm in Section 4.5.4.

In the tone processing, the spectral smoothness is evaluated in a very similar way, yet, there is one notable difference: neighboring spectral peaks (sinusoidal partials) are not needed – if no appropriate neighboring peaks are detected in the spectrum, we simply take the harmonic magnitude threshold T_h as a replacement (see Section 5.4.3). This means, for the detection of a new tone, neighboring spectral peaks at the appropriate harmonic frequencies are mandatory, yet if the tone is already started, there are a number of mechanisms which keep the tone alive even in case of masking.

For the first harmonic, the spectral support $A_{\mathrm{spec\_support}}$ is always the magnitude of the spectral peak itself. For all other harmonics, it is calculated as follows: The harmonic magnitude thresholds T_{h-1} and T_{h+1} of the lower and upper neighboring harmonics determine – together with the computed frequency modulation weight

w_{FM} – the low and the high frequency support $A_{\mathrm{support\_low}}$ and $A_{\mathrm{support\_high}}$, respectively.

$$\begin{aligned} A_{\mathrm{support\_low}} &= \min(w_{FM} \cdot T_{h-1}, T_h) \\ A_{\mathrm{support\_high}} &= \min(w_{FM} \cdot T_{h+1}, T_h) \end{aligned} \tag{5.9}$$

The harmonic magnitude threshold is either the spectral peak magnitude, or – in case no spectral peak was assigned – the damped spectral magnitude at the ideal harmonic frequency (see Section 5.4.3). Then, the spectral support $A_{\mathrm{spec\_support}}$ is a weighted sum of $A_{\mathrm{support\_low}}$ and $A_{\mathrm{support\_high}}$:

$$\begin{aligned} A_{\mathrm{spec\_support}} = &\, 0.3 \max\left(A_{\mathrm{support\_low}}, A_{\mathrm{support\_high}}\right) \\ &+ \min\left(A_{\mathrm{support\_low}}, A_{\mathrm{support\_high}}\right). \end{aligned} \tag{5.10}$$

$A_{\mathrm{spec\_support}}$ must not be greater than T_h and is decreased to this value if necessary. In case the examined harmonic number is odd, the harmonic magnitude thresholds at neighboring odd harmonic frequencies are also considered. For example, if the current harmonic number is 5, we look at the thresholds for the harmonic numbers 4 and 6, as well as 3 and 7. The second order neighbors replace the values found in step one, if their values are bigger.

Harmonicity Weighting

The harmonicity weighting is related to the problem of the mistuned harmonic. Here, the question is, at which amount of mistuning a harmonic “pops out” from the complex, and is heard as separate tone. Several studies showed that one starts to hear out a mistuned harmonic at a deviation of 1% to 2% from the ideal harmonic frequency (Moore et al., 1986; Hartmann, 1997; Lin and Hartmann, 1998), however, the mistuned harmonic continues to contribute to the pitch of the complex for mistunings of up to 8%, with the maximum effect at 3%. Now, the goal of this section is to integrate these findings into a computational model that works well on polyphonic input.

After the tone’s fundamental frequency has been estimated, the offset between the instantaneous frequency of the spectral peak and the ideal harmonic frequency can be calculated. If the spectral peak was assigned to several tone objects, the minimum offset from the ideal harmonic frequency can be used to judge each assignment. Hereby, it is important to understand that the frequency offset is always interpreted relative to the best assignment of the current peak. This way very large deviations from the ideal harmonic frequency remain possible, as long as no better option is available.

For the harmonicity weighting, the virtual pitch of the harmonic has to be compared to the fundamental frequency of the tone. So it is clear that the harmonicity weighting can only be computed, after the F0 of the tone was calculated (see Section 5.6). A spectral peak represents the harmonic of a tone well, if its instantaneous frequency is close to an integer multiple of the F0 of the tone. The offset d_h from the ideal harmonic frequency is measured in cent and calculated using

$$d_h = c_h - 1200 \cdot \log_2(h) - c_{\text{tone}}, \tag{5.11}$$

in which c_h is the instantaneous frequency of the harmonic (or rather, the IF of the assigned spectral peak) and c_{tone} is the fundamental frequency of the tone. Again, both frequencies are measured in cent. If the estimated offset d_h is bigger than 100 cent, the harmonic candidate is ruled out. This corresponds to a mistuning of about six percent.

The bigger the difference between the ideal harmonic frequency and the actual harmonic frequency, the lower is the harmonicity weighting. But there are two more features which have an impact on the perceived harmonicity. First, the overall harmonicity of the tone: if the audio signal is very noisy in general or if there is a systematic deviation from the ideal harmonic frequencies, a frequency offset of an individual harmonic is not as important. Second, the frequency offset is compared to the minimum harmonic frequency offset that is reached for the spectral peak – another example for the joint evaluation approach. We combine the three different offset measures and utilize a Gaussian bell curve to compute the final harmonicity weighting:

$$w_d(d) = \exp\left(-\frac{\ln(2)}{57 \cdot 57}\, d^2\right), \tag{5.12}$$

with

$$d = d_h + 2(d_h - d_{\text{peak,min}}) - \bar{d}_{50\text{ms}},$$

where d_h is the cent offset, $d_{\text{peak,min}}$ is the minimum frequency offset from the ideal harmonic position found for the corresponding spectral peak, and $\bar{d}_{50\text{ms}}$ is the long-term average offset from the ideal harmonic position for all harmonics of a tone (see Section 5.11.5). The width of the Gaussian function and the proportion of the different features in the calculation of d have been found empirically.

Combining the Weights: the Supported Harmonic Magnitude

Putting it all together, the supported harmonic magnitude amounts to:

$$A_{\text{h\_supported}} = w_d(d) \cdot A_{\text{spec\_support}}, \tag{5.13}$$

in which $w_d(d)$ is the harmonicity weighting and $A_{\text{spec\_support}}$ denotes the spectral support by the magnitude of the neighboring harmonics. In case $A_{\text{h\_supported}}$ is bigger than the harmonic magnitude threshold T_h, $A_{\text{h\_supported}}$ is set to T_h.

The supported harmonic magnitude is quiet error-prone, because the underlying support rating is often affected by noise and other sound sources. That is why the momentary value within one analysis frame does by no means allows a certain statement about the harmonic magnitude. Still, the harmonic magnitude can be estimated with some reliability if several measures are combined over time. At this point the long-term harmonic magnitude comes into play.

5.4.5. Harmonic Magnitude

The harmonic magnitude $\hat{A}_{\mathrm{h}}$ is the representative magnitude for the harmonic, because it determines the spectral envelope. The sum of all harmonic magnitudes constitutes the tone's magnitude. The harmonic magnitude adapts to the frame-wise estimated magnitude $A_{\mathrm{h\_supported}}$. For that matter, two questions become important: how is the magnitude of shared harmonics partitioned among tones, and how is the long-term harmonic magnitude updated.

Joint Evaluation

There are two different ways to partition the magnitude of shared harmonics among several tone objects: first, the spectral peak magnitude could be distributed among all possible sources, in a way that the sum of all harmonic magnitudes does not exceed the spectral peak magnitude. Second, the entire spectral magnitude could be assigned to the tone with the best support for the harmonic.

In our system, the spectral magnitude is distributed among all sources. Whenever the sum of the harmonic magnitudes exceeds the spectral peak magnitude, the update of the harmonic magnitude is suspended, because the sum of all harmonic magnitudes must not exceed the magnitude of the the spectral peak. This way, the "Old-Plus-New" heuristic mentioned before becomes relevant for the update of the harmonic magnitudes: since an already established tone maintains its harmonic magnitudes, the recently started tone can only profit from spectral energy that exceeds the harmonic magnitudes of existing tones.

The exact partition of shared harmonics depends on many factors: the ideal harmonic frequency, the spectral smoothness, the temporal evolution of the harmonic magnitude, but it also depends on the effects of onset asynchrony and possible masking. The final partition of the harmonic magnitudes emerges rather by a continuous adaptation over many analysis frames, than by an instant computation according to some formula.

Update of the Harmonic Magnitude

The aim is to allow a preferably quick adaptation of the magnitude to the instantaneous magnitude of the partials, but still diminish the impact of spurious spectral peaks and coinciding harmonics. Any increase of the harmonic magnitude takes place rather slowly. To the contrary, a decrease happens instantly: whenever the harmonic magnitude is bigger than the supported magnitude $A_{\mathrm{h\_supported}}$, it is immediately reduced to the smaller value.

The harmonic magnitude is increased, whenever the supported magnitude $A_{\mathrm{h\_supported}}$ is greater than the harmonic magnitude. There are two methods to control the adaptation speed for a rising magnitude: the first method defines an upper margin for the updated magnitude, the second method is based on an EMA calculation using several smoothing factors. For each update both values are calculated. The bigger value is chosen as new harmonic magnitude.

Method 1: The upper margin T_u is estimated relative to the last harmonic magnitude: $T_u = 1.09 \cdot \hat{A}_{\mathrm{h}}$. Within the interval $[\hat{A}_{\mathrm{h}}, T_u]$ the harmonic magnitude $\hat{A}_{\mathrm{h}}$ is updated immediately to the new value, but whenever the supported harmonic magnitude exceeds the margin, $\hat{A}_{\mathrm{h}}$ is set to T_u .

Method 2: The second method involves the calculation of the exponential moving average with variable half life factors. Thereby, the duration of a tone has an effect on the adaptation speed of the harmonic magnitudes (see Table 5.1). During the attack of the tone the spectral envelope is usually subject to marked fluctuations until a more stable waveform is established. During this phase method 1 is not able to adapt fast enough to the true harmonic magnitude. Hence, a quick adaptation of the spectral envelope is made possible in the beginning until – after the collection of several measurements – a more realistic picture of the tone's spectral envelope is achieved.

condition	half life period
if duration $<$ 100 ms	15 ms
else if duration $<$ 200 ms $\vee$ is FM tone	25 ms
else	1.0 s

Table 5.1. Variable Half Life Periods for Harmonic Magnitude Estimation

Table 5.1 shows an overview of the different half life periods used under several conditions. It is shown that during the first 100 ms of the tone's duration the update is rather quick – the half life factor for the EMA calculation is only 15 ms. During the second 100 ms of the tone's duration, the adaptation speed is more moderate. After 200 ms the second method prohibits a fast adaptation – with a half life factor of 1 second a sudden energy boost cannot be captured by this method. Due to the slow update of the harmonic magnitude a sudden rise in energy will induce a new salient pitch, which is detected by the pitch determination algorithm and will eventually induce a new tone onset. But there is an exception to the rule: All tones which are frequency modulated retain the aptitude for a moderately fast harmonic magnitude update.

5.4.6. Exclusive Harmonic Magnitude

Even though much care is taken to validate the onset of tones, sometimes tones are started by mistake. If tone objects are induced by noise, they usually cannot build up a high energy level, since there is no periodic spectral pattern that matches their fundamental frequency. After some time, such spurious tones are identified because of their low magnitude and their irregular pitch. However, tones that are actually octave errors have a regular magnitude and pitch. The exclusive magnitude can be used to detect and eliminate such tones.

The idea is to monitor how many harmonics of a tone are shared with other tones or – putting it the other way round – which harmonics are explained exclusively by the examined tone. However, there is no binary discrimination between the states *shared* and *exclusive*, but rather a careful evaluation of the harmonic magnitudes. This ensures that tones at integer multiples of a fundamental frequency remain possible.

If a spectral peak is assigned to more than one tone, we speak of a shared harmonic. If a tone shares all of its harmonics with other tones, it might be an octave error. For this reason, it is interesting to asses how much of the peak's magnitude is exclusively covered by one tone. The exclusive harmonic magnitude $A_{\mathrm{h\_exclusive}}$ answers this question. It is calculated as follows: for each peak, the harmonic support magnitude $A_{\mathrm{h\_supported}}$ is estimated for all tones as described in Section 5.4.4. Then, for all harmonics that share the examined peak, the harmonic magnitudes are summed up as follows:

$$S = \sum_{h}^{\text{shared harmonics}} \min(1.5 \cdot \hat{A}_{\mathrm{h}}, A_{\mathrm{h\_supported}}). \tag{5.14}$$

The sum is stored for each peak. After the assignment has finished, the stored sum S denotes how much this peak was "used" by existing tones. Then, the exclusive harmonic magnitude $A_{\mathrm{h\_exclusive}}$ for an individual harmonic is calculated as follows:

$$A_{\mathrm{h\_exclusive}} = \max(0, \quad \min(1.5 \cdot \hat{A}_{\mathrm{h}},\, A_{\mathrm{h\_supported}}) - \max(S - A_{\mathrm{peak}},\, 0)) \tag{5.15}$$

The exclusive magnitude of the tone $A_{\mathrm{exclusive}}$ is the sum of all its harmonic's exclusive magnitudes. A tone with a very low exclusive magnitude is probably not significant and can be deleted.

5.4.7. Masking

Due to the complexity of polyphonic music, it is highly probable that any particular tone is affected by concurrent sounds. Gómez et al. (2003) have compiled an overview of different methods to minimize the influence of masking in the final melody pitch contour, including typical post-processing techniques such as low-pass or median filtering, the reanalysis over past frames as well as dynamic programming techniques. Paiva (2006) fills in missing frequency values by linear interpolation to get continuous tone trajectories.

In the proposed tone tracking algorithm, masking of individual harmonics is detected, if the current spectral peak magnitude $A_{\mathrm{h\_peak}}$ is at least two times higher than the long-term spectral peak magnitude $\hat{A}_{\mathrm{h\_peak}}$. If no peak was assigned to a harmonic, the spectral magnitude $A_{\mathrm{h\_spec}}$ at the notional harmonic frequency $f_h = h \cdot F0$ is investigated instead.

The harmonic magnitude $\hat{A}_{\mathrm{h}}$ is only updated, if no masking is detected. If more than forty percent of the harmonic magnitudes of the tone's harmonics are masked, that is

$$\sum_{h=1}^{\mathrm{masked}} \hat{A}_{\mathrm{h}} \,/ \sum_{h=1}^{\mathrm{all}} \hat{A}_{\mathrm{h}} > 0.4, \tag{5.16}$$

the tone is "frozen". This means that no harmonic is updated, unless the harmonic magnitude threshold T_h is lower than the harmonic magnitude $\hat{A}_h$. In this case the harmonic magnitude is of course set to the estimated threshold.

5.4.8. Inhibition of Spectral Peaks

The inhibited peak magnitude $A_{\mathrm{peak\_inh}}$ is that part of the peak's magnitude that cannot be explained by existing tones. Let us revise the process, which finally leads to the inhibition of spectral peaks:

- At first, candidate partials (spectral peaks) are added to those tones which are close to the estimated (virtual) frequency of the partials, which is determined by the assumed harmonic relation of spectral peak pairs.
- The tone height is computed with help of the assigned spectral peaks.
- For each tone, the spectral peaks with a matching (virtual) frequency are chosen as harmonics from its set of candidate harmonics.
- The long-term spectral envelope of the tone is updated.
- Now the inhibition magnitude is estimated for each spectral peak, and finally

the pitch determination algorithm computes the predominant pitch of the current analysis frame based on the inhibited spectral peaks.

The inhibition of the spectral peak is computed as follows: for each spectral peak, the harmonic magnitudes $\hat{A}_h$ of all tone objects, which share that spectral peak, are summed:

$$A_{\text{peak\_inh}} = A_{\text{peak}} - \min\left(\sum_{i}^{\text{num tones}} \hat{A}_{h,i}\,, \quad A_{\text{peak}} \right). \qquad (5.17)$$

If the sum is greater than the spectral peak's magnitude A_{peak}, the inhibition equals the peak magnitude. In this case, $A_{\text{peak\_inh}}$ is zero – the whole spectral peak magnitude is explained by existing tones. Thus, the spectral peak does not appear in the subsequent pitch determination.

5.5. Tone Magnitudes

The tone magnitude is a measure that should correlate with the perceived loudness of the tone. Loudness is a subjective quality of a sound that is the psychological counterpart of objective physical measures like sound pressure, sound pressure level, sound intensity or sound power.

Within the melody extraction system, the term loudness is deliberately avoided, because it is impossible to say at which volume the analyzed music file is played back. From this it follows that some parameters that affect loudness perception cannot be used. An example for this constraint is the absolute threshold of hearing, which can obviously only be applied if the sound pressure level of the signal is known. As the proposed system is independent of the recording level, even badly balanced audio files can be analyzed, provided that the signal-to-noise ratio is sufficient and there is no signal distortion by means of clipping.

There are four magnitude measures that reflect the perceptual importance of the tone, namely magnitude, supported magnitude, exclusive magnitude and salience. It might seem inconvenient to manage four measures instead of only one, but the parameters all describe distinct perceptual concepts and are tailored to fulfil different tasks. All parameters used to compute the different magnitudes have been found empirically.

5.5.1. Magnitude

The tone's magnitude can be seen as the equivalent to loudness in human auditory perception, with the distinction that the computational model is highly simplified and no information about absolute levels can be provided. The tone magnitude should not be affected by the existence of other tones and in addition should also include energy from presumably masked components.

The current magnitude of the tone A_{tone} is simply the sum of the harmonic magnitudes:

$$A_{\text{tone}} = \sum_{h=1}^{20} \hat{A}_h \tag{5.18}$$

The long-term magnitude $\bar{A}_{\text{tone, 100ms}}$ is described by the exponential moving average (EMA) of the past magnitude values (see also appendix A):

$$\bar{A}_{\text{tone, 100ms}} \leftarrow \alpha \cdot \bar{A}_{\text{tone, 100ms}} + (1 - \alpha) A'_{\text{tone}} \tag{5.19}$$

in which $\bar{A}_{\text{tone, 100ms}}$ is initialized with 0. The parameter α, which is estimated using equation A.5 and the half-life time $t_{\text{HL}} = 100\text{ms}$, denotes the smoothing factor and determines how fast the magnitude is converging to the most recent measurements.

Since the start value for the iterative formula is zero, the long-term magnitude of the tone builds up gradually. An 100 ms half-life period causes a very slow build up – too slow on many occasions, so in general, the corrected long-term magnitude $\bar{A}^*_{\text{tone, 100ms}}$ is used, where the actual EMA-value is multiplied with a scaling factor that corresponds to the EMA of the past α-weights (see equations A.3 and A.4 in the appendix). This way $\bar{A}^*_{\text{tone, 100ms}}$ rises as fast as the instantaneous tone magnitude in the very beginning, but later on (as the sum of weights approaches one) remains more steady.

The proposed method to calculate the long-term magnitude is reminiscent of the leaky integrator model described by Hartmann (1997, S.74): The temporal integration relates stimulus duration and perceived intensity, that is, a sound of constant sound pressure level will be perceived with increasing loudness after a duration of 50, 100 or 200 ms. Of course the loudness is not going to rise for good. After approximately 1 second no further increase in the loudness sensation will be noted. If the tone intensity changes over time, the moment by moment loudness perception is based on the integration of the preceding 600-1000 ms.

The tone's magnitude is exploited on several occasions:

- It is utilized to measure the importance of a tone, especially in the identification of musical voices and in the establishment of magnitude thresholds (see sections 6.3.2 and 6.3.3).
- Tone offsets can be detected by the comparison of instantaneous and long-term magnitude (see Section 5.8).
- The magnitude of a tone is used for salience estimation (see Section 5.5.4).

- The maximum tone magnitude is an important reference to identify excess tones (see Section 5.8.4).

5.5.2. Supported Magnitude

The supported tone magnitude $A_{\text{tone\_supported}}$ is the sum of the harmonic magnitudes $\hat{A}_h$, but only for those harmonics, which have an assigned spectral peak in the current analysis frame:

$$A_{\text{tone\_supported}} = \sum_{h=1}^{20} \min(\hat{A}_h, A_{\text{h\_peak}}) \tag{5.20}$$

If no appropriate spectral peak can be found, the supported magnitude is zero for that very harmonic, even if the spectral magnitude suggests that it might be masked. As a consequence, the supported tone magnitude can be used to detect masking (see Section 5.8.4).

5.5.3. Exclusive Magnitude

The exclusive tone magnitude $A_{\text{tone\_exclusive}}$ shows which proportion of the magnitude A_{tone} is covered exclusively by the considered tone. For example, if most of the tone's harmonics are shared with other tones, the exclusive magnitude is low. If however, no harmonic is shared, the exclusive magnitude equals the supported magnitude $A_{\text{tone\_supported}}$.

The exclusive magnitude of the tone is the sum of its exclusive harmonic magnitudes (see Section 5.4.6):

$$A_{\text{tone\_exclusive}} = \sum_{h=1}^{20} A_{\text{h\_exclusive}} \tag{5.21}$$

It is used to identify and delete excess tones that may emerge due to octave errors in the pitch estimation algorithm (see Section 5.8.4).

5.5.4. Salience

The tone's salience is the perceptual importance or noticeability of a given tone within the sonority. An important aspect of salience is, that the significance of an item is always seen relative to any neighboring items.

Huron (2001) stated on salience:

> The mere presence of some element or property does not necessarily make it a good feature. A good feature must in some ways draw attention to itself. It must be notable, or what might be dubbed salient.

Of course the salience is dependent on the magnitude in the first place. So one aspect of the tone's salience sets the magnitude A_{tone} in relation to the magnitudes of other tones in the spectral neighborhood. But there are other features that increase the significance of a particular tone compared to occurrences of other sounds. One feature is for example frequency modulation – a tone played with vibrato is very noticeable and more likely to belong to the melody voice (Tachibana et al., 2009). Yet, even characteristics of tone color have an influence on the perceptual importance. For example, signals with eminent energy levels in the high frequency bands seem to attract the attention of the listener.

The computation of the tone's salience is also linked to the detection of musical voices. If human listeners shall analyze rich sonorities of simultaneously playing notes (e.g. a chord) often only the highest and the lowest notes can be made out with reliability. This observation can be generalized to the perception of concurrent voices in polyphonic music – as humans usually observe the outer voices as being predominant, even if the tones forming the inner voices have a comparable strength.

In our algorithm, two salience based measures are exploited: first, the frequency deviation rating described in Section 5.11.6 is used to identify frequency modulated tones. Second, the salience rating r_{salience} depicts whether or not the tone is surrounded by concurrent tones. It is computed as follows: the salience of a tone is decreased, if the frequency distance Δc between the tone and another tone is less than 1300 cent and greater than 50 cent. To compute the exact salience factor, all tones within the given frequency distance are weighted and summed according to the following formula:

$$S_{\text{salience}} = \sum_{\text{tone}=1}^{\text{num tones}} A_{\text{tone}} \cdot \left(0.2 + 0.8 \cdot e^{-\frac{(\Delta c)^2}{2 \cdot 640^2}} \right). \tag{5.22}$$

Then, the salience rating r_{salience} amounts to

$$r_{\text{salience}} = \frac{A_{\text{tone}}}{S_{\text{salience}} + A_{\text{tone}}}, \tag{5.23}$$

where A_{tone} is the current tone magnitude described in Section 5.5.1. Finally, the salience of the tone constitutes the salience rating multiplied with the current magnitude.

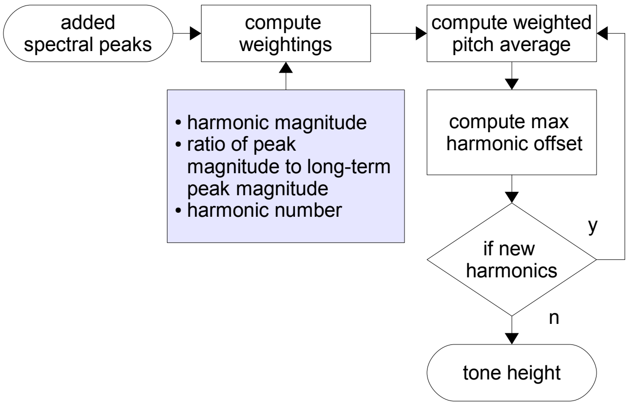

added spectral peaks
compute weightings
compute weighted pitch average
• harmonic magnitude
• ratio of peak magnitude to long-term peak magnitude
• harmonic number
compute max harmonic offset
if new harmonics
y
n
tone height

Figure 5.2. Overview of the Tone Height Estimation

5.6. Frame-wise F0 Update

As described in Section 5.4.1, candidate harmonics are assigned to tone objects whenever the harmonic's spectral/virtual pitch lies within the predicted pitch range of the tone. Since the peaks are collected from a relatively big frequency area, simply taking the average of all cent values does not lead to an adequate estimation of the tone height, because outliers are quite common. Anyhow, the low acceptance threshold during the assignment of peaks is necessary, because the tone's pitch cannot be predicted precisely. This leads to to the inclusion of many peaks which are not actually harmonics of the respective tone object. The aim of the proposed tone height estimation is to identify erroneously added peaks and exclude them from the pitch calculation.

An overview of the implemented approach can be seen in Figure 5.2. The iterative computation of the tone height is implemented using a weighted mean, which helps to factor out the most common errors. The weighted mean is similar to an arithmetic mean, however, instead of each of the data points contributing equally to the final average, some data points contribute more than others. The weightings of the frequencies depend on long term timbre features, which are better indicators of the reliability of peaks than the peak magnitude.

There are three factors that determine the weight of the harmonic within the average:

- *harmonic magnitude:* The weight is depending on the harmonic magnitude $\hat{A}_h$ described in Section 5.4.5. The spectral envelope of a tone is established over a longer time period. Any harmonic which has aggregated a high magnitude in previous analysis frames shall become significant in the weighted mean. This way the impact of noise peaks or partials belonging to other tones can often be diminished, because such peaks normally do not succeed to establish a high harmonic magnitude.

Experience has shown that it is beneficial to use the square root of the harmonic magnitude as weight, in order to spread the contribution more among all harmonics.

- *ratio of harmonic peak magnitude to long-term harmonic peak magnitude:* If the current spectral peak magnitude $A_{\text{h\_peak}}$ differs significantly from the average spectral peak magnitude of previous analysis frames $\hat{A}_{\text{h\_peak}}$, the current spectral peak might originate from a different sound source. It is advisable to handle such peaks with caution and diminish their impact. The ratio of harmonic peak magnitude and long-term harmonic peak magnitude is used as a weighting factor r_r, which is defined as

$$r_r = \begin{cases} A_{\text{h\_peak}}/\hat{A}_{\text{h\_peak}}, & \text{if} \quad A_{\text{h\_peak}} < \hat{A}_{\text{h\_peak}} \\ \hat{A}_{\text{h\_peak}}/A_{\text{h\_peak}}, & \quad \text{else,} \end{cases} \tag{5.24}$$

  in which $\hat{A}_{\text{h\_peak}}$ denotes the long term peak magnitude of the harmonic and $A_{\text{h\_peak}}$ denotes the weighted magnitude of the currently assigned spectral peak (see Section 5.4.2).

- *harmonic number:* The harmonic impact on the weighted average also depends on the harmonic number. In general the estimated instantaneous frequency is more precise for lower harmonics, because they have a better signal to noise ratio and they are better separated from other harmonics in the multi resolution spectrogram. Beyond that, the assigned harmonic number of the higher partials is often faulty, because a small variation in the estimated IF has a strong effect on the calculation of the harmonic number. From this it follows that the lower harmonics receive a greater weight than the high harmonics. The relation between harmonic impact and the harmonic number h is given by the following equation:

$$r_h = 0.4 + 0.6 \cdot \exp\left(-\frac{h^2}{2 \cdot 3^2}\right). \tag{5.25}$$

Finally, the weighting of the partial within the weighted average is estimated by:

$$w_h = r_h \cdot r_r \cdot \sqrt{\hat{A}_h}. \tag{5.26}$$

If no partial with the current harmonic number was added to the tone before, the harmonic magnitude is zero. In this case the weighting is set to a small percentage of the actual peak magnitude.

Subsequently, the average offset between the harmonic's virtual pitch and the estimated tone height is estimated. The maximum allowed frequency offset is set to

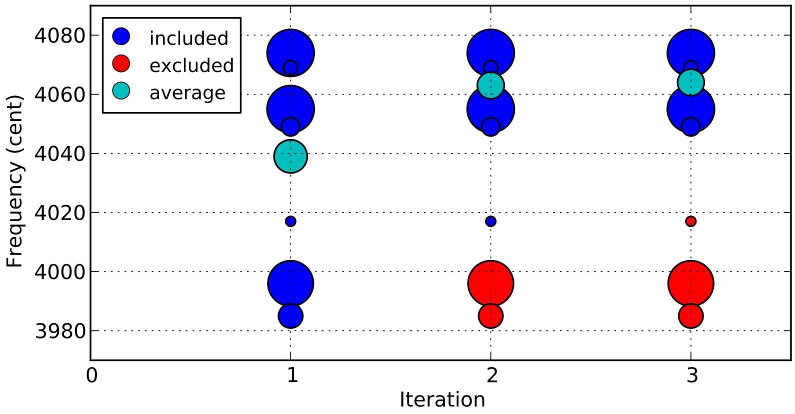


Figure 5.3. Iterative Tone Height Estimation

4/3 of the average frequency offset. However, the threshold must not drop below the minimum distance threshold of 35 cent. Now, outliers (e.g. harmonics with a frequency distance that is greater than the estimated threshold) are excluded from the tone height estimation and a new weighted average c_{tone} is calculated:

$$c_{\text{tone}} = \frac{\sum w_h c_v}{\sum w_h}, \tag{5.27}$$

in which w_h denotes the estimated weighting, and c_v denotes the harmonic's (virtual) pitch. The virtual pitch c_v (measured in cent) is computed from the harmonic's frequency f_h (measured in Hertz) as

$$c_v = 1200 \log_2 \left(\frac{f_h}{h f_{\text{ref}}} \right), \tag{5.28}$$

in which h is the harmonic number. In our algorithm, the reference frequency f_{ref} is set to the minimum tone height of 55 Hz.

Figure 5.3 illustrates the iterative process. The blue and the red scatter points stand for the distinct pitches constituted by the added partials. The size of the scatter points is determined by the estimated weighting. The cyan dot represents the weighted pitch average. As is apparent from the figure, partials that are located far away from the average pitch are excluded from the next calculation of the weighted mean. Although, such partials might be included again, if the resulting average converges towards the pitch determined by the partial.

The iterative process is terminated when no new harmonic can be excluded or included, or when the maximum number of three iterations was reached. So finally, the tone height measured in cent c_{tone} is represented by the weighted average of all harmonics included in the last iteration. While convergence to the correct result cannot be guaranteed, the procedure improves the overall estimation accuracy significantly. The parameters and thresholds given in the formulas above have been found by experiment and the maximization of the melody detection accuracy.

5.7. Perceived Pitch

While the frame-wise updated fundamental frequency is a continuous entity, it is necessary to define the perceived pitch of a tone, which can be categorized into notes of the underlying musical scale. Of course the perceived pitch also changes with time, so a final estimate can only be given, when the tone is finished. Nonetheless, it is an advantage to update the perceived pitch continuously, as the stable pitch estimates can help to find tone offsets based on the evolution of the frequency.

Two different ways to compute the stable pitch are provided: one for stable tones and one for tones with a vibrato. Tones with a rather random frequency evolution are handled after the tone is finished.

5.7.1. Stable Tones

Stable tones have – at least piecewise (more than 25 ms) – a stable frequency. A tone with a stable frequency has to meet the following requirements: first, the absolute value of the long-term frequency difference is smaller than 2 (see Section 5.11.3 for a description of the long-term frequency difference). Second, the absolute cent difference is measured 25 ms and 50 ms apart. None of the estimated differences should exceed 20 cent, otherwise the stable frequency counter is set to zero. If the requirements are met, the counter is incremented in order to count the analysis frames with a stable frequency until it reaches an equivalent of 25 ms. In this case, the tone is declared stable and the current stable tone height $c_{\text{tone\_stable}}$ as well as short-term tone height $\bar{c}_{25\text{ms}}$ are set to the average of the last three pitch values. During subsequent stable frames $c_{\text{tone\_stable}}$ is updated with the value of $\bar{c}_{25\text{ms}}$, which is now computed as EMA of the instantaneously estimated tone height.

If the tone ceases to fulfil the requirements for a stable tone, the last stable tone height is kept until a new stable pitch can be estimated. At this point the two tone heights can be compared to detect a frequency based tone offset.

The final perceived pitch, which is used for the MIDI note estimation, is the average of all fundamental frequency values between the first stable frequency frame and the last stable frequency frame.

5.7.2. Tones with Vibrato

The vibrato is a musical ornament which occurs in sustained tones of various instruments and the human singing voice. It is typically characterized by its frequency modulation rate and its extent (amount of frequency variation). Both features may vary significantly with the instrumentation and musical style, typical values in music are a rate of 6 Hz and an extent of less than ±100 cent.

As the vibrato typically exceeds the note boundaries of the equal-temperament scale, the calculation of the tone height is not straight forward. Shonle and Horan (1980) found that the pitch of vibrato tones is close to the geometric mean of the two extreme frequencies, yet experiments with asymmetrical vibrato waveforms suggest an averaging of all frequencies present. Brown and Vaughn (1996) found that for vibrato tones the pitch is perceived as mean of the frequency values, pointing out that there is no significant difference between arithmetic and geometric mean (0.1 cent). While for long vibrato tones, the averaging of all frequencies seems to be sufficient, the calculation is not straight forward for short tones. d'Alessandro and Castellengo (1994) showed that an exponentially weighted average of the F0 pattern models well empirical data of the perceived tone height. In this case more weight is given to the last part of the tone.

If the perceived tone height should be used to detect frequency-based tone offsets, the previously cited approaches do not suffice, because onset and offset and all frame-wise F0 of the tone have to be known prior to the estimation of the tone's pitch. If the average is updated right from the beginning of the tone, the first few estimated pitch values fluctuate vividly and may cause the erroneous detection of frequency-based tone offsets. Prame (1997) estimates the perceived tone height as a running vibrato-semi-cycle mean of the F0 (the average of all cent values between two extrema) to tackle this problem. Of course, in this case the extrema have to be detected on the fly.

In our algorithm, two approaches are used for pitch estimation of vibrato notes: first, the average of the identified minimum and maximum cent value is used to update the perceived tone height on the fly (Seashore, 1938, Chapter4). This value is used to detect frequency-based tone offsets. Second, the mean of all F0 in the vibrato and stable parts of a tone is used to estimate the final MIDI note number. As we calculate the arithmetic mean of the frame-wise F0 given in cent, the result is concordant with the geometric mean of the frame-wise F0 given in Hertz.

5.7.3. Unstable Tones

If the tone is finished and no stable frequency could be estimated, a perceived pitch has to be forced from the frame-wise evolution of the fundamental frequency. As the beginning and the end of a note often have a very unstable frequency, the first 70 ms and the last 50 ms are not evaluated. If the tone is too short for this procedure one third of the tone is cutoff from the beginning and one fourth from the end. Then, the average cent value from the remaining fundamental frequency values is calculated and taken as the perceived pitch.

5.8. Onset and Offset Detection

Today, several types of onset detection algorithms exist which are designed to work best in the scope of their application. Specialized onset detection algorithms have no other purpose than to detect onsets, but there are also algorithms which simultaneously transcribe multiple notes or the melody.

Most specialized onset detection algorithms use a frequency representation (spectrogram) of the signal. Thereby, experiments suggest that a similarly high level of performance can be obtained with a magnitude-based (e.g. spectral flux), a phase-based (e.g. phase deviation) or a complex domain onset detection function. Dixon (2006) suggests a method, that compares complex Fourier coefficients in adjacent frames, yet, the vast majority of onset detection algorithms employ energy-based features. This may be attributed to the fact that it is an intuitive, simple and fast solution providing good results (Bello et al., 2005). However, energy-based features cannot cover all types of tone onsets. A good example is the human singing voice which features the so-called soft onset. Soft onsets don't show a pronounced rise in the overall signal energy, but are characterized by a change in pitch or timbre.

The state-of-the-art algorithm[1] employs three parallel STFTs with different data window lengths (11.61, 23.22, and 46.44 ms) (Böck et al., 2012b). The linear magnitude spectrograms are then "filtered" (using 24 triangular filters with center frequencies aligned to the Bark scale) to obtain a compressed representation. The magnitudes as well as the magnitude differences relative to preceding frames comprise the input to a neural network, which then identifies the onsets.

Most of the onset detection algorithms focus on the analysis of monophonic recordings (Toh et al., 2008; Thoshkahna and Ramakrishnan, 2009; Heo et al., 2013). In this case the whole signal can be utilized to extract features for onset detection. For example, Heo et al. (2013) use cepstral analysis and evaluate the harmonic cepstrum regularity to detect onsets. Toh et al. (2008) evaluate different spectral and cepstral feature types on a pair of trained GMMs (onset and offset), in which the onset detection accuracy of each feature type defines the final weighting for the detection function that is based on all features (feature fusion).

Several approaches mimic the human auditory system (Klapuri, 1999; Heinz, 2006;

[1] The offline version of this system performed best in the MIREX onset detection task in 2013.

Thoshkahna and Ramakrishnan, 2009). Klapuri (1999) implements a band-wise processing (21 critical-band filters) and a psychoacoustic model of intensity, and then combines the results from separate frequency bands to detect onsets in a variety of musical signals. Heinz (2006) uses an ear model as signal processing front-end, then he segments the found pitch trajectories by monitoring the amplitude of the individual partials. Thoshkahna and Ramakrishnan (2009) propose a psychoacoustically motivated onset detection for hummed queries in a query by humming (QBH) application to detect onsets on monophonic syllables /na/ /la/ /ta/ /da/ and /hm/.

In published melody extraction systems, only a few algorithms tackle the problem of note segmentation (Paiva et al., 2008; Ryynänen and Klapuri, 2008; Gómez et al., 2012). Yet, in case they do, the Fourier spectrum is not necessarily the primary feature to find tone onsets or offsets. Instead, the evolution of each pitch track (or tone) can be monitored individually to detect significant changes in its magnitude, pitch or the spectral envelope. The pitch-based offset is a speciality of note transcription algorithms, because it requires information about the tone's fundamental frequency. Using pitch as a feature to detect note onsets can be seen as the biggest advantage of melody extraction systems – even though the unison is the most frequent interval in the melody, it comprises only about 25 percent of all possible intervals. This means, of course, that 75 percent of the tone onsets can be identified with the help of pitch alone.

Many algorithms designed for the analysis of instrumental music implement a simple onset detection mechanism based on pitch: it splits a tone whenever its F0 rounds to another semitone. This approach allows a pitch variation of ±50 cent, provided that the tone is perfectly tuned according to a specified reference frequency and an equal-temperament scale is used. While many wind and stringed instruments have fixed pitches (or their pitch variation stays within the given margin), this approach is not applicable for the human singing voice, because the frequency variation of individual tones easily exceeds semitone boundaries.

McNab et al. (1996) have implemented a pitch-based note segmentation for a QbH system: pitch tracks are divided into 20 ms segments and the average F0 is estimated in each segment. The track is split, if adjacent segments have a pitch difference of more than 50 cent. The method is not dependent on a predefined reference frequency and it adjusts well to slowly varying pitches, but our own experiments showed that too many onsets are detected in tones with a strong vibrato.

Paiva et al. (2008) combines pitch-based segmentation as well as salience-based segmentation to split pitch tracks into several notes. At first the continuous pitch track is approximated by a set of piece-wise constant functions which are obtained by the quantization of each frequency value to the corresponding MIDI note number. (The reference frequency is a prior and corresponds to 8.1758 Hz). Four stages of filtering are applied with the purpose of coping with common performance styles (vibrato and glissando), as well as jitter, pitch detection errors, intonation problems and so forth.

Gómez et al. (2012) compared two strategies for predominant fundamental frequency transcription in flamenco singing with guitar accompaniment: in the first approach, the melody voice is separated from the accompaniment and then a monophonic transcription system is employed. In the second approach, the melody pitch contour is identified in the signal, and its salience, pitch and even the corresponding harmonics are utilized to detect onsets. They found that the approach working directly with the identified melody pitch contour outperforms the source separation approach significantly. Gómez et al. (2012) also tackle the problem of over-segmentation in case of glides and vibrato. Thereby, several features are exploited, as for example tone duration, pitch stability and the existence of voiced and unvoiced frames.

In our algorithm, we use energy-based as well as pitch-based features to to detect tone onsets and offsets. However, the features are not computed from the overall spectrogram, but estimated separately for each tone.

5.8.1. Magnitude-based Onset

The energy-based onset is normally detected as a sudden rise in the tone's magnitude, which segments a continuous pitch track into several notes. Since a sharp increase in the harmonic magnitudes is prohibited by their constricted update procedure in our algorithm, a sudden rise in the tone's magnitude is impossible in our algorithm. On account of this, a new tone at the same frequency as an existing tone is detected as a salient pitch in the pitch spectrogram. If the pitch is strong enough, a new tone is started on top of an existing one.

5.8.2. Magnitude-based Offset

An onset cannot be detected with the help of the tone magnitude, but the magnitude offset is another business, as the harmonic magnitudes are allowed to decay promptly. A magnitude drop is detected by the comparison of the current tone magnitude A_{tone} and the corrected long-term magnitude $\bar{A}^{*}_{\text{tone, 100ms}}$ (see Section 5.5.1):

$$A_{\text{tone}} < \bar{A}^{*}_{\text{tone, 100ms}} \cdot \bar{r}_{\text{tone}}. \tag{5.29}$$

The variable $\bar{r}_{\text{tone}}$ defines the magnitude ratio that triggers the offset counter. So the offset threshold is basically a relative difference function, which describes the fact that a perceived change in intensity is approximately proportional to the intensity of the signal (Weber's law) (Moore, 2003, ch. 4). However, $\bar{r}_{\text{tone}}$ is not constant, but depends also on the magnitude variability of the tone: the more stable the magnitude of a tone has been in previous frames, the more noticeable are changes. The value of $\bar{r}_{\text{tone}}$ is estimated as follows:

$$\bar{r}_{\text{tone}} \leftarrow \alpha_{50\text{ms}} \cdot \bar{r}_{\text{tone}} + (1 - \alpha_{50\text{ms}}) \cdot r_{\text{tone}}, \tag{5.30}$$

with

$$r_{\text{tone}} = \begin{cases} 0.6 \cdot \dfrac{A_{\text{tone}}}{\bar{A}^*_{\text{tone, 100ms}}}, & \text{if} \quad A_{\text{tone}} < \bar{A}^*_{\text{tone, 100ms}}; \\ 0.6 \cdot \dfrac{\bar{A}^*_{\text{tone, 100ms}}}{A_{\text{tone}}}, & \text{else.} \end{cases}$$

$\bar{r}_{\text{tone}}$ is initialized with the value 0.6. For tones with a very stable magnitude the value of $\bar{r}_{\text{tone}}$ is close to 0.6. Hence an offset is triggered for a magnitude drop of about 4.5 dB. The values for r have been found by experiment.

A magnitude drop has to persist for the period of 25 ms to be valid. If the offset counter reaches the number of frames equivalent to 25 ms, the tone is marked for a magnitude offset. However, the tone is not immediately terminated to avoid that a new tone is started on the reminder of the old one. That is the reason why additional conditions are defined that ensure that the tone is finished safely. A tone is finished if one of the following conditions is met:

- its magnitude is well below the the magnitude threshold for tone onsets.
- it has an unpredictable F0, eg. the tone height prediction error e_t is greater than 35, or the long-term tone height prediction error $\bar{e}_{25\text{ms}}$ is greater than 15 (see Section 5.11.4).
- the offset counter exceeds the equivalent of 100 ms.

5.8.3. Pitch-based Offset

In the proposed pitch-based offset detection algorithm, a necessary pre-condition for the detection of a frequency offset is the estimation of a stable perceived tone height, so that a categorical attribution is possible. It is shown in Section 5.7 that a stable tone height cannot be determined instantly, especially if the pitch varies. From this it follows that a splitting point (a tone offset) can only be defined retrospectively. Furthermore, it is obvious that at least two differing tone heights have to be detected to define a pitch-based offset. For the offset detection, the current stable tone height is compared with the last estimated stable tone height. In general, the tone is split, if the absolute difference exceeds 80 cent. Admittedly, the rule is more complicated for tones with vibrato: The tone height of vibrato tones is updated with every detected maximum or minimum needing at least two extrema to compute a stable tone height, which is computed as the average of the minimum and maximum value. A split is possible after three successive extrema have been detected. However, experiments have shown that if the tone height of vibrato tones shifts to a new note, it often needs two updates (a maximum and a minimum or vice versa) to reach the new tone height. That is why the stable tone height is put on a “hold” if the frequency difference between the last estimated tone height and the current tone height exceeds 40 cent, and the final decision is postponed to the next combination of a maximum/minimum value.

Another variant for a frequency based-offset detection is a proximity threshold between consecutive frames. In the proposed algorithm, the maximum frequency difference in consecutive frames is set by the tone’s search range for new harmonics. It depends on the tone’s frequency variability and the prediction error in previous frames (see Section 5.9). The search range covers at least ±65 cent around the last pitch value, a maximum search range is not defined.

5.8.4. Deletion of Excess Tones

Sometimes, it can be difficult to detect a perceptual tone offset at all. Many instruments produce sounds which do not have a marked offset, because their amplitude decays steadily (piano, guitar, xylophone, etc.). And even when the tone has a marked release, the offset might be masked by other sounds. Still, it is important to eventually delete the so-called “lost tones”. The measures taken to detect and delete excess tones will be introduced in the following sections. The parameters used in the subsequently presented formulas have been adjusted by maximizing the melody extraction accuracy.

Masked Tones

Listening experiments with humans suggest that in the case of simultaneous masking the tone’s parameters (F0, magnitude, timbre) are put on a hold until they can be updated according to new data. When the algorithm declares a tone as masked, it tries to emulate the so-called spectral restoration: it freezes the tone for up to 150 ms, provided that the magnitude spectrum supports the masking hypothesis. Only after this time span (which is still lower than the 300 ms spectral restoration time that has been reported for humans by Warren (1999)) the tone is discarded and will be removed from the tone list. A tone is declared masked, if the magnitude sum of all masked harmonics is greater than forty percent of the tone’s magnitude. (For the definition of a masked harmonic, refer to Section 5.4.7.)

Exclusive Magnitude

The exclusive magnitude $A_{\text{tone\_exclusive}}$ differs from the tone magnitude A_{tone} in so far, as it features the portion of the tone's magnitude which cannot be covered by other tones. So if two tones share a harmonic, this harmonic might not contribute to the exclusive magnitude of the tone at all (see sections 5.4.6 and 5.5.3 for more information). The evaluation of the exclusive magnitude allows the detection of octave errors and their subsequent elimination: if a huge proportion of the tone's magnitude can be explained by other existing tones, it will be deleted. The required exclusive magnitude to pass the verification depends on the frequency variability of the tone. While tones with a variable fundamental frequency enjoy many advantages during the tone processing, in terms of the exclusive magnitude the requirement is higher, because it is very improbable that a tone with a varying frequency will share harmonics with other tones over a longer time. If it does, it means that the fundamental frequencies of at least two tones are changing in parallel, which is so improbable that one of the tones can be declared dispensable.

A tone is declared *dispensable* if the exclusive magnitude of the tone is not significant compared to the magnitude of other tones:

$$A_{\text{tone\_exclusive}} < (0.02f + 0.2 \cdot \bar{r}_{\text{freq\_deviation}})\, A_{\text{tone, max}}, \tag{5.31}$$

where $A_{\text{tone, max}}$ denotes the maximum tone magnitude in the current frame and $\bar{r}_{\text{freq\_deviation}}$ denotes the current frequency rating of the tone (see Section 5.11.6).

If a tone is declared dispensable for a longer time span (75 – 150 ms), it is deleted. The time span depends on the frequency variability of the tone.

Tone Height Prediction Error

A third mechanism to detect excess tones is the monitoring of the tone's pitch: if the F0 estimates remain unpredictable over a certain time, the tone is deleted. Hence, a counter is established that monitors the tone height prediction error e_t, which is explained in Section 5.11.4:

$$\text{counter}_t = \begin{cases} \text{counter}_{t-1} + 35, & \text{if} \quad e_t > 50 \\ \text{counter}_{t-1} + e_t - 15, & \text{else.} \end{cases} \tag{5.32}$$

In case the counter reaches a negative value, it is set to zero. If the counter is higher than 200, the tone will be terminated.

Collision of Tones

The melody extraction algorithm cannot keep apart two tones at the same F0 for a long time. Whenever two tones have the same F0 (less than 25 cent difference), the weaker tone will be removed if this condition continues for more than 30 ms. However, this condition is only evaluated as long as a tone is not declared masked.

Number of Tones

If the number of tones rises above 10, the tone with the lowest exclusive magnitude $A_{\text{tone\_exclusive}}$ is deleted. The reason for this limitation lies in the implementation of the algorithm – the allowed number of tones could be easily increased, yet under normal circumstances, this precautionary measure rarely applies.

5.9. Pitch Range Prediction and Tone Map

Only if the tone's F0 can be predicted with sufficient accuracy, it is possible to assign new harmonic candidates in the next analysis frame. A probable pitch range has to be estimated that should be big enough to cover up for the frequency variability of the tone, but it should also be small enough to avoid the assignment of harmonics from other sound sources. The implemented pitch range prediction is rather simple. A noticeable feature is that the predicted range is always centered around the last F0 of the tone – only the allowed frequency offset changes. This way, there is an inherent tendency for the tone to remain on it's current frequency. The allowed cent offset Δc amounts to

$$\Delta c = 65 + \bar{r}_{\text{abs\_freq\_dif}} + |\bar{r}_{\text{freq\_dif}}| + 0.5 \cdot e_t, \tag{5.33}$$

where $\bar{r}_{\text{abs\_freq\_dif}}$ is the long-term absolute F0 difference, $\bar{r}_{\text{freq\_dif}}$ is the long-term F0 difference and e_t is the prediction error. The parameters used in this formula have been found empirically. The variables are described in sections 5.11.2, 5.11.3 and 5.11.4, respectively. As the first two variables measure the frequency deviation of the tone, it becomes obvious that tones with a varying frequency have a bigger search range.

Now that the pitch range for the next analysis frame is predicted, the tone can be entered into the tone map. The tone map allows to refer to a tone object by its frequency. It is implemented as a simple array that covers the whole frequency range (55 Hz – 1319 Hz) with a 100 cent resolution. A pointer to the tone object is saved in all array bins that are in the predicted pitch range of the tone. As two tones can be in the same frequency range, two tone maps exist. If three (or more tones) are in the same frequency range, the weakest tone is declared masked and put on hold.

5.10. Midi Note Labeling

The output of the melody as MIDI notes is a very abstract and hence compact form of representation. Originally, the MIDI standard specified a communications protocol for synthesizers, but more standards were created afterwards to describe the character and evolution of particular sounds or sound effects. For our purpose, a very limited view on a "MIDI note" suffices: each note is defined by its onset, its discrete tone height (i.e. a MIDI note number) and the duration.

Only a few melody extraction algorithms tackle the problem of onset detection and the estimation of the perceived discrete tone height. Paiva et al. (2008) aim to explicitly distinguish individual musical notes, characterized by specific temporal boundaries and MIDI note numbers. Ryynänen and Klapuri (2006) have implemented a method for the automatic detection of singing melodies in polyphonic music that transcribes MIDI notes and detects the musical key of the recording. They derive a hidden Markov model for note events from fundamental frequencies, their saliences and an accent signal. Singing rest segments are described by a Gaussian mixture model. Laaksonen (2014) presented an algorithm for symbolic melody transcriptionf that divides the audio into segments based on an already available chord transcription, and then matches potential melody patterns to each segment. Gómez et al. (2012) implemented a note transcription system that at the same time computes the tuning frequency with the help of a histogram.

Any system that at some point quantizes frequencies from the continuous scale into pitch classes or semitone intervals needs a specified reference frequency. Since in most practical cases it is not possible to make a safe assumption about the underlying reference frequency of a music recording (Lerch, 2006), the only possibility is to estimate this value from the data itself. Several methods reported in the literature make use of histograms for the estimation (Ryynänen, 2004; Harte and Sandler, 2005; Gómez, 2006). In order to compute the tuning frequency, the histogram covers the frequency range of ±50 cent – this is the maximum possible deviation from the standard tuning frequency. Depending on the chosen resolution, a histogram contains several histogram bins which count the occurrence of spectral peaks (or pitches) in their appointed frequency range. The bin with the highest count marks the frequency offset from the standard tuning.

The histogram approach has the disadvantage that the pitch values already need to be quantized in order to construct the histogram. Dixon (1996) counters the problem of the preliminary quantization by the use of an iterative optimization mechanism that is repeated until the reference frequency converges on a reasonably stable value.

Gómez et al. (2012) enters frame-wise pitch values into a histogram with a high resolution (1 cent) using a bell-shaped window that spans several histogram bins. Moreover, the F0 values are weighted according to their derivative, giving more weight to stable pitches. After the note segmentation (including the computation of the nominal pitch of each note) the estimated tuning frequency is refined using the nominal pitch of voiced notes. The updated tuning frequency asks for the consolidation of the nominal pitches, and every correction of a note's nominal pitch is followed by the re-estimation of the tuning frequency. The process is repeated until no further correction is necessary.

McNab et al. (1996) devise an adaptive tuning frequency estimation that is capable to follow the own-tuning of individual singers. There, a constantly changing offset is employed, which is initially estimated as if the user sings to the standard reference frequency A-440, but then adjusts by referencing each note to its predecessor. This method is based on the assumption that singing errors tend to accumulate over time. Another adaptive approach, that is used to display the tuning frequency evolution for polyphonic choir music over time, is presented by Gnann et al. (2011): first, the spectral peak frequencies are allocated to the equal temperament scale using the median of seven previous reference frequency values. Then, this estimation is updated appointing the reference frequency that leads to the smallest least-squares-error when the measured peak frequencies are sorted into a semitone grid.

5.10.1. Reference Frequency Estimation

We have proposed an approach originating from the domain of circular statistics that does not require a quantization, is efficiently computable and consumes only minimal storage (Dressler and Streich, 2007). In this approach, the frequencies of tones are seen as circular quantities that "wrap around" after reaching a ±50 cent offset. Degani et al. (2014) compared three tuning frequency estimation methods (histogram, the least-square optimization introduced by Gnann et al. (2011) and our method based on circular statistics) and found that while all three reach a similar high accuracy for the estimated tuning frequency, the circular method ist the fastest to converge to the reference frequency and is the computationally most efficient.

In (Dressler and Streich, 2007), two different methods for reference frequency estimation are introduced: one adaptive reference frequency estimation, that is updated in every analysis frame, and one global reference frequency estimation. Since we assume that the reference frequency does not change within a music file, we use the global method, which gives more reliable results. However, this means that the reference frequency estimation is an offline method, since all tone heights have to be known before the final result can be computed.

For the estimation of the tuning frequency only the cent deviation from the ideal categorical frequency is of importance, no matter to which semitone in particular a tone is assigned. If a quantity "wraps around" after reaching a certain value we may speak of circular data. The simple calculation of the arithmetic mean is not an appropriate statistic for such data. In this case histogram techniques can be used to exploit a certain trend. Another way to deal with circular or directional quantities is circular statistics, a subdiscipline of statistics that is for example applied when directions or periodic time measurements (e.g. day, week, month) have to be evaluated (Batschelet, 1981). Such data are often best handled not as scalars, but as (unit) vectors in the complex plane. Each cent value c is treated as a unit vector $\hat{u}$ whose angle ϕ is the appropriate fraction of a full circle:

$$\hat{u} = 1 \cdot e^{j\phi} \tag{5.34}$$

with

$$\phi = \frac{2\pi}{100} \cdot c.$$

In order to determine the tuning frequency of the entire audio piece the complex sum of all tone heights is computed and then divided by the number of values N to get a mean $\bar{z}$ of the circular quantities:

$$\begin{aligned} \mathrm{Re}(\bar{z}) &= \frac{\sum_i \cos(\phi_i)}{N}, \\ \mathrm{Im}(\bar{z}) &= \frac{\sum_i \sin(\phi_i)}{N}. \end{aligned} \tag{5.35}$$

The mean $\bar{z}$ is again a vector in the complex plane which means it can be decomposed into a magnitude and a phase value. The magnitude $|\bar{z}|$ lies in the interval $[0, 1]$. It depends on the amount of variation in the vectors that are averaged. The more the cent values are scattered, the smaller the magnitude of the complex number $\bar{z}$ will be, whereas if $\bar{z}$ has a magnitude close to 1, the value implies a significant tendency in the data. We can therefore see $|\bar{z}|$ as a confidence measure for the tuning frequency estimate.

The phase angle $\bar{\phi}$ of $\bar{z}$ is converted back into cent to obtain the deviation from the standard tuning frequency of 440 Hz in cent:

$$\Delta c = \frac{100}{2\pi} \arg(\bar{z}). \tag{5.36}$$

With help of the estimated cent deviation Δc the reference frequency can be calculated as

$$f_{ref} = 2^{\Delta c/1200} \cdot 440\,\mathrm{Hz}. \tag{5.37}$$

The results of the calculation can be improved if each cent vector is weighted by a certain weighting factor r_i that could be the magnitude of a tone or the magnitude of a spectral peak:

$$\begin{aligned} \mathrm{Re}(\bar{z}) &= \frac{\sum_i r_i \cos(\phi_i)}{\sum_i r_i}, \\ \mathrm{Im}(\bar{z}) &= \frac{\sum_i r_i \sin(\phi_i)}{\sum_i r_i}. \end{aligned} \tag{5.38}$$

To compute the confidence we have to divide the computed complex number by the sum of weights (instead of the number of values).

5.10.2. MIDI Note Estimation

Prior to the computation of the tone's MIDI note number, a stable tone height has to be assigned to each tone. The task is not straight forward as there are many instruments that allow a substantial frequency variation within one tone. Nevertheless the perceived tone height of such tones can be matched with a stable pitch.

In Section 5.7, we distinguished three different cases: stable tones, tones with vibrato, and tones in which no stable frequency could be assigned. For all tones, which have a valid stable tone height (i.e. stable tones and tones with vibrato), the tone height used for the MIDI note estimation is the average of all cent values between the first and the last frame with a stable pitch. If no stable tone height was assigned, the average tone height is computed from the middle part of the tone. This means that beginning and end of the tone are excluded from the computation to allow some time to reach the proper pitch, as well as to cut off any unimportant frequency variations at the end of a tone. In this case the first third of the tone is cut off at the beginning and the last quarter of the tone at the end. (A more detailed description of the calculation of the perceived tone height is given in Section 5.7.)

After a perceived tone height has been assigned, the internal algorithm reference frequency (55 Hz) has to be changed to the MIDI reference frequency (8.1758 Hz). This means the tone height c_{tone} measured in cent has to be increased by 3300 cent, because

$$1200 \cdot \log_2 \left(\frac{55\,\text{Hz}}{8.1758\,\text{Hz}} \right) = 3300. \tag{5.39}$$

Then, the frequency offset to the standard tuning (A-440 Hz) Δc , which was estimated in the previous tuning frequency estimation, has to be applied. To acquire the MIDI note number, the resulting cent value is divided by 100 and rounded to the nearest MIDI note number:

$$\text{MIDI note number} = \text{round}((c_{\text{tone}} + \Delta c)/100) \tag{5.40}$$

5.11. Various Tone Features

This section contains various tone features that could not be described elsewhere. It is not necessary to read this section as a whole, rather the distinct subsections can be read when they are referred to in the text. (All half-life times given in the subsequent equations have been found empirically by maximizing the melody extraction accuracy of the system.)

5.11.1. Average Pitch

The average pitch $\bar{c}_{25\text{ms}}$ is the exponential moving average (EMA) of the fundamental frequency of the tone:

$$\bar{c}_{25\text{ms}} \leftarrow \alpha_{25\text{ms}} \cdot \bar{c}_{25\text{ms}} + (1 - \alpha_{25\text{ms}}) \cdot c_{\text{tone}}, \tag{5.41}$$

in which the parameter $\alpha_{25\text{ms}}$ is the smoothing factor that refers to a 25 ms half-life time and c_{tone} is the actual fundamental frequency of the tone measured in cent. $\bar{c}_{25\text{ms}}$ is initialized with the average fundamental frequency of the pitch track.

5.11.2. Long-Term Absolute Frequency Difference

The long-term absolute frequency difference measures the absolute frequency deviation of the tone. It is implemented as exponential moving average and estimated as follows:

$$\bar{r}_{\text{abs\_freq\_dif}} \leftarrow \alpha_{30\text{ms}} \cdot \bar{r}_{\text{abs\_freq\_dif}} + (1 - \alpha_{30\text{ms}}) \cdot |c_t - c_{t-1}|, \tag{5.42}$$

in which the parameter $\alpha_{30\text{ms}}$ is the smoothing factor that refers to a 30 ms half-life time. $|c_t - c_{t-1}|$ is the tone's absolute frequency difference between the last and the current analysis frame. $\bar{r}_{\text{abs\_freq\_dif}}$ is initialized with the average absolute cent difference of the pitch track.

5.11.3. Long-Term Frequency Difference

The long-term frequency difference measures the frequency deviation of the tone. It is implemented as exponential moving average and estimated as follows:

$$\bar{r}_{\text{freq\_dif}} \leftarrow \alpha_{20\text{ms}} \cdot \bar{r}_{\text{freq\_dif}} + (1 - \alpha_{20\text{ms}}) \cdot (c_t - c_{t-1}), \tag{5.43}$$

in which the parameter $\alpha_{20\text{ms}}$ is the smoothing factor that refers to a 20 ms half-life time. $c_t - c_{t-1}$ is the tone's frequency difference between the last and the current analysis frame. $\bar{r}_{\text{freq\_dif}}$ is initialized with zero.

5.11.4. Tone Height Prediction Error

The difference between the predicted tone height and its actual value is used as a measure of tone quality.

The calculation of the tone height prediction error is based on linear prediction, which is a simple form of first-order extrapolation: if the frequency has been changing at a certain degree, it will probably continue to change at approximately the same degree. So from the last two frequency values measured in cent, the next value is assumed at $c_{\text{pred}} = 2c_{t-1} - c_{t-2}$.

Then, the prediction error would be the absolute difference between predicted F0 c_{pred} and estimated F0 c_t (all frequencies in cent):

$$e_{\text{lp}} = |c_{\text{pred}} - c_t| = |2c_{t-1} - c_{t-2} - c_t| \tag{5.44}$$

The main obstacle in linear prediction is its inability to cover frequency fluctuations on a very small scale. This behavior might lead to a very high prediction error for some tones, even if their pitch stays in the same frequency range. To reduce the effect of small-scale fluctuations on the prediction error (especially its long-term value), a second option allows to diminish the prediction error if the current pitch is close to the average pitch $\bar{c}_{25\text{ms}}$, which is described in Section 5.11.1. The prediction error based on the offset from the average pitch is calculated as:

$$e_{\text{avg}} = 1.4 \cdot |\bar{c}_{25\text{ms}} - c_t|. \tag{5.45}$$

The resulting prediction error e_t is the minimum of e_{lp} and e_{avg}. The long-term prediction error $\bar{e}_{25\text{ms}}$ is the EMA of e_t with a half-life period of 25 ms:

$$\bar{e}_{25\text{ms}} \leftarrow \alpha_{25\text{ms}} \cdot \bar{e}_{25\text{ms}} + (1 - \alpha_{25\text{ms}}) \cdot e_t. \tag{5.46}$$

5.11.5. Average Harmonic Frequency Offset

$\bar{r}_{\text{harmonic\_offset}}$ is the long-term average cent offset from the ideal harmonic position for the harmonics of a tone. The long-term average is computed as exponential moving average with a smoothing factor $\alpha_{50\text{ms}}$ that corresponds to a half-life time of 50 ms:

$$\bar{r}_{\text{harmonic\_offset}} \leftarrow \alpha_{50\text{ms}} \cdot \bar{r}_{\text{harmonic\_offset}} + (1 - \alpha_{50\text{ms}}) \cdot r_{\text{harmonic\_offset}}, \tag{5.47}$$

in which $r_{\text{harmonic\_offset}}$ is the frame-wise average offset from the ideal harmonic position, calculated as

$$r_{\text{harmonic\_offset}} = \frac{1}{N} \sum_{h=1}^{20} |c_h - c_{\text{tone}} - 1200 \cdot \log_2(h)|.$$

In the equation, c_h is the harmonic frequency (i.e. the instantaneous frequency of the assigned spectral peak) measured in cent, c_{tone} is the fundamental frequency of the tone measured in cent and h is the harmonic number. $\bar{r}_{\text{harmonic\_offset}}$ is initialized with zero. Please note that only harmonics with assigned spectral peaks are accumulated for the frame-wise average. That is why N does not equal 20, which is the maximum harmonic number, but it equals the actual number of harmonics with assigned peaks.

5.11.6. Frequency Deviation Rating

The frequency deviation rating $\bar{r}_{\text{freq\_deviation}}$ sets the long-term frequency difference $\bar{r}_{\text{freq\_dif}}$ (see Section 5.11.3) in relation with the long-term prediction error $\bar{e}_{25\text{ms}}$ (see Section 5.11.4)

$$r_{\text{freq\_deviation}} = \begin{cases} |\bar{r}_{\text{freq\_dif}}|/e_t & \text{if } |\bar{r}_{\text{freq\_dif}}| < \bar{e}_{25\text{ms}} \\ 1, & \text{else.} \end{cases} \tag{5.48}$$

The exponential moving average of the frequency deviation rating is computed as:

$$\bar{r}_{\text{freq\_deviation}} \leftarrow \alpha_{100\text{ms}} \cdot \bar{r}_{\text{freq\_deviation}} + (1 - \alpha_{100\text{ms}}) \cdot \bar{r}_{\text{freq\_deviation}}, \tag{5.49}$$

in which the parameter $\alpha_{100\text{ms}}$ is the smoothing factor that refers to a 100 ms half-life time. $\bar{r}_{\text{freq\_deviation}}$ is initialized with zero. The frequency deviation rating is used to evaluate the frequency modulation of a tone. Values that are greater than 0.4 mark a tone as frequency modulated. In this case it is very probable that the tone belongs to the melody.

Chapter 6.

Musical Voices and Melody

6.1. Introduction

The tracking of tones is followed by the formation of musical voices. This processing step is closely related to auditory scene analysis (ASA), a term coined by psychologist Albert Bregman (1994). ASA provides a model for the organization of sound into perceptually meaningful elements, using grouping principles related to the principles of Gestalt psychology. The goal of ASA is to find out which sounds can be treated as parts of the same sound source, hereby using several cues to integrate sound into one auditory stream (simultaneous and sequential grouping) or to segregate sounds into different auditory streams. The cues used by the human auditory system to group sounds comprise frequency proximity, timbre, loudness, spatial location, temporal relations, playback speed and repetitions.

For example, an important characteristic of the human auditory system is the influence of note onset rate on the stream segregation. Tone sequences that are a quick succession of large intervals actually fail to form a recognizable melody, since the auditory system cannot integrate the individual tones into one auditory stream (Bregman, 1994, Chapter 2). The integration or segregation of such a tone sequence depends markedly on the duration of the tones – an aspect that is covered by the proposed algorithm.

Stream recognition is not restricted to music: Bird and Darwin (1998) used simultaneously presented sentences to show that the identification performance increased due to frequency difference. While the effect is most noticeable within one semitone difference, performance rises even beyond one semitone (the experiment shows a continuing increase in recognition till 6 to 8 semitones). This result can be explained by the successful segregation of the sentences into different auditory streams. Assmann and Summerfield (1990) found in an experiment, in which concurrent double vowels were used for identification purposes, that no increase in recognition was shown after 1 semitone (with steepest increase around 50 cent difference). But he found also, that listeners could reliably adjust the fundamental frequency (F0) of a harmonic complex to match the individual pitches of concurrent vowels that have F0s separated by 4 semitones or more, whereas at smaller separations they heard and matched only a single pitch (Assmann and Paschall, 1998). This means that there is a difference between the identification of two sounds and other tasks (e.g. sequential organization of sounds), strongly suggesting that the task of pitch matching requires the attribution of simultaneous sounds to distinct auditory streams.

In melody extraction algorithms, usually a simple model of stream segregation is implemented using mostly the cues loudness and frequency proximity to link melody notes together. Hereby, two different strategies can be distinguished. The first one is a rule-based approach, which decides the succession of melody tones using heuristics (Wendelboe, 2009; Joo et al., 2009; Chien et al., 2011; Salamon and Gómez, 2012), the second one is the statistical approach, which uses a probabilistic model to extract the melody path with the greatest likelihood (Hsu et al., 2009; Rao and Rao, 2009; Durrieu et al., 2010; Song et al., 2014). Both approaches gain comparable results, yet rule-based approaches usually are computationally more efficient.

In the proposed melody extraction system, an auditory streaming model is implemented that takes the frame-wise frequency and magnitude of tones as input. With this information, so-called voice objects are established, which in turn capture salient tones close to their preferred frequency range. Although no statistical model is implemented, probabilistic relationships that can be observed in melody tone sequences are exploited. While the tone's loudness is the most important selection criterion, small intervals between successive notes are preferred. Another problem to be addressed is the identification of non-voiced portions, i.e. frames where no melody tones occur. In the subsequent sections the approach towards auditory scene analysis is described as given in (Dressler, 2012).

6.1.1. Statistical Properties of Melodies

By voices musicians mean a single line of sound, more or less continuous, that maintains a separate identity in a sound field or musical texture. The melody has certain characteristics that establish it as the predominant voice in the musical piece. Of course, a musical voice is not a succession of random notes – tones belonging to the same voice usually have a similar timbre, intervals between notes have a certain probability, there are rules regarding harmony and scale, and onset times of notes can be related to a rhythmical pattern.

Unfortunately the retrieval of high level musical features from polyphonic music is a challenging task in itself. Even for the most prominent voice (i.e. the melody), it is difficult to identify note onsets or to assign a note name to a tone with a varying frequency.

However, a melodic succession of tones has statistical properties that can be more easily exploited. Huron (2006) states that pitch proximity is the best generalization

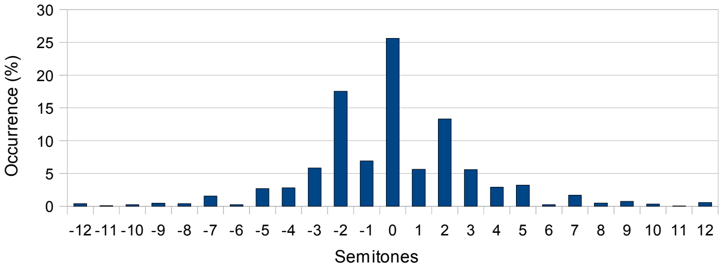


Figure 6.1. Histogram of Note Intervals in Melodies

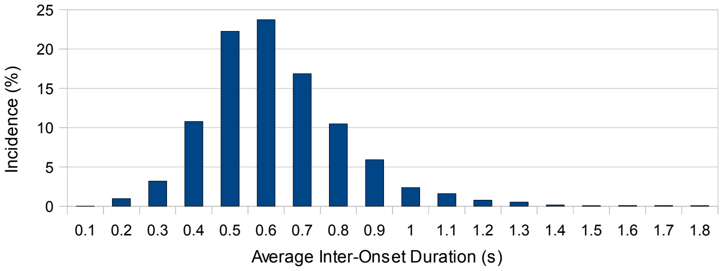


Figure 6.2. Histogram of the Average Note Duration in Melodies

about melodies. This statement is well supported by the interval statistics[1], as melodies consist mostly of tone sequences that are typically close to one another in pitch (see Figure 6.1). Indeed, the unison is the most frequent interval by a great margin, followed by the whole tone interval.

Other essential cues that help to distinguish musical voices are the central pitch tendency and regression to the mean (Huron, 2006, Chapter 5): the most frequently occurring pitches lie near the center of the melody's range. A necessary consequence of this tendency is the fact that after a melodic leap (an interval of more than three semitones) away from the center of the tone distribution, the following interval will change direction with a high probability. Regression to the mean is the most general explanation for this post-leap reversal.

The duration of melody tones lies normally in the range of 150 to 900 ms (see Figure 6.2). Notes at faster rates occur, but they usually do not contribute to the perception of melody (Warren, 1999, Chapter 5). If a familiar tune is played at a rate faster than approximately 50 ms per note, the piece will not be recognizable, although the global melodic contour can be perceived. Yet, a very slow playback (i.e. durations of more than one second) is possible.

[1] The Fraunhofer Institute in Ilmenau has gathered a collection of 6000 MIDI songs containing multiple genres, ranging from classical to contemporary charts music. Nearly one million notes were analyzed to compile a statistic of interval occurrences and the average note durations in melody tone sequences.

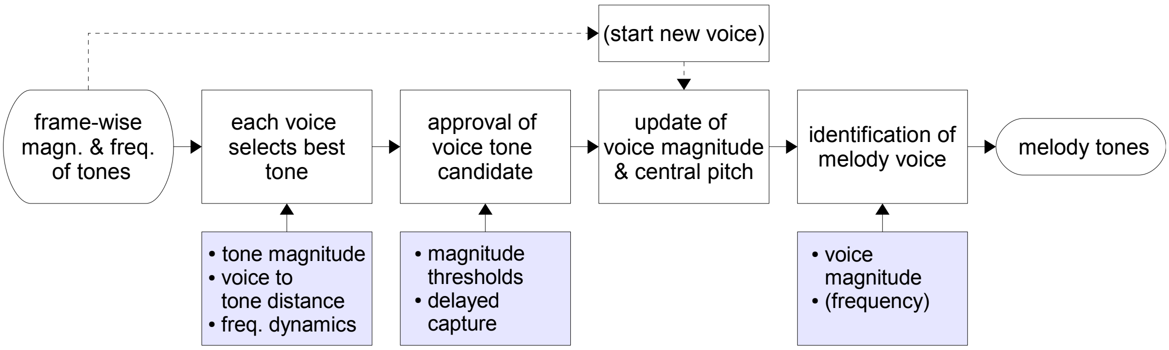

(start new voice)
frame-wise magn. & freq. of tones
each voice selects best tone
approval of voice tone candidate
update of voice magnitude & central pitch
identification of melody voice
melody tones
• tone magnitude
• voice to tone distance
• freq. dynamics
• magnitude thresholds
• delayed capture
• voice magnitude
• (frequency)

Figure 6.3. Overview of the voice estimation algorithm

The dynamic range, which denotes the ratio between the largest and smallest occurring magnitudes in a tone sequence, is another important cue. Usually, tones that belong to the same voice have more or less the same sound level. It should be noted, however, that especially the human singing voice has a rather high dynamic range with ratios of more than 20 dB between the loudest tones and the softest ones.

The process that is required by the human auditory system as it analyzes mixtures of simultaneous and sequential sound entities has been coined auditory scene analysis (Bregman, 1994). All of the aforementioned statistical properties of melodies in fact enable the sequential grouping of sounds by the human auditory system. These "primitive" grouping principles are not only valid for music, but also for speech, environmental sounds, and even for noise.

Still, the ability of humans to distinguish concurrently sounding voices is limited. Huron (1989) investigates the ability of musically trained listeners to continuously report the perceived number of voices in a polyphonic musical performance. While Huron questions the musical significance of his experiment, because it does not evoke a natural listening situation, one important take away is that there is a marked worsening of the human performance, when a three-voice texture is augmented to four voices. If errors occur, the number of voices is underestimated in 92 percent of the cases. Another finding of the experiment is the fact that inner voices are more difficult to detect. The reaction time for the identification of an inner voice is twice as long, and often they are not detected at all.

6.2. Overview

Figure 6.3 shows the processing steps performed in each analysis frame (i.e. every 5.8 ms). The input to the algorithm are the magnitude and frequency of the tone objects. The starting point of a new voice object is a salient tone that has not been added to an existing voice. In each analysis frame, every voice independently selects one tone, preferring strong tones that are close to its central pitch. If the selected tone passes all magnitude thresholds, it is added to the voice (after a certain delay period). The magnitude and central pitch of the voice are updated, whenever it has an added voice tone: the voice assembles a magnitude corresponding to the magnitude of the captured tone, and at the same time the voice's central pitch gradually moves towards the pitch of the added tone. Finally, the melody voice is chosen from the set of voices. The main criterion for the selection is the magnitude of the voice. Only tone objects of the current melody voice qualify as melody tones.

6.3. The Estimation of Musical Voices

In the scope of the melody extraction algorithm, a voice is an object that is defined by its magnitude $\bar{A}_{\text{voice}}$, its frequency $\bar{c}_{\text{voice}}$ and the frequency range $[c_{\min}, c_{\max}]$.

The formation of voices is controlled by the frame-wise updated magnitude and frequency of tone objects, which have a fundamental frequency in the range between 55 and 1319 Hz. The time advance between two successive analysis frames denotes 5.8 ms.

6.3.1. Start Conditions

The first question to ask is at which point a new voice should be started[2]. The conditions for starting a new voice object are as follows:

- A voice is started from a tone which was not included in an existing voice.
- The tone reached at least once the maximum magnitude among all other tones.
- The magnitude of the tone has passed at least once the global magnitude threshold.
- There is no voice that could capture the tone, or the duration of the tone is greater than 200 ms, or the tone was finished.

[2] The conditions given here are crafted for the purpose of melody extraction, which aims at the identification of the predominant melody line. If voices besides the predominant one shall be extracted, it is advisable to define more inclusive conditions.

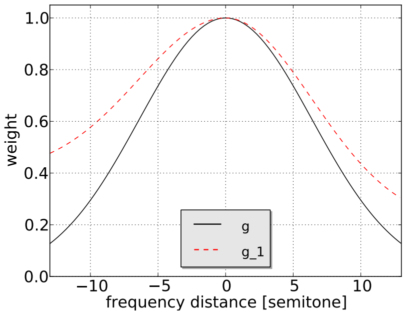


Figure 6.4. Weighting Functions

6.3.2. Selection of Voice Tone Candidates

In each analysis frame, the voice object searches for a strong tone in the frequency range of ±1300 cent around its current central pitch. The best choice, at the one hand, ensures the smoothness of the voice tone sequence, at the other hand, embraces tones with a strong magnitude. In contrast to most existing approaches using optimal path finding methods, the smoothness of the melody line is evaluated in terms of central pitch, and not with respect to the last added tone. This strategy might not give the best results in every situation, but it reinforces the importance of the central pitch, and allows an easier recovery after an erroneous addition of a tone.

The Rating of Voice Tone Candidates

Each voice independently chooses only one tone – the object with the maximum rating A_{rating}:

$$A_{\text{rating}} = C \cdot D \cdot A_{\text{tone}} \cdot g_1(\Delta c). \tag{6.1}$$

The rating is calculated from the following four criteria, which have been established by experiment:

- *Magnitude:* The tone magnitude A_{tone} is a good indicator for the perceptual importance of a tone.

- *Frequency distance weight:* The voice should preferably select a tone that is close to its central pitch. That is why the magnitude of the tone is weighted by a function that takes into account the frequency distance Δc between the tone's pitch c_{tone} and the central pitch of the voice $\bar{c}_{\text{voice}}$:

$$g_1(\Delta c) = r + (1 - r) \cdot g(\Delta c) \qquad \text{with } \Delta c = c_{\text{tone}} - \bar{c}_{\text{voice}}. \tag{6.2}$$

The parameter $r = 0.4$ if Δc is negative, otherwise $r = 0.2$, and g is the function

$$g(\Delta c) = e^{-0.5\frac{(\Delta c)^2}{640^2}}. \tag{6.3}$$

Figure 6.4 shows that the resulting weighting function g_1 is asymmetric – the weighting is biased towards tones from the lower frequency range. There are two reasons for this asymmetry. First, overtone errors cannot be avoided entirely, so in doubt the lower pitch is probably the true fundamental frequency. Second, tones in the lower frequency range of an instrument or the human voice are often softer, so the weighting compensates this difference.

- *Comparison with the average magnitude:* The magnitude of the selected tone candidate should be in the order of the previously added magnitudes. For the comparison we use the maximum tone magnitude $\hat{A}_{\text{tone}}$ and the long term exponential moving average (EMA) of the maximum tone magnitudes of previously added tones[3]. If $\hat{A}_{\text{tone}}$ is more than 10 dB below or above the long term average, the rating is halved. Accordingly a magnitude factor C is set to 1 or 0.5 in the final rating.

- *Frequency deviation:* Sounds with changing attributes attract attention. Human listeners particularly focus on tones with vibrato or pitch glides. If a tone shows persistently more than 20 cent frequency difference in between analysis frames the rating is doubled. Accordingly, a deviation factor D is set to 1 or 2 in the final rating.

Different Voices Competing for the Same Tone

Any tone object preferably belongs to only one voice. In practice, there are often ambiguous situations in which an exclusive assignment to one voice is not the optimal solution.

The priority is on voices with a larger voice magnitude: this means a previously added tone may still be added by another voice, if the new owner has a larger magnitude than the current owner of the tone. Having said that, any voice which has a smaller magnitude than the current owner of the tone is prohibited to add the tone. The priority on strong voices is also reflected in the selection of the tones which was described in the previous section. The aim is that weaker voices shall avoid tones that are already added to strong voices (i.e. a voice with a large magnitude $\bar{A}_{\text{voice}}$). Hence, two more rating factors are introduced to direct the attention of weak voices to other suitable tone candidates:

- *Comparison of voice magnitude:* Whenever the tone is already included in a stronger voice, the original rating A_{rating} is multiplied with the factor 0.7.

[3] A detailed description of the exponential moving average can be found in the appendix.

- *Comparison of voice bidding:* If two voices aim at the same tone, the rating A_{rating} is decreased by the factor 0.7 for the voice with the lower bidding, but only if it is also the weaker voice. The voice bidding is the product of voice magnitude and the distance weight given in equation 6.2: $A_{\text{bidding}} = \bar{A}_{\text{voice}} \cdot g_1(\Delta c)$.

As the voice magnitudes and the voice biddings of the current frame are not known prior to the tone selection process, the values of the last analysis frame are used for the comparison. As the values usually change rather slowly, they are still significant. Furthermore, this provision ensures that the output is independent of the explicit order in which voices bid for tones.

6.3.3. Approval of Voice Tones

Even though one voice tone candidate is selected in each analysis frame, it is not clear whether the particular tone belongs to the voice or not, as melodies also contain rests. Two different techniques are employed to perform the voicing detection, namely the use of adaptive magnitude thresholds and the delayed capture of tones.

The magnitude thresholds are characterized by a quick rise time and a slow exponential decay. The decay of the threshold is determined by its distinct half life period, which may comprise durations from 150 ms up to several seconds.

Short Term Magnitude Threshold

The short term magnitude threshold is estimated for each voice individually. It secures that shortly after a tone is finished no weak tone is added to the voice prematurely. Hence, it is especially useful to bridge small gaps between tones of a voice. The short term threshold is adaptive and decays with a half-life time of 150 ms. Whenever the current voice tone has a magnitude which is larger than the current threshold reference value $T_{150\text{ms}}$, the threshold is updated to the new maximum. If the currently observed magnitudes are smaller than the threshold, it decays with the given half-life time:

$$T_{150\text{ms}} \leftarrow \begin{cases} A_{\text{tone}}, & \text{if} \quad A_{\text{tone}} > T_{150\text{ms}}; \\ \alpha_{150\text{ms}} \cdot T_{150\text{ms}}, & \text{otherwise.} \end{cases} \tag{6.4}$$

The parameter $\alpha_{150\text{ms}}$ controls the decay of the magnitude threshold. The calculation of its value is described in equation A.5. The tone passes the threshold if it is no more than 6 dB below $T_{150\text{ms}}$.

Long Term Magnitude Threshold

The long term magnitude threshold $T_{5\mathrm{s}}$ is basically the same as the short term threshold, with the distinction that it decays with a half-life period of 5 seconds. In order to pass the threshold, the tone's magnitude should not be more than 20 dB below $T_{5\mathrm{s}}$.

Long Term EMA Magnitude Threshold

A high dynamic range of 20 dB within a tone sequence is not exceptional – a prominent example is the human singing voice. However, if a relatively high dynamic range is allowed, many tones from the accompaniment will pass the magnitude threshold, too. This is especially true for instrumental music that often contains several simultaneous voices with a comparable strength. Besides the long term threshold, which is based on the maximum magnitude, another threshold is introduced which is computed as the exponential moving average of the previously added voice tone magnitudes. This threshold is updated whenever the voice has an approved voice tone, provided that the tone's duration is between 50 and 500 ms:

$$T_{\mathrm{EMA\_5s}} \leftarrow \alpha_{5\mathrm{s}} \cdot T_{\mathrm{EMA\_5s}} + (1 - \alpha_{5\mathrm{s}}) \cdot \hat{A}_{\mathrm{tone}}. \tag{6.5}$$

The EMA is estimated with the current peak magnitude $\hat{A}_{\mathrm{tone}}$, which denotes the biggest magnitude the tone has reached so far. At the start of the voice the magnitude threshold is set to one third of the maximum magnitude of the first added voice tone. As the threshold reflects the dynamic range of previous voice tone magnitudes, the actual threshold value can be defined more strictly. In order to pass the threshold, the tone's magnitude should not be more than 10 dB below $T_{\mathrm{EMA\_5s}}$.

Delayed Capture of a Tone

The approval of a new voice tone is often delayed to allow some time for the start of a more suitable tone. The delay time depends on the distance between the candidate voice tone and the preferred frequency range of the voice (see Section 6.3.4). All tones within the preferred frequency interval are added immediately, provided that they pass the magnitude thresholds. All other tones face a delay that depends on their magnitude and the frequency distance between tone and the preferred frequency range.

In order to estimate the delay, a short term pitch $\bar{c}_{\mathrm{st}}$ is defined for each voice object. (The computation of $\bar{c}_{\mathrm{st}}$ is described in Section 6.3.4.) The tone may only be added after $\bar{c}_{\mathrm{st}}$ has approximately reached the frequency of the voice tone candidate (i.e. less than 100 cent distance). Figure 6.5 illustrates the delayed capture of alternating tones.

6.3.4. Update of Voice Parameters

Magnitude Update

The voice magnitude $\bar{A}_{\text{voice}}$ is updated whenever the voice has an approved voice tone. The magnitude depends on the tone's rating magnitude A_{rating} as given in equation 6.1. The use of the rating magnitude ensures that a voice profits more from tones that are close to its current central pitch. In order to update the magnitude values, we use the exponential moving average (EMA).

$$\bar{A}_{\text{voice}} \leftarrow \alpha_{500\text{ms}} \cdot \bar{A}_{\text{voice}} + (1 - \alpha_{500\text{ms}}) \cdot A_{\text{rating}}. \tag{6.6}$$

The parameter $\alpha_{500\text{ms}}$ is a smoothing factor that corresponds to a half-life period of 500 ms. The EMA calculation is initialized with a fraction of the peak magnitude of the first tone: $\bar{A}_{\text{voice}} = 0.2 \cdot \hat{A}_{\text{tone}}$.

Central Pitch

The central pitch of the voice $\bar{c}_{\text{voice}}$ is an important parameter, as it defines the preferred frequency range for the selection of tones (see Figure 6.5). It is established over time according to the pitches of approved voice tones. While the adaptation could be implemented as EMA of previous frequencies, it is beneficial if the adaptation speed also depends on the tone's magnitude. This means the central pitch moves faster towards strong tones. That is the reason why at first a weight $\bar{A}_{\text{w}}$ is defined that allows to evaluate the current rating of a tone in relation to the EMA of previous ratings:

$$\bar{A}_{\text{w}} \leftarrow \left(\bar{A}_{\text{w}} - A_{\text{rating}}\right) \alpha_{500\text{ms}} + A_{\text{rating}}. \tag{6.7}$$

The EMA is initialized with $\bar{A}_{\text{w}} = 0.2 \cdot \hat{A}_{\text{tone}}$ at the beginning of the voice. With the help of the weight $\bar{A}_{\text{w}}$ we can finally update the central pitch:

$$\bar{c}_{\text{voice}} \leftarrow \frac{\bar{A}_{\text{w}}\bar{c}_{\text{voice}} + (1 - \alpha_{500\text{ms}}) \cdot A_{\text{rating}} \cdot c_{\text{tone}}}{\bar{A}_{\text{w}} + (1 - \alpha_{500\text{ms}}) \cdot A_{\text{rating}}}. \tag{6.8}$$

The parameter $\alpha_{500\text{ms}}$ is a smoothing factor, which corresponds to a half-life period of 500 ms[4]. The parameter A_{rating} refers to the rating magnitude of the voice tone as given in equation 6.1, and c_{tone} is the pitch of the voice tone. The initial value for the iterative calculation is the frequency of the first added voice tone: $\bar{c}_{\text{voice}} = c_{\text{tone}}$. As $\bar{A}_{\text{w}}$ is close to zero at the start of the voice, the central pitch changes more rapidly

[4] Since the weight $\bar{A}_{\text{w}}$ depends on many factors, the parameter α does not exactly set any half-life period for the central pitch update. Yet the corresponding time span gives a reference point for the approximate adaptation speed.

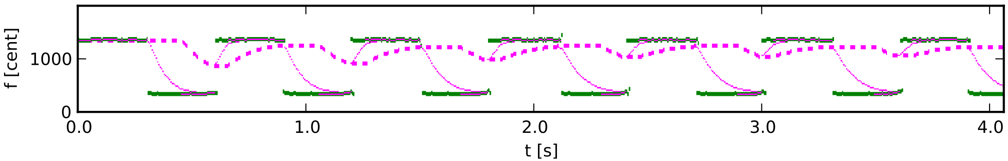


Figure 6.5. Alternating tones: dashed line - central pitch of the voice $\bar{c}_{\text{voice}}$, thin line - short term pitch of the voice $\bar{c}_{\text{st}}$.

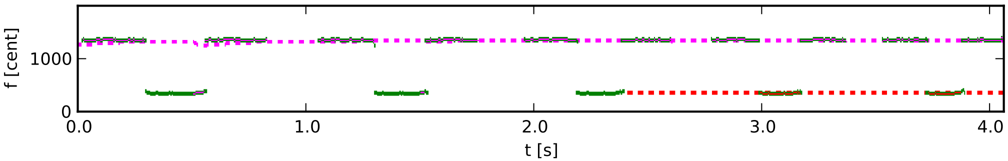


Figure 6.6. Example of alternating tones: The duration of the tones is decreased over time. Soon the alternating tones cannot be captured by the first voice and a second voice is started.

after the start of the voice (see also Figure 6.5). This is, however, a deliberate decision, as the "true" central pitch has to be established over a longer time period.

Sometimes the frequency of a tone sequence does not prevail close to a central pitch, but moves upwards or downwards in one direction. As the central pitch adapts quite slowly, the update might not be fast enough to capture the succession of tones, and soon the tones fall outside the maximum search range of the voice. To avoid this, there is an immediate update of the central pitch, if $|\bar{c}_{\text{voice}} - c_{\text{tone}}| > 900$. In this case the central pitch is set to the maximum distance of 900 cent.

Frequency Range

Intervals that are greater than an octave are rarely found in melody tone sequences. Consequently the search range for voice tones is limited to the range of ±1300 cent around the central pitch of a voice.

Moreover, a preferred frequency range R_{pref} is defined which is given by the frequency range between the last added voice tone frequency and the central pitch of the voice.

Short Term Pitch

The short term pitch $\bar{c}_{\text{st}}$ seeks to emulate the time that is needed to focus attention to a tone that is outside the preferred frequency range of the voice (see Figure 6.5). It is updated whenever the voice tries to capture a new voice tone, so it is updated even without an approved voice tone.

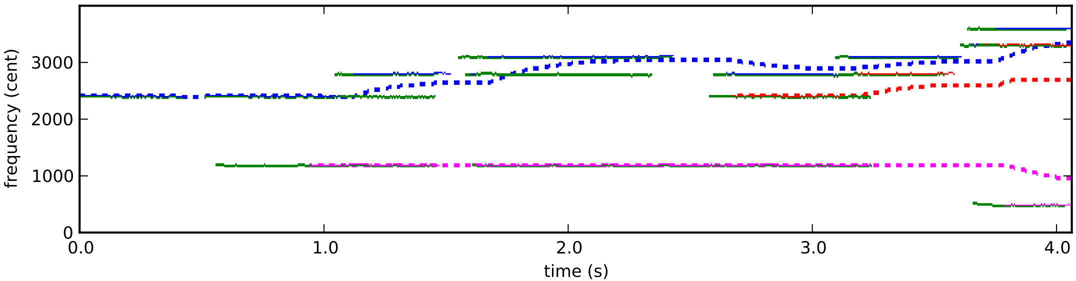


Figure 6.7. Three voices are recognized – the melody voice (blue), the bass voice (magenta), and an inner voice (red).

The short term pitch $\bar{c}_{\mathrm{st}}$ can immediately be set to any frequency within the preferred frequency range R_{pref}. So if the distance to a voice tone candidate can be decreased by changing the short term pitch to a frequency within R_{pref}, $\bar{c}_{\mathrm{st}}$ is set to that value. Apart from that, the short term pitch is updated very much like the central pitch of the voice – namely by using a weighted EMA. At first, a weight $A_{\mathrm{w\_st}}$ is defined that allows to compare the tone's current rating A_{rating} with the magnitude of previously added voice tones. For this purpose we determine $A_{\mathrm{w\_st}}$ as the average of the long term EMA magnitude threshold $T_{\mathrm{EMA\_5s}}$ and the short term magnitude threshold T_{150ms}:

$$A_{\mathrm{w\_st}} = 0.5 \cdot (T_{\mathrm{EMA\_5s}} + T_{\mathrm{150ms}}). \tag{6.9}$$

Finally, the short term pitch is updated using a weighted EMA:

$$\bar{c}_{\mathrm{st}} \leftarrow \frac{A_{\mathrm{w\_st}}\bar{c}_{\mathrm{st}} + (1-\alpha_{\mathrm{30ms}}) \cdot A_{\mathrm{rating}} \cdot c_{\mathrm{tone}}}{A_{\mathrm{w\_st}} + (1-\alpha_{\mathrm{30ms}}) \cdot A_{\mathrm{rating}}}. \tag{6.10}$$

The parameter α_{30ms} is again the smoothing factor. Figure 6.5 shows how $\bar{c}_{\mathrm{st}}$ is used to capture tones: only if the thin line reaches the voice tone candidate (i.e. less than 100 cent distance), the tone may be added to the voice.

6.4. The Identification of the Melody Voice

The most promising feature to distinguish melody tones from all other sounds is the magnitude. The magnitude of the tones is of course reflected by the voice magnitude. Hence, the voice with the highest magnitude is in general selected as the melody voice. It may happen that two or more voices have about the same magnitude and thus no clear decision can be taken. In this case, the voices are weighted according to their frequency: voices in very low frequency regions receive a lower weight. The magnitude thresholds are defined for each voice individually. As they depend solely on the past tones of the voice, they cannot take effect on all soft tones. Therefore, it is recommended that a global magnitude threshold is estimated from the identified melody tones. Subsequently, the melody tones should be compared to the global threshold.

Chapter 7.

Evaluation

7.1. Introduction

This chapter is dedicated to the quantitative evaluation of the proposed melody extraction system. While the overall performance of the system is important, another interesting fact is the impact of individual processing modules. That is why we also present some evaluation results for intermediate output that covers the following areas: predominant pitch estimation, multiple fundamental frequency estimation and tone tracking, and melody extraction.

The available ground truth offers some variety, but it does not cover the whole bandwidth of real-world music. Often individual datasets are very small. Other datasets have a huge number of files, but nonetheless only include a certain type of music. If the estimation accuracy is given for a small or biased dataset it might not correlate well with the real performance of the algorithm.

If algorithms are based on a machine learning approach, it is assumed that the training dataset is independent from the test set, so that a possible overfitting of the statistical model can be easily detected. But actually, the problem of overfitting can also be found in rule-based approaches. As long as the available ground truth consists only of small datasets, an evaluation should always be conducted on datasets that have not been used for parameter estimation.

One opportunity to do so is the Music Information Retrieval Evaluation eXchange (MIREX), an initiative that hosts a number of music information retrieval tasks (Downie et al., 2014). The algorithm performance was evaluated in three MIREX tasks, namely multiple fundamental frequency estimation, tone tracking, and audio melody extraction. In the following section we will present and discuss the results.

7.2. Predominant Pitch Determination

The evaluation of the predominant pitch estimation is not directly covered by MIREX tasks. That is the reason why we present some results for those processing steps using publicly available datasets, and hope that the data allows some insight. The pitch determination algorithm was evaluated in (Dressler, 2011) and the results are given below.

7.2.1. Evaluation Metrics

In spite of the promising MIREX results (see the subsequent sections), it is difficult to identify the contribution of the pitch extraction method to the overall performance, because there are more processing steps involved in the extraction of the melody. Salamon and Gómez (2009) have estimated the potential performance of a chroma-based pitch salience function by considering an increasing number of salient pitch peaks: presuming an ideal pitch selection process, the melody is identified correctly as soon as one of the peak candidates matches the transcribed reference frequency. They tested the performance of the pitch salience function on the publicly available dataset ADC 2004 – a dataset containing polyphonic music (20 excerpts of about 20s including MIDI, Jazz, Pop and Opera music as well as audio pieces with a synthesized voice) as well as the melody transcriptions (timestamps versus melody pitch every 10 ms).

We adopt the idea of Salamon and Gómez (2009) for the evaluation of the proposed algorithm, however, using modified conditions for the analysis. Since the above-quoted approach uses chroma features, octave errors are not detected. In our evaluation, the reference frequency is not mapped to an octave range. The pitch is identified correctly if the frequency is less than 50 cent away from the ground truth. Moreover, a magnitude threshold that lies 10 dB below the maximum pitch magnitude of the analysis frame is imposed on the pitch candidates.

7.2.2. Results and Discussion

Figures 7.1 and 7.2 show that the estimation accuracy converges towards a "glass ceiling" with a rising number of pitch candidates. The limiting value for the ADC 2004 database amounts to 93%. It does not differ significantly from the value of about 90% given in (Salamon and Gómez, 2009). However, an improvement can be noted for the most salient pitch peak (77 versus 71 % in (Salamon and Gómez, 2009)), even though the conditions used for our evaluation are more strict[1].

[1] Unfortunately the estimation results for the MIREX 2005 development collection (MIREX05 train) cannot be compared meaningfully since the reference data has been corrected only recently.

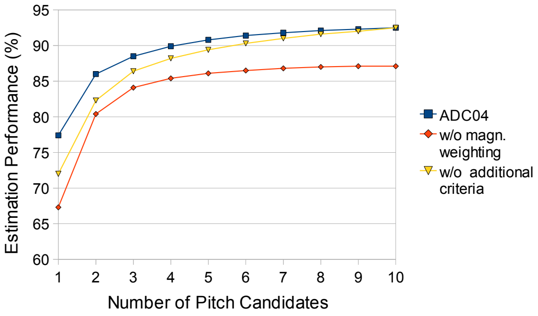


Figure 7.1. ADC04 dataset: Potential Performance vs Peak Number

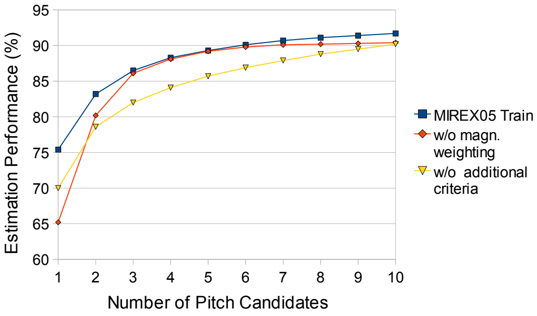


Figure 7.2. MIREX05 train: Potential Performance vs Peak Number

Another interesting aspect is the impact of the different parameters on the algorithm performance (see figures 7.1 and 7.2). The most important individual processing step – apart from the basic algorithm structure – is the magnitude weighting introduced in Section 3.5. At first sight, the magnitude weighting seems to be counterproductive, because it in fact takes away power from the fundamental frequency. Yet, during subsequent processing the fundamental frequency will also profit from the strong weighting of the overtones.

Still, one significant advantage remains: the notes from a potential bass voice are damped while at the same time the melody notes are boosted. As a consequence the greatest impact of the magnitude weighting is observed if the audio recordings contain musical voices of comparable strength. In fact, the improvement of the estimation accuracy can be attributed rather to the improvement in individual test files than to general characteristics of the data collections. For example, the estimation accuracy is increased markedly for test files with a strong bass voice (for example midi1: +52%, midi4: +58%, train12: +42%, train13: +38%), while the detection of the predominant pitch is slightly worse for files which have additional voices with a higher frequency than the actual melody voice (daisy2: -10%, train10: -8%).

No other criterion or parameter has such a marked effect on the estimation accuracy like the magnitude weighting. Yet, the small contributions of the individual measures sum up to a significant improvement, as can be noted from the yellow curve in the diagram. Here, the spectral smoothing, the attenuation by intermediate peaks, and the harmonic number weighting have been omitted. Instead, the weighted spectral magnitude A_{peak} was added as virtual magnitude to the pitch spectrogram.

Surprisingly, the spectral envelope smoothing has no significant effect on the estimation of the predominant voice in the frame-wise evaluation. This may be contributed to the fact that most of the music pieces tested have a strong melody voice. However, the spectral smoothing plays an important role for the estimation of multiple fundamental frequencies.

7.3. Multiple Fundamental Frequency Estimation

The tone processing front-end of the melody extraction algorithm was evaluated at the MIREX multiple fundamental frequency estimation task in 2014 (Dressler, 2014a). The goal of this task is to identify the fundamental frequencies (F0) on a frame-by-frame basis. Algorithm parameters (for example the FFT window size, parameters for the pitch estimation and tone tracking, as well as the timing constants of the adaptive thresholds) have been adjusted using the melody extraction training data of ISMIR 2004 and MIREX 2005. However, there is one modification for the MIREX multiple-F0 task: the maximum allowed frequency range for tones was increased to cover frequencies between 55 Hz and 2093 Hz. Nonetheless, it should be noted that the used parameter setting is probably not the best choice to maximize the estimation accuracy for the multiple-F0 task – in particular, as the dataset for melody extraction consists mostly of musical pieces with a singing voice, while the multiple-F0 dataset includes solely instrumental music.

40 test files were analyzed for this task: 20 excerpts from recordings of woodwind, bassoon, clarinet, horn, flute and oboe (with a polyphony ranging from 2 to 5), 12 excerpts from a quartet recording of bassoon, clarinet, violin and sax (with a polyphony ranging from 2 to 4), and 8 files from synthesized MIDI (with a polyphony ranging from 2 to 5).

7.3.1. Evaluation Metrics

A pitch estimate is assumed to be correct if it is within a half semitone (±50 cent) of a ground-truth pitch for that frame. Only one ground-truth pitch can be associated with each returned pitch. Two different sets of evaluation metrics are used to estimate the algorithm performance. The first set estimates the performance in terms of precision, recall and overall accuracy using the following equations:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{7.1}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{7.2}$$

$$\text{Accuracy} = \frac{TP}{TP + FP + FN}, \tag{7.3}$$

in which TP is the number of correctly identified pitches (true positives), FP is the number of identified pitches which do not occur in the ground truth (false positives), and FN is the number of pitches that are not identified by the algorithm (false negatives).

The second set of evaluation metrics was proposed by Poliner and Ellis (2007) in order to measure the accuracy of polyphonic piano transcriptions. The metric computes an error score E_{tot} that takes into account the so-called substitution errors E_{subs}, which allow the substitution of any false positive F0 with a missing ground-truth F0. The number of errors is set into relation to the total quantity of notes:

$$E_{\text{subs}} = \frac{\sum_{t=1}^{T} \min(N_{\text{ref}}(t), N_{\text{sys}}(t)) - N_{\text{corr}}(t)}{\sum_{t=1}^{T} N_{\text{ref}}(t)}, \tag{7.4}$$

in which N_{ref} is the number of pitches in the ground truth data, N_{sys} is the number of pitches returned by the system, N_{corr} is the number of correctly identified pitches, and t is the index of the current analysis frame.

The other components of the metric are missing pitches E_{miss} and false alarm errors E_{fa}. While E_{miss} refers to the number of ground-truth reference notes that could not be matched with any system output (i.e. misses after substitutions are accounted for), E_{fa} refers to the number of pitches that cannot be paired with any ground truth (false alarms beyond substitutions):

$$E_{\text{miss}} = \frac{\sum_{t=1}^{T} \max(0, N_{\text{ref}}(t) - N_{\text{sys}}(t))}{\sum_{t=1}^{T} N_{\text{ref}}(t)}, \tag{7.5}$$

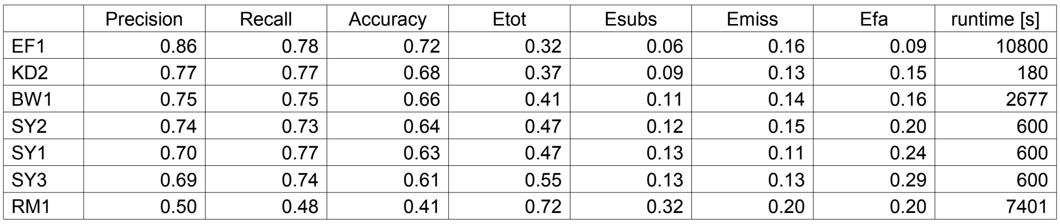
$$E_{\text{fa}} = \frac{\sum_{t=1}^{T} \max(0, N_{\text{sys}}(t) - N_{\text{ref}}(t))}{\sum_{t=1}^{T} N_{\text{ref}}(t)}. \tag{7.6}$$

The total error is estimated as follows:

$$E_{\text{tot}} = \frac{\sum_{t=1}^{T} \max(N_{\text{ref}}(t), N_{\text{sys}}(t)) - N_{\text{corr}}(t)}{\sum_{t=1}^{T} N_{\text{ref}}(t)}. \tag{7.7}$$

7.3.2. Results and Discussion

Table 7.1. Task 1: Multiple Fundamental Frequency Estimation Results



	Precision	Recall	Accuracy	Etot	Esubs	Emiss	Efa	runtime [s]
EF1	0.86	0.78	0.72	0.32	0.06	0.16	0.09	10800
KD2	0.77	0.77	0.68	0.37	0.09	0.13	0.15	180
BW1	0.75	0.75	0.66	0.41	0.11	0.14	0.16	2677
SY2	0.74	0.73	0.64	0.47	0.12	0.15	0.20	600
SY1	0.70	0.77	0.63	0.47	0.13	0.11	0.24	600
SY3	0.69	0.74	0.61	0.55	0.13	0.13	0.29	600
RM1	0.50	0.48	0.41	0.72	0.32	0.20	0.20	7401

Table 7.1 shows the results for the MIREX task "frame-wise multiple F0 estimation" (Dressler, 2014a). The accuracy of our submission KD2 marks the second best result (68%) out of seven submissions – the best algorithm was submitted by Anders

Elowsson and Anders Friberg reaching an excellent overall accuracy of 72 percent (Elowsson and Friberg, 2014).

Compared with our result of the year 2012, the accuracy of our algorithm has improved by 4 percent. Thereby the main improvement lays in the better estimation of the number of concurrent voices, leading to a higher recall (0.77 instead of 0.66). While a better trade-off between precision and recall might be achieved by using a more suitable dataset for the parameter estimation, it may not be possible to reach a much better result without loosing some generality in terms of the input data. The difference between accuracy (68%) and accuracy chroma (70%) denotes only 2 percent, showing that octave errors are not a huge problem for the algorithm. It can also be noted that the submitted algorithm stands out due to a very short run-time, being sixty times faster than the highest ranked submission.

7.4. Tone Tracking

The tone processing front-end of the melody extraction algorithm was evaluated at the MIREX multiple fundamental frequency tracking task in 2014 (Dressler, 2014a). The goal of this task is to track the frame-wise F0 estimates and form tones by giving an onset time, offset time and a distinct tone-height. A total of 34 files were analyzed in this subtask: 16 excerpts from woodwind recordings, 8 excerpts from the IAL quintet recording and 6 piano recordings.

7.4.1. Evaluation Metrics

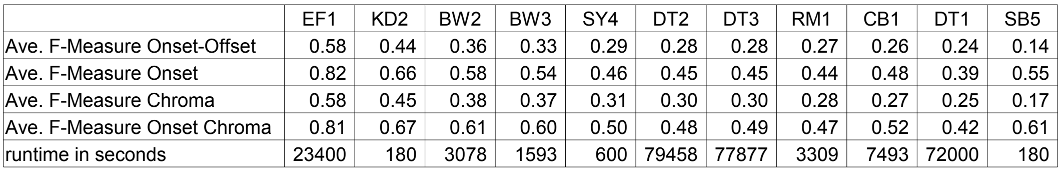
For this task the F-Measure is reported, which is the harmonic mean of precision and recall (see equations 7.1 and 7.2) for each input file:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{7.8}$$

Then, the average is calculated from the results of the individual files.

As stated in the MIREX Wiki of this task, a ground truth note is assumed to be correctly transcribed if the transcription system returns a note that is within a half semitone of that note, and the returned note onset is within a 100 ms range (±50 ms) of the onset of the ground truth note, and its offset is within a 20% range of the ground truth note offset. The evaluation of the note offset is omitted in the "onset-only" subtask.

Table 7.2. Task 2: Tone Tracking Results



	EF1	KD2	BW2	BW3	SY4	DT2	DT3	RM1	CB1	DT1	SB5
Ave. F-Measure Onset-Offset	0.58	0.44	0.36	0.33	0.29	0.28	0.28	0.27	0.26	0.24	0.14
Ave. F-Measure Onset	0.82	0.66	0.58	0.54	0.46	0.45	0.45	0.44	0.48	0.39	0.55
Ave. F-Measure Chroma	0.58	0.45	0.38	0.37	0.31	0.30	0.30	0.28	0.27	0.25	0.17
Ave. F-Measure Onset Chroma	0.81	0.67	0.61	0.60	0.50	0.48	0.49	0.47	0.52	0.42	0.61
runtime in seconds	23400	180	3078	1593	600	79458	77877	3309	7493	72000	180

7.4.2. Results and Discussion

Table 7.2 shows the results of the note tracking task. Four different measures are given: Average F-measure Onset-Offset, Average F-measure Onset, Average F-Measure Chroma, and Average F-measure Onset Chroma. In general, it is much easier to detect note onsets than note offsets – a fact that becomes apparent if the onset-offset categories are compared with the onset only categories. The chroma measure does not count octave errors.

Reaching an average F-measure of 0.44, our algorithm (labeled KD2) ranked second out of eleven submissions in the most important category (Average F-measure Onset-Offset). The best result of 0.58 is marked by the submission of Elowsson and Friberg. Again, the chroma result (0.45) differs not much from the regular F-measure. This means that octave errors don't play a mayor role in the transcription results.

The proposed algorithm is the fastest algorithm among all submissions. It runs 130 times faster than the highest ranked submission.

7.5. Audio Melody Extraction

The melody extraction algorithm was evaluated at the MIREX audio melody extraction task in 2014 (Dressler, 2014b). The aim of this task is to extract melodic content from polyphonic audio. Four datasets were available for the evaluation:

- MIREX09: 374 excerpts of 20-40s of Chinese Karaoke songs (singing voice, synthetic accompaniment). The same database was tested with different melodic voice to accompaniment ratios. (+5dB, 0dB, and -5 dB RMS)
- MIREX08: 8 excerpts of 60s from north Indian classical vocal performances.
- MIREX05: 25 excerpts of 10-40s from the following genres: Rock, R&B, Pop, Jazz, Solo classical piano.
- ADC04: 20 excerpts of about 20s including MIDI, Jazz, Pop and Opera music as well as audio pieces with a synthesized voice.

The corresponding reference annotations of the predominant melody include a succession of pitch frequency estimates at discrete time instants (5.8/10 ms grid). Zero frequencies indicate time periods without melody. The estimated frequency was considered correct whenever the corresponding ground truth frequency is within a range of ±50 cents.

7.5.1. Evaluation Metrics

To maximize the number of possible submissions, the transcription problem was divided into two subtasks, namely the melody pitch estimation and the distinction of melody and non-melody parts (voiced/unvoiced detection). It was possible to give an additional pitch estimate for the frames that are declared unvoiced by the algorithm. Those frequencies are marked with a negative sign.

In order to evaluate the results of the proposed melody extraction system five different measures are given.

The Overall Accuracy is the probability that voiced frames GV and unvoiced frames GU of the ground truth are correctly identified and the correct pitches TPC are assigned within half a semitone (±50 cent), e.g.

$$\text{Overall Accuracy} = \frac{TPC + TN}{GV + GU}, \tag{7.9}$$

in which TN are the correctly identified true negative frames (i.e. the frames where no melody is playing).

The Raw Pitch Accuracy considers only voiced frames GV and also evaluates the correctly identified additional pitch guesses FNC that are given for the frames labeled as unvoiced:

$$\text{Raw Pitch Accuracy} = \frac{TPC + FNC}{GV}, \tag{7.10}$$

in which TPC are the correctly identified melody pitches that are labeled as voiced.

Raw Chroma Accuracy is the probability that the chroma (i.e. the note name) is correct over the voiced frames GV. This measure ignores octave errors and like Raw Pitch Accuracy also evaluates the correctly identified additional pitch guesses $FNCchr$:

$$\text{Raw Chroma Accuracy} = \frac{TPCchr + FNCchr}{GV}. \tag{7.11}$$

Voicing Recall is the probability that a frame which is truly voiced is labeled as voiced:

$$\text{Voicing Recall} = \frac{TP}{GV}, \tag{7.12}$$

in which TP is the number of correctly labeled voiced frames and GV is the number of voiced frames in the ground truth.

Voicing False Alarm is the probability that a frame which is not actually voiced is nonetheless labeled as voiced:

$$\text{Voicing False Alarm} = \frac{FP}{GU}, \tag{7.13}$$

in which FP is the number of frames that are erroneously labeled as voiced, and GU is the number of unvoiced frames in the ground truth.

7.5.2. Results and Discussion

Table 7.3. Melody Extraction Results of MIREX 2014 (the starred result did not perform voicing detection)

Algorithm	Overall Accuracy (%)	Raw Pitch (%)	Raw Chroma (%)	Voicing Recall (%)	Voicing False Alarm(%)
SG2 (2011)	**75.1**	79.5	82.3	86	23.7
KD3	73.3	**80.6**	**82.5**	90.9	41.0
KD1	72.2	79.3	81.3	86.4	32.5
DD1	71.4	72.2	75.0	85.9	29.6
CWJ3	67.5	73.4	75.1	73.8	19.7
IYI1	59.8	68.9	73.1	84.5	39.3
SL1	57.1	52.4	55.3	73.2	22.6
YJ2*	54.0	74.8	78.6	100.0	99.6
LPSL1	40.6	40.5	46.6	67.8	35.9

The unweighted average[2] of the evaluation results for all datasets is shown in Table 7.3, in which our submissions are labeled by the submission shortcodes KD1 and KD3. The submission KD1 includes a new multiple F0 estimation front-end that was developed for the MIREX multiple F0 estimation and tracking task. Submission KD3 contains a pitch estimation front-end that places more weight on the predominant periodicity.

The Overall Accuracy is the most important statistic, because it evaluates the segmentation between melody and non-melody parts as well as the pitch detection[3]. The MIREX 2014 evaluation results show that the presented algorithms have the best Overall Accuracy for the unweighted average of all datasets among the eight

[2] Traditionally in MIREX, the results for each dataset have been weighted by the number of files in the dataset. However, the inclusion of the huge MIREX09 dataset makes this option questionable, because the weighted average would eventually measure the algorithm performance on Chinese Karaoke songs.

[3] The starred submission did not perform voiced/unvoiced detection, so the overall accuracy cannot be meaningfully compared to other systems.

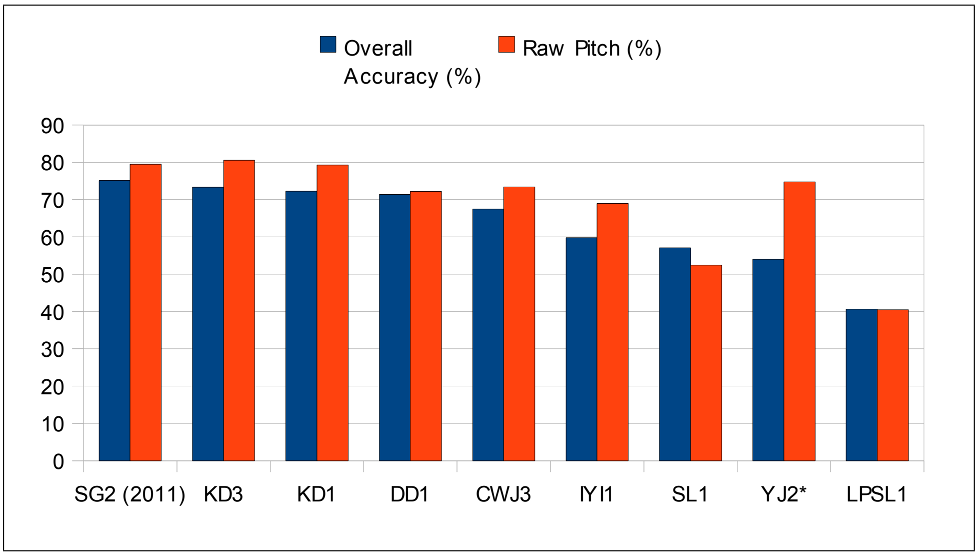


Figure 7.3. Comparison between Raw Pitch Accuracy and Overall Accuracy

participating algorithms. However, they do not reach the outstanding result of Justin Salamon's submission for MIREX 2011 (Salamon and Gómez, 2011).

It seems that our algorithms cope well with instrumental music, as there is no performance break-down for audio input with an instrumental lead voice (datasets ADC04 and MIREX05). One reason (besides the presented approach to auditory streaming) might be that the proposed method is not specifically adapted to a human melody voice. This might also explain the moderate results for the MIREX09 database that contains Chinese karaoke songs. A comparison between submissions KD1 and KD3 shows that the multiple F0 estimation front-end does not improve the Overall Accuracy of the melody extraction. A reason might be that the multiple F0 estimation enhances the tones of the accompaniment, so that it becomes more difficult to extract the predominant melody voice. Nonetheless, we see potential in the multiple F0 estimation front-end as it also extracts timbre information for individual tones.

The algorithm KD3 reaches the best values for the Raw Pitch Accuracy and the Raw Chroma Accuracy, showing that a better balance between Voicing Recall and Voicing False Alarm might improve the Overall Accuracy of the algorithm.

Unfortunately, no runtime data was published in the year 2014. Personal communication with the task captain of the audio melody extraction task revealed that our algorithm was the fastest among all submissions running about 15 minutes to complete all datasets.

Figure 7.4 shows the trend of the MIREX audio melody evaluation results for dif-

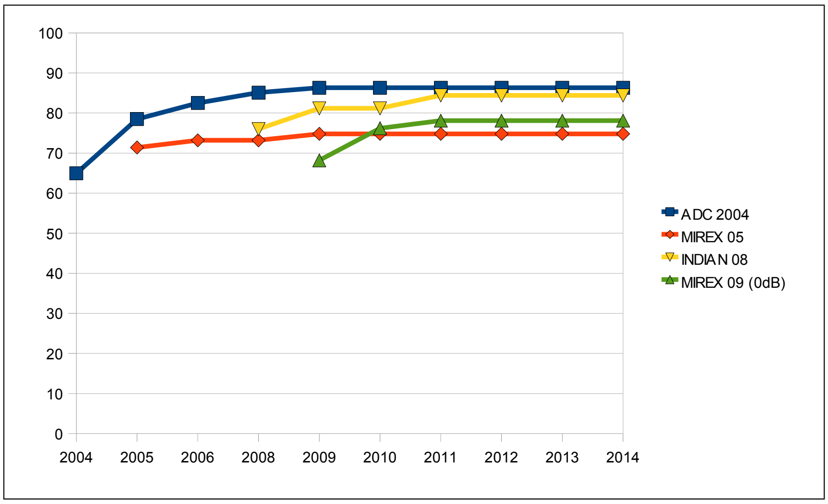


Figure 7.4. Evaluation Trends in Audio Melody Extraction (best Overall Accuracy that was reached for distinct datasets)

ferent datasets[4]. There is clearly much room for improvement in the future, though it can be noted that there was no increase in the melody extraction accuracy during the last three years. In current systems, melody and accompaniment are mainly separated by their sound intensity. It is obvious that much better results cannot be achieved with this feature alone. For example, the human voice has a high dynamic range (about 20 dB) and the instrumental accompaniment naturally reaches the volume of the softer human melody parts. Thus, more high-level features have to be incorporated into the melody extraction process to achieve better results.

[4] As the dataset ADC 04 was no official test set in the year 2005, the graph shows the performance of the author's submission to MIREX 2005 on this dataset.

Chapter 8.

Conclusion and Perspectives

8.1. Summary and Conclusions

Having reached the last chapter of this thesis, it is time to summarize the work. The Chapter “State-of-the-Art” discussed trends in the methodology that have appeared in recent years and showed that the influence of auditory models has decreased in favor of spectral analysis using FFT. We also noted a decline of pitch estimators using the autocorrelation function - at present, the prevailing pitch estimation method is the subharmonic summation. We found that most melody extraction algorithms exploit two characteristics to identify the melody voice: the predominance of the melody voice and the smoothness of the melody pitch contour. Two algorithm types for the identification of the melody voice can be distinguished: on the one hand, probabilistic frameworks are used to find the optimal succession of tones; on the other hand, there are rule-based approaches that trace multiple pitch contours over time. Both approaches are used with an equal incidence.

Chapter 3 dealt with the multi-resolution FFT (MR FFT) – a spectral analysis front-end that combines the multi-resolution approach of auditory models with the computational efficiency of the FFT. With the help of the MR FFT, a good frequency resolution was gained in the low frequency range and a good time resolution in the high frequency range. Regarding the spectral analysis, we found that it is not necessary to classify spectral peaks as sinusoids or noise, as the subsequent pitch estimation copes well with noisy input data.

A novel method for the estimation of the predominant pitch was introduced in Chapter 4. This method features the pair-wise evaluation of spectral peaks that helps to decrease the number of pitch candidates and to reduce octave errors. The predominant pitch extraction uses information from the tone tracking module to inhibit all spectral peaks which are already explained by existing tones. In this way, the pitch estimation identifies starting points for tones besides the predominant one. Contrary to other multiple F0 estimation methods (as for example the joint pitch estimation or the iterative pitch estimation), the proposed method is computationally very efficient, since the pitch spectrogram is only computed once per analysis frame.

In Chapter 5, we proposed a novel approach to tone tracking that exploits the principle of spectral smoothness in the frequency dimension as well as in the time dimension. As harmonics are not allowed to rise instantly to high magnitude values, the influence of noise and concurrent sound sources can be diminished. Moreover, the problem of shared harmonics was addressed, as well as octave errors and masked tones. Another challenge is the onset and offset detection in polyphonic music: in algorithms used for monophonic music, the most common onset detection functions are energy-based. However, since the magnitude of tones cannot be estimated with reliability in polyphonic music, an additional pitch-based onset detection function was used. This function is able to detect onsets not only in tones with a stable frequency, but also in tones that exhibit a pronounced vibrato.

Furthermore, we aim at the determination of MIDI notes, bridging the gap between audio and symbolic oriented methods. In addition to the estimation of the tone onset, tone offset and the perceived tone height, it is also necessary to estimate the tuning frequency of the musical piece. In this way, music can be analyzed that has not been recorded with the standard tuning of 440 Hz. We proposed a method based on circular statistics to compute the tuning frequency. As it uses all measured tone heights to compute the tuning, it converges to a stable result very quickly.

In Chapter 6, we proposed a musical voice tracking algorithm that is able to follow several voices at once. If two or more voices of comparable strength are present, the method offers the opportunity to output all salient voices instead of only one. This has a great advantage over probabilistic methods, which can only denote the pitch contour with the maximum likelihood. Another advantage is the capability to produce results with a relatively small latency (about 500 ms), while most probabilistic methods work offline. Primarily, the proposed voice tracking uses the tone's magnitude and the proximity to other melody notes to decide whether a tone belongs to the melody or not. However, it can be easily extended to also evaluate the timbre of the tone, provided that the timbre of a tone can be determined from the sound mixture.

The MIREX audio melody extraction evaluation shows that the proposed algorithm is among the state-of-the-art algorithms and that it works well on multiple genres of music. The algorithm was also evaluated in the MIREX multiple F0 estimation and tracking task, delivering the second best result. It is noteworthy that in addition to the very good overall accuracy, the algorithm was the fastest among the MIREX submissions.

8.2. Perspectives for Future Work

Improving the melody extraction accuracy further: This is a very obvious suggestion, yet during recent years the overall melody extraction accuracy estimated during the MIREX audio melody extraction challenge has not been significantly increased. The MIREX results suggest that a glass ceiling has been reached. While this is not easily overcome, the MIREX statistics show that there is at least theoretically room for improvement. The difference between raw pitch and overall accuracy, which amounts to 7.3 percent, shows that an enhancement in the voicing detection could result in a significant improvement. Unfortunately, it does not suffice to just decrease the magnitude threshold for melody tones, because in this case many notes from the accompaniment would be detected as melody tones. So while it might be possible to balance the ratio between false positives and false negatives a bit better, the question is whether there are possibilities to break the glass ceiling and improve the results significantly.

Towards timbre estimation from polyphonic music: Goto and Hayamizu (1999) first showed that the extraction of melody from real-world music recordings is possible. Their idea that the melody voice is the predominant voice in the musical mixture has paved the way for a very interesting research topic. Now research has reached a point, where it is necessary to look beyond the assumption that the melody is the predominant voice. Presently only two features are used to decide the voicing in our algorithm: the tone's magnitude and the frequency proximity to other melody tones. Of those two features, the magnitude of the tone is by far the most important one. The louder a tone, the greater is the probability that the tone belongs to the melody voice. However, the assumption that melody is the predominant voice is not valid for many musical recordings. This means that other features are needed to decide the voicing besides loudness. We think that the most important one is the timbre of the tone. It can be used to distinguish different instruments or the singing voice from the accompaniment. In order to estimate timbre, the harmonic magnitudes of all tones present in the signal have to be determined. Furthermore, the problem of shared harmonics has to be solved. While we have promising preliminary results in timbre estimation, results are not yet applicable for the detection of melody tones. Also, it is not clear how the potentially faulty timbre data can finally be used to enhance the algorithm.

Usage of higher-level cognitive information: The presented algorithm is mostly a bottom-up approach in which data is provided by the lower processing modules to the higher processing modules. However, music contains high-level information that could be exploited to improve the algorithm performance. Prior knowledge about musical events that also have predictive power could include expectations about the evolution of the melody and detected musical patterns. Also, context information (key, harmony, meter and rhythm) could be utilized to enhance the melody extraction accuracy.

Improve the evaluation methodology: Salamon and Urbano (2012) pointed out that current collections used for evaluation in MIREX are either too small or not sufficiently heterogeneous. That is why they called out for an annotation challenge, in which they emphasize the importance of establishing a common protocol for ground truth annotation (because small variances, specifically time offsets between the annotation and the audio, can have a dramatic effect on the results). Another idea worth considering is a listening test of synthesized versions of the extracted melody. This would help in the qualitative evaluation of the perceived accuracy of different types of extraction errors. For instance, a false negative (missing tone) might be less annoying than a false positive (erroneously detected tone). It might also be more acceptable to follow the bass voice for a time rather than having the extracted melody jump permanently between two tones. Since listening tests are very time consuming, another possibility could be to evaluate a more abstract representation of the melody data: instead of the melody pitch-contour MIDI notes could be used.

Gómez (2006) suggests that evaluation methodologies should include some robustness tests, for instance an increasing level of noise (decreasing SNR) in the music recording. Such a test set could also include melodies in unusual frequency ranges, uncommon timbres (such as sinusoids), inharmonic pitched sounds, white noise, or auditory illusions.

Specialized algorithms: Another opportunity for future work is to implement algorithms especially crafted for distinct genres or instruments (piano music, pop music with singing voice, etc.). Every genre or lead instrument has its own peculiarities that often can be exploited to improve the melody extraction accuracy – be it through distinct parameter settings or through entirely new concepts in the audio processing. For example, if the timbre of the lead instrument is known beforehand, those spectral regions could be amplified, or one could try to separate the sound source before attempting melody extraction.

Source separation prior to melody extraction in stereo files: Barry et al. (2004) presented a real-time sound source separation algorithm that performs the task of source separation based on the lateral displacement of a source within the stereo field. This algorithm exploits the interaural intensity difference that exists between left and right channels for a single source. The interesting aspect of this work is that the melody is in the vast majority of cases placed in the center of the stereo field (together with the percussion). With the help of the before-mentioned algorithm, it is possible to inhibit all sound sources that are placed outside the center, and in this way to enhance the melody voice in stereo recordings.

Applications that can handle erroneous melody data: While at present the extracted melody is far from perfect, it does not mean that the melody extraction algorithm is useless. Rather, the task is to build applications that can handle the errors. A possible candidate would be a Query-by-Humming system that is based on an automatically generated database of melodies. In contrast to the MIREX audio melody extraction challenge, it is not necessary to output only one voice as melody. If two voices have an equal loudness, they could both be saved as potential melody. In this way, at least one type of extraction errors can be avoided.

Another application could be audio to text alignment, in which the singing voice is synchronized with the song text. In this case, the timbre of the melody voice has to be estimated, but because only some valid points are needed to synchronize the data, there is no need for one hundred percent perfection.

The last suggestion for an application is cover song detection. At present, most applications that determine cover songs use harmony for the identification. This works very well in the small test collections offered, but because many songs share chord progressions, harmony alone is not sufficient in the long term. The melody of the song is a better indicator, as long as the application is robust enough to handle the typical melody extraction errors.

Bibliography

P. Assmann and Q. Summerfield. Modeling the perception of concurrent vowels: vowels with different fundamental frequencies. *Journal of the Acoustical Society of America*, 88(2):680–697, 1990. (Cited on page 97.)

P. F. Assmann and D. D. Paschall. Pitches of concurrent vowels. *Journal of the Acoustical Society of America*, 103:1150–1160, 1998. (Cited on page 98.)

D. Barry, B. Lawlor, and E. Coyle. Real-time sound source separation: Azimuth discrimination and resynthesis. In *Proc. 117th. Audio Engineering Society Convention*, pages –, San Francisco, CA, USA, October 28-31 2004. (Cited on page 124.)

E. Batschelet. *Circular Statistics in Biology.* Academic Press, 1981. (Cited on page 91.)

J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035 – 1047, September 2005. (Cited on pages 15, 16, and 82.)

E. Benetos and S. Dixon. Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1111–1123, October 2011. (Cited on pages 16 and 52.)

E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: breaking the glass ceiling. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012. (Cited on page 14.)

E. Benetos, S. Cherla, and T. Weyde. An efficient shift-invariant model for polyphonic music transcription. In *Proc. of the 6th International Workshop on Machine Learning and Music*, Prague, Czeck Republic, 2013. (Cited on pages 13 and 14.)

N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE transactions on Audio, Speech and Language Processing*, 18:538–549, 2010. (Cited on page 14.)

J. Bird and C. J. Darwin. *Psychophysical and physiological advances in hearing*, chapter Effects of a difference in fundamental frequency in separating two sentences, pages 263–269. Wiley, 1998. (Cited on page 97.)

S. Böck, A. Arzt, F. Krebs, and M. Schedl. Online real-time onset detection with recurrent neural networks. In *Proceedings of the 15th International Conference on. Digital Audio Effects (DAFx-12)*, York, UK, September 2012a. (Cited on page 22.)

S. Böck, F. Krebs, and M. Schedl. Evaluating the online capabilities of onset detection methods. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal, 2012b. (Cited on pages 16 and 82.)

A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*, volume 1 MIT Press paperback. MIT Press, Cambridge, Mass., Sept. 1994. (Cited on pages 51, 64, 97, and 100.)

J. C. Brown and M. Puckette. A high resolution fundamental frequency determination based on phase changes of the fourier transform. *Journal of the Acoustical Society of America*, 94:662–667, 1993. (Cited on pages 29 and 36.)

J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992. (Cited on page 10.)

J. C. Brown and K. V. Vaughn. Pitch center of stringed instrument vibrato tones. *Journal of the Acoustical Society of America*, 100(3):1728–1735, 1996. (Cited on pages 17 and 81.)

E. M. Burns and W. D. Ward. Categorical perception – phenomenon or epiphenomenon: Evidence from experiments in the perception of melodic musical intervals. *Journal of the Acoustical Society of America*, 63(2):456–468, 1978. (Cited on page 17.)

P. Cancela. Tracking melody in polyphonic audio. MIREX 2008. In *4th Music Information Retrieval Evaluation eXchange (MIREX)*, Philadelphia, Pennsylvania,USA, Sept. 2008. (Cited on page 15.)

C. Cao, M. Li, J. Liu, and Y. Yan. Singing melody extraction in polyphonic music by harmonic tracking. In *Proceedings of the 8th Conference on Music Information Retrieval (ISMIR)*, Austria, Vienna, Sept. 2007. (Cited on pages 15 and 49.)

R. P. Carlyon. Further evidence against an across-frequency mechanism specific to the detection of frequency modulation (fm) incoherence between resolved frequency components. *Journal of the A*, 95(2):949–961, 1994. (Cited on page 65.)

W. Chai. Melody retrieval on the web. Master's thesis, M.I.T. Media Laboratory, 2001. (Cited on page 16.)

F. Charpentier. Pitch detection using the short-term phase spectrum. In *Proc. IEEE Internat. Conf. on Acoust., Speech, Signal Processing*, pages 113–116, Tokyo, Japan, 1986. (Cited on pages 29 and 36.)

Y.-R. Chien, H.-M. Wang, and S.-K. Jeng. An acoustic-phonetic approach to vocal melody extraction. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011. (Cited on pages 17 and 98.)

Y.-R. Chien, H.-M. Wang, and S.-K. Jeng. Simulated formant modeling of accompanied singing signals for vocal melody extraction. In *Proc. of the 9th Sound and Music Computing Conference (SMC)*, Copenhagen, Denmark, 2012. (Cited on pages 10, 12, and 17.)

J. Chowning. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, chapter Perceptual Fusion and Auditory Perspective, pages 261–275. The MIT Press, Cambridge, MA, U.S.A., 1999. (Cited on pages 64 and 65.)

C. d'Alessandro and M. Castellengo. The pitch of short-duration vibrato tones. *Journal of the Acoustical Society of America*, 95(3):1617–1630, 1994. (Cited on pages 17 and 81.)

C. J. Darwin and G. J. Sandell. Absence of effect of coherent frequency modulation on grouping a mistuned harmonic with a vowel. *Journal of the Acoustical Society of America*, 97(5):3135–3138, 1995. (Cited on page 65.)

C. J. Darwin, V. Ciocca, and G. R. Sandell. Effects of frequency and amplitude modulation on the pitch of a complex tone with a mistuned harmonic. *Journal of the Acoustical Society of America*, 95:2631–2636, 1994. (Cited on pages 41, 64, and 65.)

A. de Cheveigné. Pitch perception models. In C. Plack and A. Oxenham, editors, *Pitch: neural coding and perception.* Springer Verlag, New York, 2005. (Cited on page 11.)

A. de Cheveigné and H. Kawahara. Multiple period estimation and pitch perception model. *Speech Communication*, 27:175–185, 1999. (Cited on pages 9, 13, 22, 35, and 50.)

A. de Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002. (Cited on page 12.)

A. de Cheveigné, H. Kawahara, M. Tsuzaki, and K. Aikawa. Concurrent vowel identification. i. effects of relative amplitude and f0 difference. *Journal of the Acoustical Society of America*, 101:2839–2847, 1997. (Cited on pages 62 and 66.)

A. Degani, M. Dalai, R. Leonardi, and P. Migliorati. Comparison of tuning frequency estimation methods. *Multimedia Tools and Applications*, 69:1–18, 2014. (Cited on page 90.)

C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch. *Music Information Retrieval Meets Music Education.* Dagstuhl Follow-Ups. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012a. (Cited on page 4.)

C. Dittmar, K. F. Hildebrand, D. Gaertner, M. Winges, F. Müller, and P. Aichroth. Audio forensics meets music information retrieval - a toolbox for inspection of music plagiarism. In *20th European Signal Processing Conference (EUSIPCO 2012)*, 2012b. (Cited on page 4.)

S. Dixon. A dynamic modelling approach to music recognition. In *Proceedings of the International Computer Music Conference (Computer Music Association, San Francisco CA)*, pages 83–86, 1996. (Cited on page 89.)

S. Dixon. Onset detection revisited. In *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada, 2006. (Cited on pages 16 and 82.)

J. S. Downie. Music information retrieval. *Annual review of information science and technology*, 37(1):295–340, January 2003. (Cited on page 3.)

J. S. Downie, X. Hu, J. H. Lee, K. Choi, S. J. Cunningham, and Y. Hao. Ten years of MIREX (music information retrieval evaluation exchange) : Reflections, challenges and opportunities. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014. (Cited on page 109.)

K. Dressler. Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 247–252, Montreal, Quebec, Canada, Sept. 2006. (Cited on pages 10, 22, and 23.)

K. Dressler. Audio melody extraction for MIREX 2009. In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, 2009. (Cited on page 22.)

K. Dressler. Pitch estimation by the pair-wise evaluation of spectral peaks. In *AES 42nd International Conference*, Ilmenau, Germany, 06 2011. (Cited on pages 36, 38, 39, 49, and 110.)

K. Dressler. Towards computational auditory scene analysis: Melody extraction from polyphonic music. In *Proc. of 9th Int. Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*, London, UK, 2012. (Cited on page 98.)

K. Dressler. Multiple fundamental frequency extraction for MIREX 2014. In *10th Music Information Retrieval Evaluation eXchange (MIREX)*, Taipei, Taiwan, October 2014a. (Cited on pages 112, 113, and 114.)

K. Dressler. Audio melody extraction for MIREX 2014. In *10th Music Information Retrieval Evaluation eXchange (MIREX)*, Taipei, Taiwan, October 2014b. (Cited on pages 12 and 115.)

K. Dressler and S. Streich. Tuning frequency estimation using circular statistics. In *Proc. of the 8th Int. Conf. on Music Information Retrieval, (ISMIR 2007)*, pages 357–360, Vienna, Austria, Sept. 2007. (Cited on pages 17 and 90.)

Z. Duan, J. Han, and B. Pardo. Harmonically informed multi-pitch tracking. In *10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, October 2009. (Cited on page 13.)

J.-L. Durrieu. *Automatic transcription and separation of the main melody in polyphonic music signals.* PhD thesis, Telecom ParisTech, Paris, France, 2010. (Cited on pages 14, 21, and 49.)

J.-L. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):564–575, Mar. 2010. (Cited on pages 12, 17, and 98.)

A. Elowsson and A. Friberg. Polyphonic transcription with deep layered learning. In *10th Music Information Retrieval Evaluation eXchange (MIREX)*, 2014. (Cited on pages 14, 22, and 114.)

V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010. (Cited on pages 10 and 13.)

J. Flanagan and R. Golden. Phase vocoder. *Bell System Technical Journal*, 45: 1493–1509, Sept. 1966. (Cited on pages 29 and 36.)

A. Gerson and J. L. Goldstein. Evidence for a general template in central optimal processing for pitch of complex tones. *Journal of the Acoustical Society of America*, 63(2):498–510, 1978. (Cited on page 40.)

V. Gnann, M. Kitza, J. Becker, and M. Spiertz. Least-squares local tuning frequency estimation for choir music. In *Proc. of the 131st AES Convention*, 2011. (Cited on page 90.)

E. Gómez. *Tonal Description of Music Audio Signals.* PhD thesis, UPF, Barcelona, Spain, 2006. (Cited on pages 89 and 124.)

E. Gómez, A. P. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32:2003, 2003. (Cited on page 72.)

E. Gómez, F. Cañadas, J. Salamon, J. Bonada, P. Vera, and P. Cabañas. Predominant fundamental frequency estimation vs singing voice separation for the automatic transcription of accompanied flamenco singing. In *Proc. of 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal, 2012. (Cited on pages 16, 83, 84, and 89.)

M. Goto. A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication (ISCA Journal)*, 43(4):311–329, Sept. 2004. (Cited on pages 12, 15, 18, 22, and 49.)

M. Goto and S. Hayamizu. A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, pages 31–40, Aug. 1999. (Cited on pages 7 and 123.)

F. J. Harris. On the use of windows for harmonic analysis with the Discrete Fourier Transform. In *Proc. IEEE, vol. 66, no. 1*, pages 51–83, Jan. 1978. (Cited on page 31.)

C. A. Harte and M. B. Sandler. Automatic chord identification using a quantised chromagram. In *Proc. of the 118th AES Convention, number 6412*, Barcelona, Spain, May 2005. (Cited on page 89.)

W. M. Hartmann. *Signals, Sound, and Sensation.* Springer-Verlag, 1997. (Cited on pages 40, 67, and 74.)

T. Heinz. *Ein physiologisch gehörgerechtes Verfahren zur automatisierten Melodietranskription.* PhD thesis, TU Ilmenau, 2006. (Cited on pages 9, 11, 22, 82, and 83.)

H. Heo, D. Sung, and K. Lee. Note onset detection based on harmonic cepstrum regularity. In *IEEE Int. Conference on Multimedia and Expo (ICME)*, San Jose, California, USA, 2013. (Cited on page 82.)

D. J. Hermes. Measurement of pitch by subharmonic summation. *Journal of the Acoustical Society of America*, 83(1):257–264, 1988. (Cited on page 12.)

H. Honing. *Musical Cognition: A Science of Listening.* Transaction Publishers, New Brunswick, N.J., 2011. (Cited on page 4.)

C.-L. Hsu, L.-Y. Chen, J.-S. R. Jang, and H.-J. Li. Singing pitch extraction from monaural polyphonic songs by contextual audio modeling and singing harmonic enhancement. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, Oct. 2009. (Cited on pages 17 and 98.)

D. Huron. Voice denumerability in polyphonic music of homogeneous timbres. *Music Perception*, 6(4):361–382, 1989. (Cited on page 100.)

D. Huron. What is a Musical Feature? Forte's Analysis of Brahms's Opus 51, No. 1, Revisited. *Music Theory Online*, 7(4), July 2001. (Cited on page 75.)

D. Huron. *Sweet Anticipation: Music and the Psychology of Expectation.* The MIT Press, Cambridge, Massachusetts, 2006. (Cited on pages 2, 98, and 99.)

Y. Ikemiya, K. Yoshii, and I. Katsutoshi. MIREX 2014: Audio melody extraction. In *10th Music Information Retrieval Evaluation eXchange (MIREX)*, 2014. (Cited on pages 12, 17, and 21.)

S. Joo, S. Jo, and C. D. Yoo. Melody extraction from polyphonic audio signal MIREX 2009. In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, Kobe, Japan, Oct. 2009. (Cited on pages 17 and 98.)

F. Keiler and S. Marchand. Survey on extraction of sinusoids in stationary sounds. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 51–58, Hamburg, Germany, 2002. (Cited on pages 28 and 31.)

C. Kereliuk and P. Depalle. Improved hidden Markov model partial tracking through time-frequency analysis. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 111–116, Espoo, Finland, September 1-4 2008. (Cited on page 15.)

A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999. (Cited on pages 82 and 83.)

A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Int. Conf. on Music Information Retrieval (ISMIR)*, pages 216–221, 2006. (Cited on pages 11, 14, 21, and 52.)

A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Trans. Audio, Speech and Language Processing*, 16(2): 255–266, February 2008. (Cited on pages 9, 10, 12, 14, and 35.)

A. Klapuri and M. Davy, editors. *Signal processing methods for music transcription.* Springer, 2006. (Cited on page 35.)

A. P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11 (6):804–816, 2003a. (Cited on pages 12, 13, 33, 35, 50, and 51.)

A. P. Klapuri. *Signal Processing Methods for the Automatic Transcription of Music.* PhD thesis, Tampere University of Technology, Dec. 2003b. (Cited on pages 10, 21, and 22.)

A. Laaksonen. Automatic melody transcription based on chord transcription. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014. (Cited on pages 16 and 89.)

M. Lagrange, S. Marchand, M. Raspaud, and J.-B. Rault. Enhanced partial tracking using linear prediction. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 141–146, London, UK, September 2003. Queen Mary, University of London. (Cited on page 15.)

A. Lerch. On the requirement of automatic tuning frequency estimation. In *Proc. of the 7th International Conference on Music Information Retrieval, (ISMIR 2006)*, pages 212–215, Victoria, Canada, Oct. 2006. (Cited on page 89.)

J.-Y. Lin and W. M. Hartmann. The pitch of a mistuned harmonic: Evidence for a template model. *Journal of the Acoustical Society of America*, 103(5):2608–2617, May 1998. (Cited on page 67.)

R. C. Maher and J. W. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 95(4):2254–2263, 1994. (Cited on page 12.)

S. Marchand and M. Lagrange. On the equivalence of phase-based methods for the estimation of instantaneous frequency. In *Proceedings of the 14th European Conference on Signal Processing (EUSIPCO'2006)*, Florence, Italy, September 2006. (Cited on pages 11, 28, and 29.)

P. Masri. *Computer Modeling of Sound for Transformation of Musical Signals*. PhD thesis, University of Bristol, Dec. 1996. (Cited on page 22.)

P. Masri and A. Bateman. Identification of nonstationary audio signals using the FFT,with application to analysis-based synthesis of sound. In *Proc. IEE Colloquium on Audio Engineering, Digest No. 1995/96*, pages 11/1–11/6, London, UK, 1995. (Cited on page 22.)

M. Mathews. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, chapter Introduction to Timbre, pages 79–87. MIT Press, 2001. (Cited on page 32.)

M. Matthews. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, chapter Introduction to Timbre, pages 79–87. The MIT Press, 1999. (Cited on page 65.)

McAulay and Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744 – 754, 1986. (Cited on page 14.)

R. J. McNab, L. Smith, I. Witten, C. Henderson, and S. Cunningham. Towards the digital music library: tune retrieval from acoustic input. In *Proceedings of the ACM International Conference on Digital Libraries – DL'96*, 1996. (Cited on pages 83 and 90.)

R. Meddis and M. J. Hewitt. Modeling the identification of concurrent vowels with different fundamental frequencies. *Journal of the Acoustical Society of America*, 91(1):233–245, 1992. (Cited on page 62.)

C. Micheyl and A. J. Oxenham. Pitch, harmonicity and concurrent sound segregation: Psychoacoustical and neurophysiological findings. *Hearing Research*, 266 (1-2):36–51, 2010. (Cited on page 62.)

S. Mithen. *The Singing Neanderthals: The Origins of Music, Language, Mind and Body.* Orion Books, 2006. (Cited on page 3.)

B. C. Moore. *An Introduction to the Psychology of Hearing*, volume 5th. Academic Press, San Diego, California, 2003. (Cited on pages 11, 36, 47, and 84.)

B. C. Moore, B. Glasberg, and R. Peters. Thresholds for hearing mistuned partials as separate tones in harmonic complexes. *Journal of the Acoustical Society of America*, 80(2):479–483, 1986. (Cited on page 67.)

M. Müller. *Fundamentals of Music Processing.* Springer, 2015. (Cited on pages 14 and 16.)

A. H. Nuttall. Some windows with very good sidelobe behavior. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 1:84–91, Feb. 1981. (Cited on pages 27, 28, and 31.)

R. P. Paiva. *Melody Detection in Polyphonic Audio.* PhD thesis, University of Coimbra, 2006. (Cited on pages 11, 22, 49, and 72.)

R. P. Paiva, T. Mendes, and A. Cardoso. An auditory model based approach for melody detection in polyphonic musical recordings. *Lecture Notes in Computer Science - Computer Music Modeling and Retrieval: Second International Symposium, CMMR 2004*, 3310:21–40, Feb. 2005. (Cited on pages 9 and 12.)

R. P. Paiva, T. Mendes, and A. Cardoso. From pitches to notes: Creation and segmentation of pitch tracks for melody detection in polyphonic audio. *Journal of New Music Research*, 37(3r):185–205, 2008. (Cited on pages 16, 17, 83, and 89.)

T. W. Parsons. Separation of speech from interfering speech by means of harmonic selection. *Journal of the Acoustical Society of America*, 60(4):911–918, 1976. (Cited on page 50.)

A. Pertusa and J. M. Iñesta. Multiple fundamental frequency estimation using gaussian smoothness. In *IEEE International Conference on Acoustics, Speech and Signal Processing: ICASSP 2008*, pages 105–108, 2008. (Cited on pages 10, 13, 35, and 52.)

S. Pinker. *WIe das Denken im Kopf entsteht.* Kindler, München, 1998. (Cited on page 3.)

C. J. Plack and A. J. Oxenham. *Pitch: neural coding and perception*, chapter Overview: The Present and Future of Pitch, pages 1–6. Springer Verlag, 2005. (Cited on page 35.)

G. E. Poliner and D. P. Ellis. Classification-based melody transcription. *Machine Learning Journal*, 65(2-3):439 – 456, December 2006. (Cited on pages 12 and 49.)

G. E. Poliner and D. P. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007. (Cited on page 112.)

E. Prame. Vibrato extent and intonation in professional western lyric singing. *The Journal of the Acoustical Society of America*, 102(1):616–621, 1997. (Cited on page 81.)

M. Puckette and J. C. Brown. Accuracy of frequency estimates using the phase vocoder. *IEEE Transactions on Speech and Audio Processing*, 6(2):166–176, 1989. (Cited on page 21.)

V. Rao and P. Rao. Improving polyphonic melody extraction by dynamic programming based dual f0 tracking. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, Sept. 2009. (Cited on pages 17, 18, 21, and 98.)

V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2145 – 2154, November 2010. (Cited on pages 12 and 49.)

D. Recoskie. Constrained nonnegative matrix factorization with applications to music transcription. Master's thesis, University of Waterloo, 2014. (Cited on pages 13 and 14.)

M. Ryynänen. *Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies.* PhD thesis, Tampere University of Technology, Mar. 2004. (Cited on page 89.)

M. Ryynänen and A. Klapuri. Transcription of the singing melody in polyphonic music. In *Proc. 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 222–227, Victoria, Canada, Oct. 2006. (Cited on pages 16, 18, and 89.)

M. Ryynänen and A. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008. (Cited on page 83.)

J. Salamon. *Melody Extraction from Polyphonic Music Signals.* PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2013. (Cited on pages 10, 21, and 49.)

J. Salamon and E. Gómez. A chroma-based salience function for melody and bass line estimation from music audio signals. In *6th Sound and Music Computing Conference (SMC 2009)*, pages 331–336, Porto, Portugal, 2009. (Cited on page 110.)

J. Salamon and E. Gómez. Melody extraction from polyphonic music: MIREX 2011. In *7th Music Information Retrieval Evaluation eXchange (MIREX)*, 2011. (Cited on pages 10, 21, and 118.)

J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing,*, In Press (2012), 2012. (Cited on pages 11, 15, 17, 18, and 98.)

J. Salamon and J. Urbano. Current challenges in the evaluation of predominant melody extraction algorithms. In *Proc. 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal, October 2012. (Cited on page 123.)

C. E. Seashore. *Psychology of Music.* McGraw-Hill, New York, London, 1938. (Cited on page 81.)

X. Serra. *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition.* PhD thesis, Stanford University, 1989. (Cited on pages 15, 22, 28, and 31.)

J. I. Shonle and K. E. Horan. The pitch of vibrato tones. *Journal of the Acoustical Society of America*, 67(1):246–252, 1980. (Cited on pages 17 and 81.)

L. Song, M. Li, and Y. Yan. Melody extraction for vocal polyphonic music based on bayesian framework. In *The 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu*, Kitakyushu, Japan, 2014. (Cited on pages 10, 12, 17, 21, and 98.)

J. Sundberg. Vibrato and vowel identification. *Speech Transmission Laboratory, Quartery Progress and Status Report (KTH, Stockholm) STL-QPSR*, 16(2-3):49–60, 1975. (Cited on page 65.)

J. Sundberg. Perception of singing. *Speech Transmission Laboratory, Quartery Progress and Status Report (KTH, Stockholm) STL-QPSR*, 20(1):1–48, 1979. (Cited on page 33.)

H. Tachibana, T. Ono, N. Ono, and S. Sagayama. Melody extraction in music audio signals by melodic component enhancement and pitch tracking. In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, Kobe, Japan, Oct. 2009. (Cited on page 76.)

H. Tachibana, T. Ono, N. Ono, and S. Sagayama. Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2010)*, pages 425–428, Dallas, Texas, March 2010. (Cited on pages 10, 19, and 21.)

E. Terhardt. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *Journal of the Acoustical Society of America*, 71(3):679–687, Mar. 1983. (Cited on pages 36, 37, 39, 44, and 59.)

B. Thoshkahna and K. Ramakrishnan. An onset detection algorithm for query by humming (qbh) applications using psychoacoustic knowledge. In *Proc. of the 17th European Signal Processing Conference (EUSIPCO 2009)*, Glasgow, Scotland, 2009. (Cited on pages 82 and 83.)

C. C. Toh, B. Zhang, and Y. Wang. Multiple-feature fusion based onset detection for solo singing voice. In *Proc. of 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*, Philadelphia, Pennsylvania, USA, 2008. (Cited on page 82.)

C. D. Tsang and L. J. Trainor. Spectral slope discrimination in infancy: Sensitivity to socially important timbres. *Infant Behavior and Development*, 25(2):183–194, 2002. (Cited on page 32.)

J. Vos and R. Rasch. The perceptual onset of musical tones. *Attention, Perception and Psychophysics*, 29(4):323–335, 1981. (Cited on page 57.)

R. M. Warren. *Auditory perception: an new analysis and synthesis.* Cambride University Press, 1999. (Cited on pages 15, 86, and 99.)

M. Wendelboe. Using OQSTFT and a modified SHS to detect the melody in polyphonic music (MIREX 2009). In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, Kobe, Japan, Oct. 2009. (Cited on pages 17, 49, and 98.)

C. Yeh. *Multiple fundamental frequency estimation of polyphonic recordings.* PhD thesis, Université Paris VI, France, June 2008. (Cited on page 52.)

C. Yeh, A. Roebel, and X. Rodet. Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1116–1126, 2010. (Cited on pages 10, 13, and 35.)

T.-C. Yeh, M.-J. Wu, J.-S. R. Jang, W.-L. Chang, and L. I-Bin. A hybrid approach to singing pitch extraction based on trend estimation and hidden Markov models. In *IEEE ICASSP*, Kyoto, Japan, 2012. (Cited on page 12.)

Appendix A.

Exponential Moving Average

A simple moving average (SMA) is the unweighted mean of the previous N data points. An exponential moving average (EMA) applies weighting factors to all previous data points which decrease exponentially, giving more importance to recent observations while still not discarding older observations entirely. The smoothing factor α determines the impact of past events on the actual EMA. It is a number between 0 and 1. A lower smoothing factor discards older results faster.

The graph in Figure A.1 shows the weight decrease for the last 15 data points and $\alpha = 0.8$. The first weight has the value of $1 - \alpha$, while any other weight can be calculated using the value of the preceding weight multiplied with α. The sum of all weights approaches 1 for a big number of data points. Consequently, the computation of the EMA can be expressed by the following formula

$$\bar{y}_t = (1 - \alpha) \sum_{i=0}^{t-1} \alpha^i y_{t-i}, \tag{A.1}$$

where the observation at a time period t is designated y_t, and the value of the EMA at time period t is designated $\bar{y}_t$.

Of course the application of equation A.1 is inconvenient, because it represents an infinite sum. The same result can be achieved by using the following recursive formula for time periods $t > 0$:

$$\bar{y}_t = \alpha \cdot \bar{y}_{t-1} + (1 - \alpha) \cdot y_t. \tag{A.2}$$

Equation A.2 shows that the EMA at time period t can be calculated very efficiently from only two numbers: the current observation data y_t and the preceding EMA value $\bar{y}_{t-1}$. Thus, a big advantage of this method is, that (besides the last EMA value) no previous data has to be stored in order to calculate the average.

The first EMA value $\bar{y}_0$ at time period $t = 0$ is undefined and has to be initialized to make the recursive computation possible. This may be done in a number of different ways. Most commonly it is set to value of the first observation data y_0. The problem of this technique is, that the first observation gains a huge impact on the subsequent

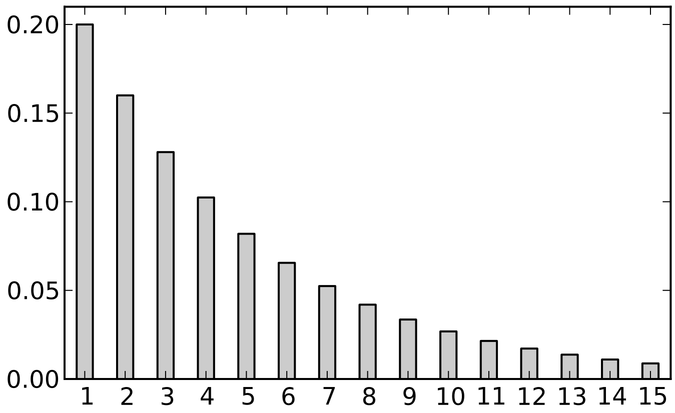


Figure A.1. Exponential Weights for $\alpha = 0.8$

EMA results. This can be very disturbing, especially if the first data point is not very representative for the rest of the data.

As another option the first EMA value can be set to 0. In this case the observations have comparable weights, but the first calculated EMA values do not represent an average of the observations. Rather, the EMA value starts close to zero and then approaches the average slowly – just like a sampled capacitor charging curve. Nonetheless, this technique makes sense for some applications, for example it can be useful if the average magnitude of a tone starts with low values.

If it is not acceptable, that the first EMA value is set to 0, the most accurate EMA values can be achieved with the following procedure. At the beginning the EMA value is set to 0. The update of the EMA is computed as given in equation A.2. Yet, for each time period, the sum of the used weights $\bar{w}_t$ is also calculated recursively using:

$$\bar{w}_t = \bar{w}_{t-1} \cdot \alpha + (1 - \alpha) \qquad \text{with} \quad \bar{w}_0 = 0. \tag{A.3}$$

Finally, the corrected EMA value is the EMA value divided by the sum of weights:

$$\bar{y}_t^* = \frac{\bar{y}_t}{\bar{w}_t}. \tag{A.4}$$

Both of the last mentioned techniques are used in the melody extraction algorithm. If the EMA is used in calculations, the assumed start value of the EMA is 0. If the corrected EMA value is used, it is specified explicitly.

For the actual implementation it is important to figure out optimal values for the smoothing factor α. A more intuitive measure than the smoothing factor is the so called half-life period. It denotes the time span over which the initial impact of an observation decreases by a factor of two. Taking into account the desired half-life t_{HL} and the time period between two EMA calculations Δt, the corresponding smoothing factor is calculated as follows:

$$\alpha = 0.5^{\frac{\Delta t}{t_{\mathrm{HL}}}}. \tag{A.5}$$

In the scope of the melody extraction algorithm Δt is the time advance between two successive MR FFT frames ($\approx$ 5.8ms).